Research and Practice on Agile Development Method for Teaching with Lego Blocks

Ming Zhu
College of Computer Science and Technology
Shandong University of Technology
Zibo, China

Kang Du
School of Chemistry and Chemical Engineering
Shandong University of Technology
Zibo, China

Jing Li
College of Computer Science and Technology
Shandong University of Technology
Zibo, China

Abstract:- Agile development is a lightweight software development method with user-centered, iterative and evolutionary features. This method has gradually become a hot topic in research on teaching software engineering. However, the existing teaching methods of agile development may have some problems such as requiring a long teaching period and insufficiency of evaluating results of teaching. To solve such problems, a teaching method of agile development in a classroom using Lego blocks is proposed. In this method, models of Lego blocks are used as the project to be constructed, instead of using a real software project. In a 100-minute classroom teaching session, the Lego blocks models are built iteratively by Scrum in agile development. After the class, data of the practice are gathered and analyzed, and performances of students in the classroom are evaluated. Through several experimental practices, results show that this teaching method can achieve the expected goal of teaching agile development.

Keywords:- *Software Engineering; Classroom Teaching; Agile Development; Scrum; LEGO Blocks.*

I. INTRODUCTION

With the rapid development of the new generation of information technology, great changes are taking place in the field of software engineering. As a widely adopted software development method, the agile development method takes the user's needs as the core and adopts the method of multi-round iteration and team cooperation to develop software [1]. Among them, the Scrum agile development method has been widely used in the industry [2]. The application of the agile development method in undergraduate teaching practice has become a research hotspot of software engineering curriculum reform in recent years. The practical teaching of agile development mainly focuses on the application of this method in curriculum design and classroom teaching, which can improve students' software development ability to a certain extent, and enable curriculum construction to quickly follow up industrial changes [3]. However, there are some problems in the existing teaching methods, such as complex estimation methods, long course cycle, and incomplete result evaluation. Learning the agile development method puts forward higher requirements for the participants' software development experience and practical ability, which increases the difficulty of classroom teaching.

To ameliorate these problems, this paper proposes a teaching method based on the Lego blocks to teach Scrum agile development process. The main contributions include: 1. A classroom teaching method based on Lego blocks for agile development is proposed. It uses constructs of Lego blocks to simulate the components in a software project. This practice of agile development can be completed in 100 minutes. 2. A method based on Lego blocks is proposed to evaluate the teaching of agile development. It includes the process of collecting, adjusting, screening, analyzing and evaluating data of the Scrum practice based on Lego blocks.

The rest of this paper is structured as follows: Section II introduces the background knowledge of agile development methods. Section III explains the challenge that the classroom teaching for agile development methods encounters. Section IV introduces the objectives of the agile development teaching in the classroom, and Section V proposes the design and evaluation of the classroom teaching method based on Lego blocks. Section VI describes the practices of performing the proposed teaching method to indicate its effectiveness. Section VII provides the conclusion of the paper.

II. AGILE DEVELOPMENT METHODOLOGY

The birth of the Agile Manifesto in February 2001 marked the creation of agile development methods [4]. Compared with traditional software development methods, agile development has the following characteristics: individual and interaction are more important than processes and tools, workable software is more important than complex documents, customer collaboration is more important than contract negotiation, and response to change is more important than always following a fixed plan. Agile development methods include extreme programming, crystal method, dynamic system development method, lean development, and Scrum [5]. Among them, Scrum is one of the most prevalently used development methods, which can reflect the core idea of agile development [2]. In the Scrum method, people involved in the development of the project include a product owner, a Scrum master, and Scrum development team [2]. Then, according to user needs, requirements of the project are translated into prioritized user stories. The Scrum development team collaborates to gradually realizes all user stories through multiple iterations. After each iteration, the Scrum team summarizes the current iteration of development through standing meetings, constantly adjusts the plan, and sorts out the development list of the next iteration to guide the follow-up work. Furthermore, to support Scrum development, several auxiliary techniques are required, which include planning poker technique, a list of user stories, burndown chart.

The planning poker technique estimates the size of user stories and develops release and iteration plans [6]. Firstly, developers independently evaluate each user story, and then through group discussions to further clarify the user's needs. Each developer writes down his/her estimates on poker cards without discussing it with each other. The numbers used in the estimation are usually from the Fibonacci sequence. Secondly, when all developers have completed their estimates, the cards are displayed together to show the results. If all estimates are the same, record the results and continue to estimate the next user story. Otherwise, the team needs to discuss the estimated results together, give their reasons respectively, and re-estimate the user story until all the numbers are agreed upon.

A user story is a technique to capture a description of a software requirement [7]. A backlog of user stories records necessary information of user stories, in which a description of each user story contains the type of user, what they want and why, as well as story point [8]. As mentioned previously, for each user story, the story point is evaluated by planning poker technique. Usually, user stories are arranged from high to low by priority in the backlog, and the development team selects them, in turn, to design, implement, and test in each sprint. Once a user story is completed, the product owner needs to confirm whether the corresponding requirement is fulfilled or not.

A burndown chart is a graphic representation that illustrates the progress of each sprint [9]. It is used to evaluate how quickly the team is working through user stories. In the burndown chart, the quantity of work remaining is shown on a vertical axis, while the time that has passed on a horizontal axis. When the remaining workload becomes zero at the end of a sprint, the team completes all the user stories in the backlog amid the sprint. It is customary to update a burndown chart gradually. By the end of each sprint, the progress of the development is summarized and analyzed to guide the development in the next sprint.

III. THE CHALLENGE OF TEACHING AGILE DEVELOPMENT IN CLASSROOM

In the traditional teaching of software engineering, it often emphasizes waterfall model, spiral model, rapid prototype model, iterative model, incremental model, and some other models [10]. Among them, the agile development model is considered as both iterative and incremental that has several benefits, such as embracing changes, continuous integrations, collaborations between product owners and developments [2]. However, in reality, developing a software project by following the agile development model often takes several sprints, where each sprint requires a week or more days. Therefore, how to design and practice the classroom teaching for the agile development model, specifically for the Scrum model, has become a challenge [10].

To tackle this challenge, several initiatives have been proposed by researchers, which include but not limited to the following. Steghöfer *et al*. propose a method to analyses the changes they have made in teaching agile methodologies, practices, and principles in four courses to address a specific dilemma that is students need to apply agile methods to learn them, but students perceive the learning and application of agile methods as less important than delivering a finished product [11]. Subburaj *et al*. study the globalization of software development activities, and aim at designing instructional materials and assessment tools to develop a unique mix of skill sets [12]. Fitsilis *et al*. raise a survey among current and former computer science students, gathering data from 200 students that attended university in the past 5 years, to investigate students' views on how the teaching of agile methods help to shape their future skills in working in the software industry [13]. Kurkovsky's research describes a LEGO-based activity for multiple teams to practice collaborative design, parallel development, and component integration to illustrate the advantages of well-designed component interfaces [14]. Yang et al. propose an experimental framework of a new pedagogical method using LEGOs for teaching software estimation and measurement concepts and methods, and this framework is evaluated by analyzing practice sessions performed two groups of students [15].

Most of the above-mentioned research, to a certain extent, can better the quality of teaching and student satisfaction, however, further improvements to project estimation, teaching process, and result evaluation are still necessary. This motivates our work on a classroom teaching method based on Lego blocks and an evaluation method that assesses the quality of the proposed teaching method.

IV. OBJECTIVES OF AGILE DEVELOPMENT TEACHING IN THE CLASSROOM

The agile development of classroom teaching methods based on Lego blocks requires teaching students systematic agile development ideas, cultivating students' correct development thinking, and strengthening the scientific and practical nature of classroom teaching. This teaching method has the following classroom teaching objectives.

A. Mastery of the agile development process. The basic goal of this classroom teaching method is to teach students to develop relevant knowledge and methods in a limited time. The various methods and links used in the teaching process are to better illustrate the process of agile development so that students can have a deeper understanding of agile development methods.
B. Team organization and collaboration ability training. One of the basic concepts of agile development is team development. The quality of team cooperation directly determines whether the agile development process can go smoothly. In classroom teaching, the corresponding user story point estimation, multi-round sprint development, and after-class homework discussion are set up to train students' team cooperation ability to stimulate students' interest and motivation in learning.
C. Cultivation of engineering thinking and engineering quality. Students need to follow the development process and engineering specifications, develop good development habits and develop in strict accordance with the order of evaluation, iteration, regular meetings, adjustment and delivery of agile software development. This kind of classroom teaching method should lay a foundation for students' future engineering development.

V. DESIGN AND EVALUATION OF THE CLASSROOM TEACHING METHOD

Given the challenge faced by teaching agile development, according to the abovementioned objectives, an agile development classroom teaching method based on Lego blocks is proposed, which mainly includes three stages: pre-class preparation, classroom teaching, and homework evaluation, as shown in Figure 1.

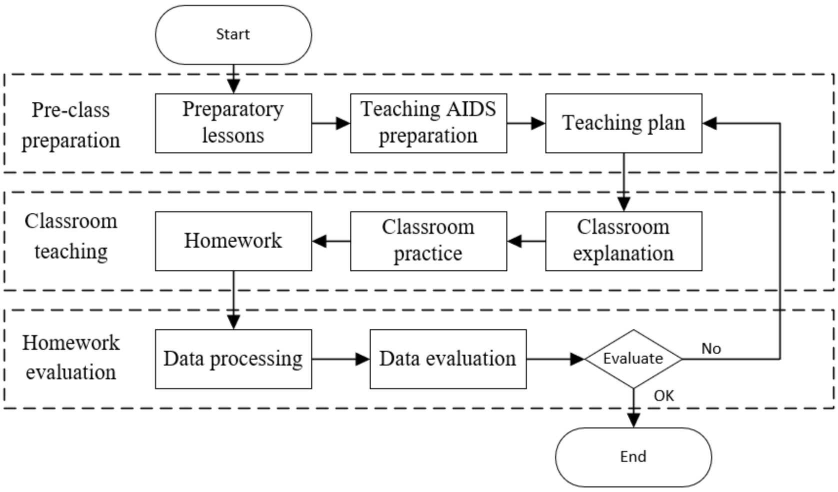

Fig 1:- Classroom Plan for Agile Software Development

A. Pre-class preparation

In the pre-class preparation stage, teachers should formulate a reasonable classroom teaching plan according to the students' learning situation and the target requirements of teaching practice activities. This stage mainly includes three parts: preparation of lessons, preparation of teaching aids and teaching plan.

- Preparatory lessons: Before this round of classroom teaching, the teacher needs to arrange preview tasks ahead of time, students need to understand the background knowledge of agile development and a brief introduction of development methods. The teacher needs to clarify teaching objectives, key and difficult points, teaching methods, study textbooks and understand the situation of students' learning.

- Teaching aids preparation: This course needs to prepare planning poker and Lego blocks. The numerical range of planning poker needs to conform to the size of Lego blocks. The size and difficulty of the selected Lego blocks model need to be moderate to meet the curriculum requirements and students' capacities.

- Teaching plan: This class exploits the discussion method and experiment method in classroom teaching. It chooses models of Lego blocks that are more operable and acceptable to students as the simulation of a software project. It adopts planning poker, user stories and burndown chart as techniques for developing the project. This plan aims to carry out practice teaching of the agile development method in two classes (100 minutes), and the teacher and students participate in the practice together. At the end of the class, the teacher and students summarize and evaluate the classroom practice activities.

B. Classroom Teaching

Classroom teaching is the central part of this method, which mainly includes three parts: classroom explanation, classroom practice, and homework, as shown in Figure 2.

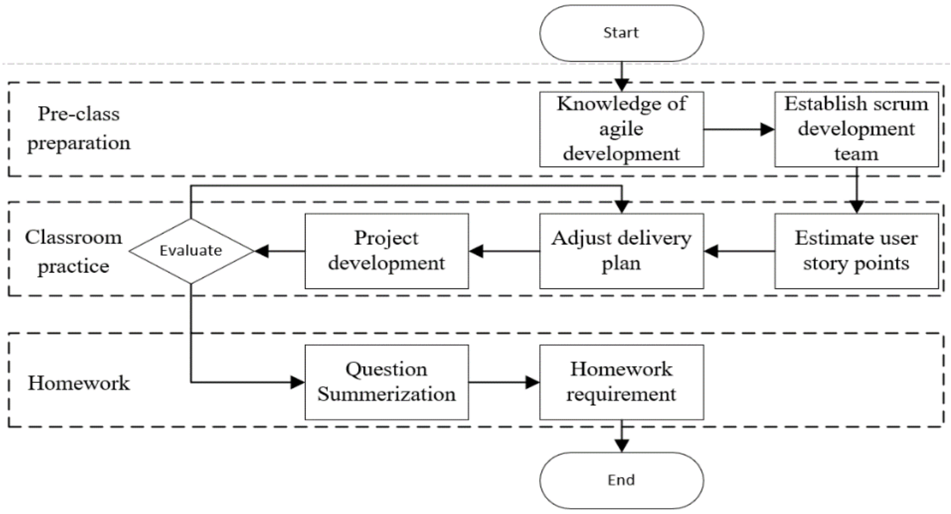

Fig 2:- Classroom Teaching

- Classroom explanation: In class, the teacher introduces the plan and requirements of this course, and establishes a scrum development team. It's planned to take 20 minutes. Firstly, the teacher introduces the basic knowledge of Scrum development and the auxiliary tools for Scrum to the students. It plans to take 10 minutes. Secondly, the teacher guide students to establish Scrum development teams, where each team is with a Scrum master. The teacher acts as the product owner of each team. It takes around 10 minutes for the team formation.

- Classroom practice: Students develop models of LEGO blocks according to the product owner's requirements. The total duration for the classroom practice is 70 minutes. First of all, the development team uses planning poker to evaluate the corresponding user story points of models of Lego blocks, which takes 15 minutes. Secondly, the team makes a development plan, estimates the development speed of the Lego project under optimistic and pessimistic conditions, defines the number of iterations required. The second step takes 10 minutes. Then, the team starts a sprint to build models of Lego blocks according to the backlog. By the end of the sprint, models will be assessed by the product owner, and the team may adjust the delivery plan according to the actual development speed. In the class, students can carry out three sprints, where each sprint takes 15 minutes and the total duration is 45 minutes.

- Homework: After the classroom practice, the teacher summarizes the practice and poses questions related to students' performances of the practice, and requests each team to submit a report recording the process and result of practice as homework. The summarization by the teacher takes around 10 minutes. Specifically, the report needs to include but not limit to the following essential components: the burndown charts for three in-class sprints; reexamination of the actual development progress after the three sprints; adjustment of the development speed under optimistic and pessimistic conditions based on the actual development speed; adjustments of the duration of the project if it is necessary; risks and problems identified in the three sprints that may affect the completion of the project, and strategies that are adopted to mitigate or solve the identified risks and problems.

C. Homework Evaluation

Homework evaluation requires the teacher to analyze and evaluate the reports submitted by students, including three parts: data processing, data evaluation, and summary feedback, as shown in Figure 3.

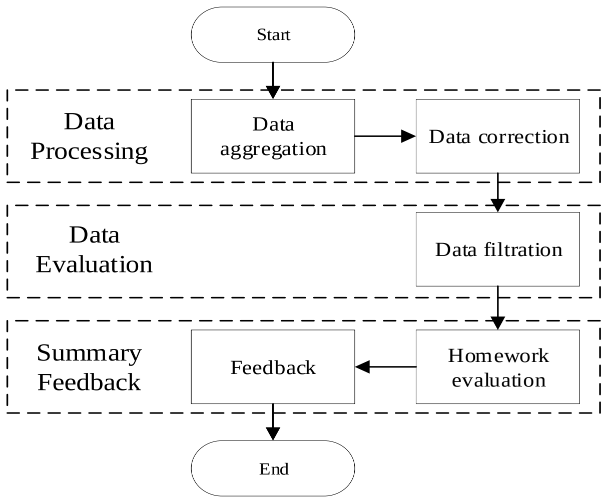

Fig 3:- Process of work evaluation

1) Data processing

Firstly, the data are aggregated. The teacher mainly collects the project workload in terms of user story points and the average speed of Scrum development in student practice reports. Among them, the project effort is the sum of all user stories' points, as shown in formula (1).

$$Effort = \sum Story\ Point \qquad \text{Formula (1)}$$

The average speed of Scrum development (*Vi*) is defined as the amount of project effort (*Effort*) that a team can accomplish in a unit time (*t*) (or an iteration), as shown in formula (2).

$$Vi = \frac{Effort}{t}$$ Formula (2)

Then, the data are corrected for further processing. To evaluate the development process more accurately, according to the characteristics of the classroom teaching method based on Lego blocks, this paper considers 10 main factors that can influence the development process in Table 1. These influencing factors are called friction factors, each of which indicates a reason that hinders the development process. To evaluate the impact of friction factors on the development process, this paper proposes a friction method to calculate the impact degree. By introducing friction into the analysis process, the Scrum development speed can be further adjusted to reflect the actual development situation of the project more accurately.

Friction Factors	Regular	Influence	Greater impact	Influential
Unreasonable Story Point Setting	1	0.98	0.95	0.91
Problems in the division of personnel and tasks	1	0.98	0.94	0.89
Understanding of the project is insufficient	1	0.99	0.95	0.91
Uncertainty of iteration time	1	0.99	0.97	0.95
No parts found	1	0.96	0.93	0.9
Unreasonable Sprint Planning	1	0.98	0.95	0.92
Team inefficiency	1	0.98	0.96	0.94
The project has some difficulty	1	0.97	0.94	0.91
Lack of experience	1	0.98	0.96	0.94
Low efficiency of product validation	1	0.99	0.98	0.97

Table 1:- Friction Factor: Factors Affecting Project Development (Friction Factors）

Friction (*FR*) is a numerical expression of the speed, which is mainly based on 10 friction factors (*FF*), as shown in formula (3).

$$FR = \prod_{i=1}^{10} (FF)_i$$ Formula (3)

By incorporating frictional force (*FR*) into the projected velocity analysis process, the adjusted average velocity (*V*) can be calculated based on average velocity (*Vi*), as shown in formula (4).

$$V = (Vi)^{FR}$$ Formula (4)

2) Data Evaluation

To improve the accuracy and reliability of performance evaluation, the adjusted data should be filtered before the evaluation of data. According to the practical experience, this paper utilizes the formula (5) to filter the average speed of Scrum development

$$V = 0.9Vi + 0.1$$ Formula (5)

If the adjusted average velocity *V* can be distributed in the smaller range of the straight line or both sides of the straight line where formula (5) is located, the accuracy of the estimated results will be higher.

After the adjustment and filtering data, it is necessary to evaluate the project completion of each Scrum development team. Firstly, formula (6) and formula (7) are used to calculate the adjusted Scrum development time ($T_{correction}$) and the actual Scrum development time (*T*) respectively.

$$T_{correction} = \frac{Effort}{V}$$ Formula (6)

$$T = \frac{Effort}{Vi}$$ Formula (7)

Then the ratio of the adjusted speed of Scrum development time to the actual Scru development time, namely the efficiency ratio (E), is calculated by formula (8), which can be used as a basis for evaluating the completion of each development team.

$$E = \frac{T_{correction}}{T}$$ Formula (8)

The closer the *E* is to 1, the closer the actual development time and optimization time are. The more deviated the *E* is from 1, the more the actual development situation cannot meet the planning. It is stipulated here that the efficiency ratio $E \in (120\%, 80\%)$ is a better estimate, $E \in (150\%, 120\%) \cup (80\%, 50\%)$ is a good estimate, and the rest is a worse estimate.

3) Data Evaluation

Combined with the evaluation of homework and the summary of students' reports, reflect on the problems existing in the practice stage. The problem-solving method is fed back to the process of making teaching plan in the course preparation stage, and the original plan is improved properly, so that the next teaching activity can be carried out smoothly, in order to achieve the ideal effect.

ISSN No:-2456-2165

VI. PRACTICES OF THE TEACHING METHOD

To verify the feasibility of the agile development method of Lego blocks proposed in this paper, classroom teaching activities have been practiced many times among the third-year students majoring in computer science and technology in a university in China. The teaching practice lasted for two years. The first year's practice consists of a class with 40 students who are divided into eight groups with five in each group. The second year's practice consists of a class with 40 students, and the formation of groups is the same as that of the previous year. LEGO-10696 is used in classroom teaching. Thirteen LEGO models including crocodile, eight petals, windmill cabin, and locomotive are selected to be developed. The teaching time is 100 minutes. Finally, 16 sets of valid data were collected with 8 sets from each year. The feasibility validation method is validated by the data processing and data evaluation steps according to the agile development speed estimation method proposed in Section 4 job evaluation.

D. Data processing

➢ *Data aggregation*

The workload, optimistic average speed, pessimistic average speed and the average speed of Scrum development in actual development are collected and summarized. 8 groups of practice data are randomly selected, as shown in Table 2.

Group	1	2	3	4	5	6	7	8
Effort	636	136	752	165	105	52	70	145
***V* optimistic**	160.5	34	160.5	27.5	21	10.4	18	25
***V* Pessimism**	107	31	107	23.6	15	8.7	14	20.17
Vi	132.3	36.3	132.3	23.3	17	10	16.3	23.5

Table 2:- Summary of classroom teaching feedback data

➢ *Data adjustment*

Based on the feedback from the practice reports of the above 8 groups of development teams, the average speed of Scrum development with optimistic estimation is used as the average speed, and the adjusted average speed is calculated by using formulas (3) and (4). The adjusted average velocity is compared with the actual average velocity, as shown in Table 3.

Group	1	2	3	4	5	6	7	8
V	125	28.86	131.3	20.58	17.04	10.4	14.37	20.66
Vi	132.3	36.3	132.3	23.3	17	10	16.3	23.5

Table 3:- Comparison of the average and actual average velocities of the adjusted optimistic estimates

E. Data evaluation

➢ *Data filtering*

Figure 4 shows the average velocity and the corresponding adjusted average *Vi* and *V* distribution, data filtering curve.

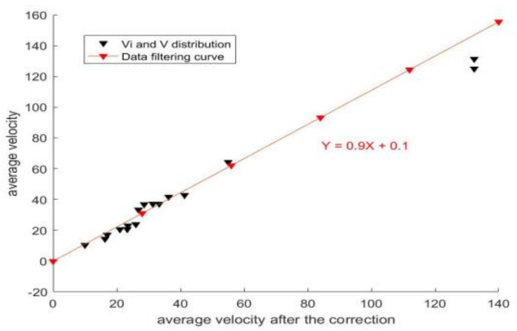

Fig 4:- Average Velocity and Positive Average Velocity Distribution Curve

From Figure 4, it can be concluded that there is a strict first-order curve relationship between the average velocity and the average velocity after the correction, which shows that the data adjustment and optimization scheme is feasible. Sample points near the red line are selected for subsequent evaluation.

➢ *Evaluation*

The efficiency ratio evaluation formula (8) of the complete evaluation is used to further process and evaluate the practical data, and Table 4 is obtained.

Group	1	2	3	4	5	6	7	8
Effort	636	136	752	165	105	52	70	145
Tcorrection	5.09	4.71	5.73	8.02	6.16	5	4.87	7.02
T	4.81	3.75	5.68	7.08	6.18	5.2	4.29	6.17
E	105.82%	125.6%	100.88%	113.28%	99.68%	96.15%	113.52%	113.78%

Table 4:- Efficiency ratios of team completion time

From Figure 4, it can be concluded that there is a strict first-order curve relationship between the average velocity and the average velocity after the correction, which shows that the data adjustment and optimization scheme is feasible. Sample points near the red line are selected for subsequent evaluation.

➢ *Summary feedback*

This classroom practice is guided by the agile development classroom teaching method based on Lego blocks. The results of the data analysis show that the groups participating in classroom practice can achieve good development results. Classroom teaching activities are carried out step by step according to the classroom plan, and finally achieve the established teaching objectives, and the teaching plan is feasible. At the same time, through the analysis of the friction factor table, find out the factors that affect the progress of the development team, and make corresponding consideration and response in the next round of classroom teaching.

VII. CONCLUSIONS

This paper introduces an agile teaching method in class based on Lego blocks. Instead of developing a real software project, this method uses models of Lego blocks as the project to be constructed. It proposes a design and evaluation method that tailed the Scrum development process to fit in-classroom teaching based on Lego blocks within 100 minutes. Specifically, the approach is composed of pre-class preparation, classroom teaching, and homework evaluation stages. Students iteratively develop models of Lego blocks based on user stories through team collaboration and communication with product owners. Once the models of Lego blocks are delivered, the development process and delivery results are evaluated by the proposed evaluation method. To evaluate the effectiveness of the proposed method, a 2-year practice of teaching Scrum with Lego blocks is carried out. The results of practice show that this teaching method can achieve the expected goal of teaching agile development.

REFERENCES

[1]. Schwaber, K. and M. BeedleAgile, software development with Scrum, Prentice-Hall, 2002.
[2]. Girma, M., Garcia N. M. and Kifle, M., Agile Scrum Scaling Practices for Large Scale Software Development, Proceedings of the 4th International Conference on Information Systems Engineering (ICISE), Shanghai, pp.34-38, 2019.
[3]. Hsu, H.J., Lin, E., Chang, K. and Hsiao, E., Practicing Scrum in Institute Course, Proceedings of the 52nd Hawaii International Conference on System Sciences, Hawaii, pp.7770-7778, 2019.
[4]. Fowler, M. and Highsmith, J., The Agile Manifesto. Software Development, 2001.
[5]. Anwer, F., Aftab, S., Waheed, U. and Muhammad, S.S., MuhammadAgile Software Development Models TDD, FDD, DSDM, and Crystal Methods: A Survey, International Journal of Multidisciplinary Sciences and Engineering, Vol. 8, No.2, pp.1-10, 2017.
[6]. Mahnič, V. and Hovelja, T., On Using Planning Poker for Estimating User Stories, Journal of Systems and Software, Vol.85, No.9, pp.2086-2095,2012.
[7]. Ormsby, M. and Busby-Earle, C., Agile Scaled Steps of Doneness: A Standardized Procedure to Conceptualizing and Completing User Stories Across Scrum Teams and Industries. In: Damiani E., Spanoudakis G., Maciaszek L. (eds) Evaluation of Novel Approaches to Software Engineering. ENASE 2019. Communications in Computer and Information Science, vol.1172. Springer, pp.364-377, 2019
[8]. Gannon, M., An Agile Implementation of Scrum, 2013 IEEE Aerospace Conference, Big Sky, MT, pp. 1-7, 2013.
[9]. Woodward, C. J., Cain, A., Pace, S., Jones, A. and Kupper, J. F., Helping Students Track Learning Progress Using Burn Down Charts, Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), Bali, pp.104-109, 2013.
[10]. Tiwari, S. and Rathore, S. S., Teaching Software Process Models to Software Engineering Students: An Exploratory Study, 2019 26th Asia-Pacific Software Engineering Conference (APSEC), Putrajaya, pp. 308-315, 2019.
[11]. Steghöfer, J., Knauss, E., Alégroth, E., Hammouda, I., Burden, H. and Ericsson, M., Teaching Agile - Addressing the Conflict between Project Delivery and Application of Agile Methods, IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C), Austin, TX, pp.303-312, 2016.
[12]. Subburaj, V.H., Hunt, E., Spaulding, A. and Webb, J., Challenges In Teaching Global Software Engineering To Undergraduate Students: Course Design, The Online Journal of Science and Technology, Vol.9, No.1, pp.68-74, 2019
[13]. Fitsilis, P. and Lekatos, A., Teaching software project management using agile paradigm, Proceedings of the 21st Pan-Hellenic Conference on Informatics, New York, pp.1–6, 2017.
[14]. Kurkovsky, S., Using LEGO to teach software interfaces and integration, Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, New York, pp.371–372, 2018.
[15]. Yang, Y. and Laird, L., Teaching Software Estimation through LEGOs, IEEE 29th International Conference on Software Engineering Education and Training (CSEET), Dallas, pp.56-65, 2016.