I.J. Modern Education and Computer Science, **2016, 7, 13-21**
Published Online July 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2016.07.02

The Proposed Methods to Improve Teaching of Software Engineering

Afra A. Alabbadi and Rizwan J. Qureshi
Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
Email: a1.edu@live.com, rmuhammd@kau.edu.sa

***Abstract*—**The use of software is an essential part of our modern lives. Hence, this increases the importance of studying Software Engineering (SE) course. In general, the software engineering graduates usually lack the necessary skills, expertise, abilities, and sufficient knowledge when beginning their careers in the software industry. Due to that, a majority of students find it difficult to find relevant jobs. This paper proposes novel methods to teach SE course to improve the teaching and enhance knowledge and skills of students. It is proposed to include: identifying the course learning objectives (CLOs) and the required skills of the SE course, combining cooperative learning and mastery learning strategies to teaching software engineering, using social media to teach SE course, and establish the OSES in an educational institution. The goal of this paper is to improve the quality of SE teaching and facilitate students learning to prepare them for their future careers. Qualitative technique is used as a research design to evaluate the proposed solution. The results indicate that this proposal is supported by the majority of professionals working in the academia and industry.

***Index Terms*—**Software engineering, education, cooperative mastery, social media, industry experts, methods of teaching.

I. Introduction

Software Engineering (SE) is the discipline providing tools and methods to build quality software within deadline and a limited budget in a systematic method [1]. SE course is of importance that should be taught in effective methods. It comprises many knowledge and skills, which is a basic requirement for each student and must be acquired. The skills are diverse from solving problems, dealing with several requirements, working in teams, the developing a programming and analytical skills and communication skills.

A generic software engineering course in most of the educational institutions cannot prepare graduates to face the actual-time industry environment challenge [2]. The current methods of teaching software engineering adopted at many educational institutions and universities appear to have a significant gap, compared to the real-world application of SE. Generally, universities focus on the theoretical aspect of SE while giving less interest to the practical skills that enhance student's ability to be equipped with the needed skills for their future career. In addition, students often consider the teaching delivery methods in software engineering courses as abstract concepts without actual practical impact. Therefore, students often do not have a deep understanding of important ideas and their realistic implementations in SE [3]. The educational institutions should improve the education methods and promote student's skills in all fields of SE. In addition, it is fundamental for researchers and universities to develop efficient teaching methods to provide students with required skills. Such approaches should be tailored in order for the development of required skills of the students to become experts in all fields of SE. In addition, the new approaches should adopt activities simulating their individual learning while facilitating their ability to acquire real-time experiences for the learners for a better future professional career.

The paper is structured as follows. In section II, the related work is presented. The statement of the problem is described in section III. In section IV, the suggested proposed solution is presented. The solution is validated by the means of the survey in section V.

II. Related Work

Zhang et al. [4] developed an approach based on Computer Supported Collaborative Learning (CSCL) to carry out collaborative learning theories between students and teachers for online software engineering education. The system provided functional cognitive apprenticeship where the learners can interact using question and answer or discussions between learners by prompting questions and answers feature of the system, which distribute cognition suggests. Based on the analysis of the results, the researchers proved some patterns of collaborative learning, which collaborative learning theories suggest. In the experiment, few questions and answers were exchanged; a utility of this function was not confirmed. However, this study is a small-scale study in terms of the size of the sample and the period. In order for the study to be more reliable, the study needs to use a broad sample and an extended period.

The main objectives in [5] are to develop and to improve approach for software engineering education quality and training. The primary idea is to state a goal, recognized as important by the partners from three several areas of science, education, and business, and then to attempt to unify their efforts to perform the set

goal. The main principles of the approach are presented, with a systematic methodology INSPIRE and the formal method constructed in it. A few real-life examples of the way the application is applied are briefly described. At last, the researchers should be considered for a collection of long-term goals of science and education together with the short-term business-oriented goals, through the suitable formulation of some particular tasks of the joint work.

A new methodology is presented to teach software engineering (SE) courses [6]. The approach is based on combining several activities in SE course such as gaming, workshops, role-plays and demo of SE tools. The methodology assists students in enhancing their knowledge, problem-solving skill, communication skill, and quality to complete projects. The primary limitation in the methodology is the assessments focused on student understanding and their application more than on memory.

The research applied inverted classroom concept to introductory software engineering education to measure the efficiency of the motives for the controlled experiment, one of the two courses was carried out in the traditional teaching method with software development projects. The other course incorporated the use of watching pre-VOD at a time outside of the classroom and in-class activities like the design studio and game storming. The learning performance of students who have taken part was measured and the results of the project were compared to evaluate the effectiveness of the inverted classroom in teaching software engineering. The main limitation of educational models is the need of the continuous communication between instructors and learners, which can be difficult to understand software engineering. Therefore, Inverted classroom videos need to be improved in order for the students to focus more on planning and class activities.

Ding [8] proposed a novel framework based on experiences learned from the previous projects. The framework involves plans to constructing an effective collaboration infrastructure, and guidelines for designing a quality collaboration courses. The primary goal is to promote the quality and efficiency of teaching SE courses. The framework is useful in the future for other works that can be involved in other global collaboration projects. However, there is a need to mainstream the framework to include the students in different countries then standardize courses contents and the collaboration infrastructure.

Most of the projects used in Software Engineering (SE) curricula lack the practical realism of engineering projects with the fun factor, which cover other computer science (CS) disciplines. Claypool et al. [9] attempted to promote interest in a currently SE curriculum by using of computer game-based projects. A set of project-based modules and game-centric have been developed, which make students able to: first, participate in the various phases of the software lifecycle, taking one project from Identify requirements into maintenance. Second, it helps student to deal with real issues in project and team management on the course of a two-semester project. Finally, it introduces students to various aspects of computer game design. The results of the research indicate into showing improved performance and class participation. The work needs to explore other opportunities for game-enhanced of other CS courses.

Shata [10] proposed an approach to teaching SE course based on experience in adopting an applied-oriented manner. The paper focused on the most commonly used models, which students might encounter in their future carriers. In addition, it ignored traditional subjects that are taught for comparison purposes only. The approach has enabled students to use development processes to design software based on software engineering principles, to gain skills necessary for industrial practices, and work in diverse roles in software development. Nevertheless, the course project needs to do with more investigation on the proper size of it.

A team of SE researchers, from institutions in various territories, has embarked on the research project, which aimed to improve the teaching of SE for four years undergraduate levels [11]. The project is a set of workshops, for that the goal is to gather data which will be the basis of the research effort. The data-gathering goal sought to identify a set of topics, which is considered suitable for teaching SE, along with identification of the years at which the subjects should be taught. The paper also recommended using technology in the curriculum design as a critical phase. The primary goal of the paper is to improve a learning online environment that will be available to instructors who are teaching in the area of SE. The research group needs to continue its efforts to recruit SE researchers and educators from institutions in other territories to participate in the research and the workshops [11].

According to the role of gaming technologies together with software engineering techniques, Xie et al. developed a web-based educational gaming environment (Pex4Fun) [12]. The central idea of Pex4Fun games is a coding duel where the player has to resolve a particular programming problem. Teaching based on coding is complicated, which needs more skills and knowledge.

Jun et al. [13] introduced four approaches, i.e., project-based training, case-based teaching, research-based learning, and integrated practice. Besides, an experience-based approach is presented, which will help to teach principles, and practices of software engineering at Dalian Polytechnic University (DLPU). The students have expressed a high level of satisfaction. They also indicated that what they learned in the course apply to their careers. So, the experience-based approach should be implemented in a large experiment.

Sepahkar et al. [14] presented a case study of executing a project-based learning program in Isfahan University of Technology. The project-based learning program developed based on primary steps: Project definition, setting up a team, holding a workshop, System analyze, system design and system test. The primary objectives of the program are educating its member and giving them a practical experience in web programming. The report states that this case study was entirely successful.

Nevertheless, a team project should contain experienced people for the success of the project.

Gamification and collaborative learning methods can be used to motivate students to achieve course objectives in the educational environment [15]. The paper presented a case study and an evidence-based method where interaction analysis and k-means clustering are used to make gamification profiles. The profiles can be utilized with an agent-based simulation to test how computer supported collaboration system users interact with the gamification elements and how the collaboration dynamics change. The main limitation of this study is that results cannot be generalized due to the limited sample size.

An approach is proposed to the SE course based on the Problem Based Learning (PBL) [16]. The main idea of PBL is to make students focus on a problem to solve it. There are three principles of PBL. First, students solve a problem weekly. Second, students work jointly to solve problem. Third, the instructor's role is to help the students and instruct them.

Dagnino [17] presented a method to teach SE that focuses on industrial experiences. The primary goal of the approach is to improve how the graduate level SE courses are taught. This method is based on new elements that are incorporated in the SE course. The result of the paper shows is showing an improvement in the instructor evaluation from the students. However, the method should be implemented in a wider experiment and longer time to apply to undergraduate students.

Berkling et al. [18] described the setup for a gamified classroom for the topic of SE course. The objective of gamifying the course was to increase motivation of student during allowing for independent learning with emphasis and flexible speed. In addition, there is motivational theory in gamification factors, outlining that autonomy, mastery and purpose lead to these objectives. The adopted approach also deals with the large differences regarding background knowledge of the students. At the end of the course, a student survey was conducted. Students did not receive the gamification ideas in a positive light. Therefore, the methodology needs to include some points that help students to figure out the concept of gamification approach.

There are some limitations in all the previously mentioned approaches. These limitations are displayed in Table 1.

III. Problem Statement

Most of the Software Engineering graduates face difficulties to join software industry. In addition, students suffer from traditional ways of teaching, which focused on teaching theoretical that led to a lack of practical experience. This paper attempts to address the following problem that specified from the previous research [6],[8], and [13].

- How can the quality of teaching software engineering be improved?
- What methods do we use to enhance SE knowledge and skills for students?
- How can students be equipped with required skills and knowledge to work in software development industry?

IV. The Proposed Solution

Instructors should use modern methods, strategies and techniques in the regular classroom, especially in teaching SE effectively. Among the educational theories and teaching strategies that have taken the attention of modern teachers, two most common ones are cooperative learning and mastery learning. These two strategies are an important parts of numerous education-improvement programs. We propose novel methods combining cooperative learning and mastery learning to teach SE. That helps to improve the quality of teaching software engineering, and enhance student knowledge and the skills required to achieve determined course learning objectives (CLOs) in our approaches. Moreover, today's generation of students has extensive experience in social media. Therefore, we propose the use of social media in teaching SE to enhance awareness, learning and knowledge sharing. In addition, we suggest the OSES and hire of industry experts to prepare students for software development in industry and fill in the gap between industry and academics.

A. Identify the CLOs of the SE Course with the help of students

The setting of educational objectives is a necessary first step in any educational process. And it is the main guided for the teacher and learner alike. To achieve this step first, the instructor divides the concepts, skills, and content of SE they want the students to learn, into instructional units. For each instructional unit, the instructor determines the course learning objectives (CLOs). At this step, we recommend the involvement of students in the identification and formulation of CLOs. To increase their motivation to learn and achieve these objectives, the course learning objectives should include required skills, diagrams, models and concepts that must be given more time, so that the students can master them. Then, instructor should provide students with CLOs at the beginning of the semester. As for the learners, they will work on each unit content in a series of sequential steps to achieve the predetermined achievement level.

In the end, students are expected to achieve the following educational objectives of the course (not necessarily all):

- Possess the necessary professional skills in the field of software engineering and that makes them confident of the development of high-quality software solutions in various application areas under realistic constraints.

Table 1. Summary of the Related Work.

Title	Summary
A Collaborative Learning Support System for Software Engineering Education [4].	• The learners can do interactions like discussions or question and answer among learners by a question and answering feature of the system. However, in the experiment, as few questions and answers were exchanged, a utility of this function was not confirmed. • The proposed approach should implement in a broad sample and an extended period.
A collaborative approach to software engineering education [5].	• The researchers should be considered for a collection of long-term goals of science and education together with the short-term business-oriented goals, during suitable formulation of some particular tasks of the joint work.
Activity based-teaching learning in software engineering-An experience [6].	• The assessments focused on student understanding and their application more than on memory.
Applying Inverted Classroom to Software Engineering Education [7].	• The main limitations of educational models are the needs of the continuous communication between instructors and learners, which can be difficult to understand software engineering. • Inverted classroom videos need to improve, so that students can focus more on planning and class activities.
A framework for global collaboration in teaching software engineering [8].	• There is a need to mainstream the framework to include the students in different countries then standardize courses contents and the collaboration infrastructure.
Teaching Software Engineering Through Game Design [9].	• The work needs to explore other opportunities for game-enhanced of other CS courses which help to motivate and to improve the CS learning of undergraduate students or earlier introductory courses.
Teaching Software Engineering A Critical Path Method [10].	• The course project needs to do with more investigation on the proper size of it. • Researchers must address how to prevent some students from took the project to a very detailed level and developing very detailed diagrams to the point that it affected their coding and later phases.
Enhancing Teaching & Learning of Software Engineering in an International Environment [11].	• The research group needs to continue its outreach effort to recruit SE researchers and educators from institutions in other territories to participate in the workshops and research effort.
Educational software engineering: Where software engineering, education, and gaming meet [12].	• Teaching based on coding is complicated, which is needed more skills and knowledge.
Exploration and reformation of teaching methods for "Software Engineering [13].	• The experience-based approach needs to implement in a large experiment.
Defining Project Based Learning steps and evaluation process for software engineering students [14].	• A team project should contain experienced people for the success of the project.
Creating Software Engineering Student Interaction Profiles for Discovering Gamification Approaches to Improve Collaboration [15].	• The main limitation of this study is that results cannot be generalized due to the limited sample size.
Improving Software Engineering Education through Enhanced Practical Experiences [16].	• Students may resist accepting the proposed approach because they prefer traditional classroom-based learning.
Increasing the Effectiveness of Teaching Software Engineering: A University and Industry Partnership [17].	• The methods should implement in a wider experiment and longer time to apply to undergraduate students.
Gamification of a Software Engineering Course [18].	• Students do not understand the factors of gamification. So they not able to propose changes for the next iteration of the course. • The methodology needs to include some points that help students to figure out the idea of gamification approach.

- Participation and success in their careers through teamwork, ethical behavior, and effective communication.
- Understanding of the importance of lifelong learning through professional development and practical training.

B. Teach the SE Course Using Cooperative Mastery Strategy

We propose novel methods to teach SE by combining cooperative learning and mastery learning. The method helps to improve the quality and effectiveness of teaching SE, enhance student knowledge, and the skills required

and prepare students for software development in industry. Cooperative mastery strategy is achieved through creating teams of 4–6 students to perform tasks of SE courses. If students work in groups, a variety of ideas, knowledge and approaches could be exchanged, which leads to many benefits of cooperative learning, which might not be available in traditional learning. There should be defined roles for every student in the team to perform tasks and rotating these roles for every assignment or periodically. Assign various roles to team members that include coordinator, checker, recorder and group process monitor. In addition, the Instructor should be made heterogeneous teams in abilities and intelligence levels. Because with heterogeneous teams, the weak students benefit from observing better students, and the better students benefit by having a deeper understanding when they help poorer students to deal with problems of the SE. Also, every student should make sure each member of the group can report and discuss on what he/she did. Diversity in methods to teach SE units during lectures is required to prevent monotony and increase students' motivation. To address this, methods to be adopted may include discussion, problem-solving, case-based teaching, project based learning, gaming, simulation, and storytelling.

At the end of each unit, students should be given assessment as a means of measuring if learning objective has been mastered or not. The instructor can use practical assessment, diagnostic, formative tests to perform the assessment which could as serve avenue to direct teaching instruction later. It is important at ech stage the instructor ensure that learners are able to demonstrate to have mastered 80% or more understanding based on the unit's assessment, before moving on to next unit. At this stage, instruction goes in one of two:

- **If the student has mastered 80% of the learning objective:** Some students will achieve well on the first test, proving that they have mastered the unit skills and concepts. The instructor's initial instruction was appropriate for these students, and they have no need for the corrective task. To guarantee their continued learning the instructor should provide the students with particular “extension” or “enrichment” activities to extend their learning experiences. These activities are self-selected by students. In addition, activities may involve specific projects, academic games and reports or a variety of complex tasks.
- **If the student has mastered 80% of the learning objective:** In a situation some students do not perform to expected level on the first test, proving that they have not mastered the unit skills and concepts. It could be that, the instructor's initial instruction art this stage is considerd not appropriate for the students, and the need for a corrective task. At this stage, the instructor provides series of corrective tasks, which can involve different activities, individualized learning, and additional time to complete a task, and these students will receive feedback on their task and will be encouraged to revise their work until the objective is mastered. With the feedback gained from the formative assessment, each student has a detailed instruction to finish so that he/she master the skills of the unit. The correction work prevents most learning difficulties from accumulating and becoming a large learning problem. It also gives the instructor an actual means to a different type of instruction to students’ individual learning needs. As a result, majority students learn well and master the necessary learning objectives in all units. Fig. 1 shows the methods of corroborative-mastery strategy.

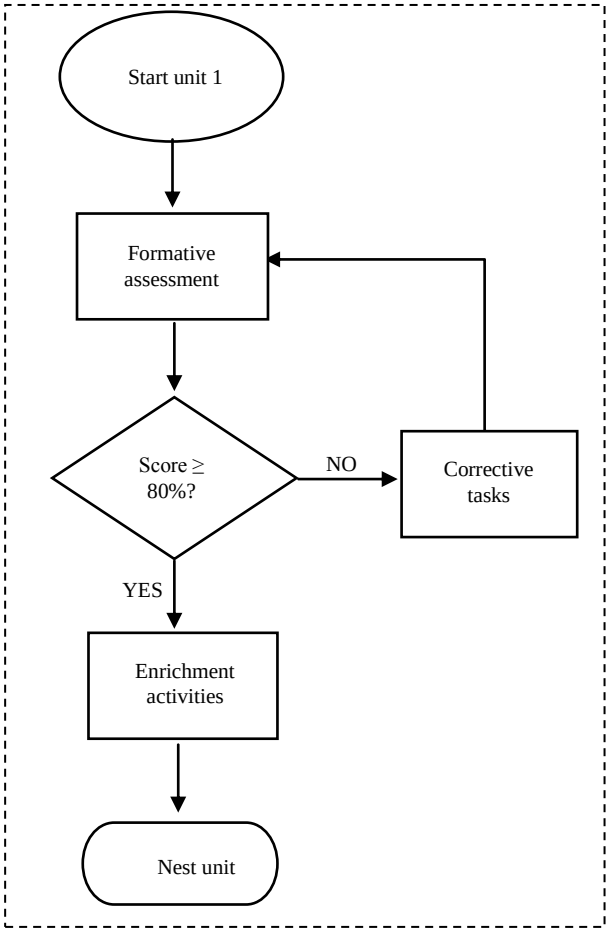

Fig.1. The cooperative-mastery strategy.

C. Effect of Using Social Media to Teach SE Course

Today's software developers usually use of social media, whether as an assistant or integrated into a broad range of tools such as issue trackers, code editors, and IDEs [19]. In addition, The effective communication between students and the community enhances the effectiveness of teaching. So, we propose to teach SE using social media to achieve universal benefits Where the desired information about software engineering course will be available on the internet to help communication between students, and faculty from various universities and software developers. The role of

social media usage to teach SE enhances awareness, learning and knowledge sharing. It increases attendee interaction and engagement between student and community. To avoid the disadvantages of the misuse of social media by students, we recommend that students should be allowed to interact with these sites under the constant supervision by the teacher as shown in fig. 2.

There several types of social media features that can be used to support SE teaching and practice. They are as following:

- **Wikis** that have a broad adoption in SE.
- **Blogs** are frequently used to support requirements engineering.
- **Microblogs** used to exchange a lot of information between software developers.
- **Tagging** used to in project for tagging issues.
- **Social networking is** a useful feature to share, discover and collaborate between software developers.
- **Podcast** provides tutorials and interviews for professional software developers as shown in fig. 3.

D. *Establish the OSES Industry Experts to Improve Quality and Fill the Gap between Industry and Academics*

Usually, faculty lack in industry experience so, the research proposed to Establish the Office Software Engineering Services (OSES) in educational institution .which, administered by one or more industry experts to provide SE education services to teachers and students. In the fact, hire a software industry experts for workshops, consulting, and supervision of the education process. Having industry experts in the educational institution helps both the instructor and student like workshops from industry experts are useful as they enhance the knowledge of the students and fill the gap between industry work and the students. The Office Software Engineering Services (OSES) can be provide:

- Workshops periodically.
- Consulting and services for teachers and students.
- Supervising on Identify the correct CLOS in line with the industry.
- Help students to communicate with international experts through social media.
- Guiding and Stimulate students in their learning practice.
- Presentation of industry stories.

V. VALIDATION

The online questionnaire technique is used to validate proposed solution that mentioned early. The questionnaire consists of 16 close ended questions which cover our goals: Goal 1- Identify the CLOs of the SE Course with the help of students. Goal 2- teach the SE course using cooperative mastery strategy. Goal 3 - an effect of using social media to teach SE course. Goal 4 - Establish the OSES industry experts to improve quality and fill the gap between Industry and academics. The questions were answered on a likert scale as shown in Table 2.

A. *Goal 1- Identify the CLOs of the SE Course with the Help of Students*

The four questions in this goal measure the importance of Identify the CLOs of the SE course with the help of students. Table 3 is showed that 3, 20.25% are agreed to define CLOs and give it to students at the beginning of the semester. 22 % are strongly agreed. Furthermore, 7.5% are responding neutrally. However, 2.5% are disagreed while other 0.5% are strongly disagreed. Fig. 4 is showed the result graphically.

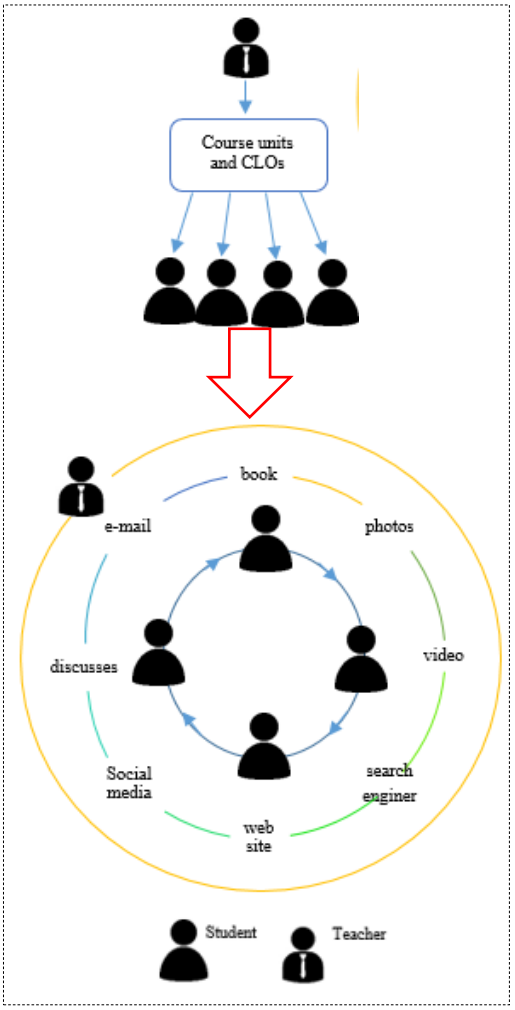

Fig.2. Social Media to teach SE.

Table 2. Likert Scale

5	Strongly Agreed
4	Agreed
3	Neither Agreed Nor Disagreed
2	Disagreed
1	Strongly Disagreed

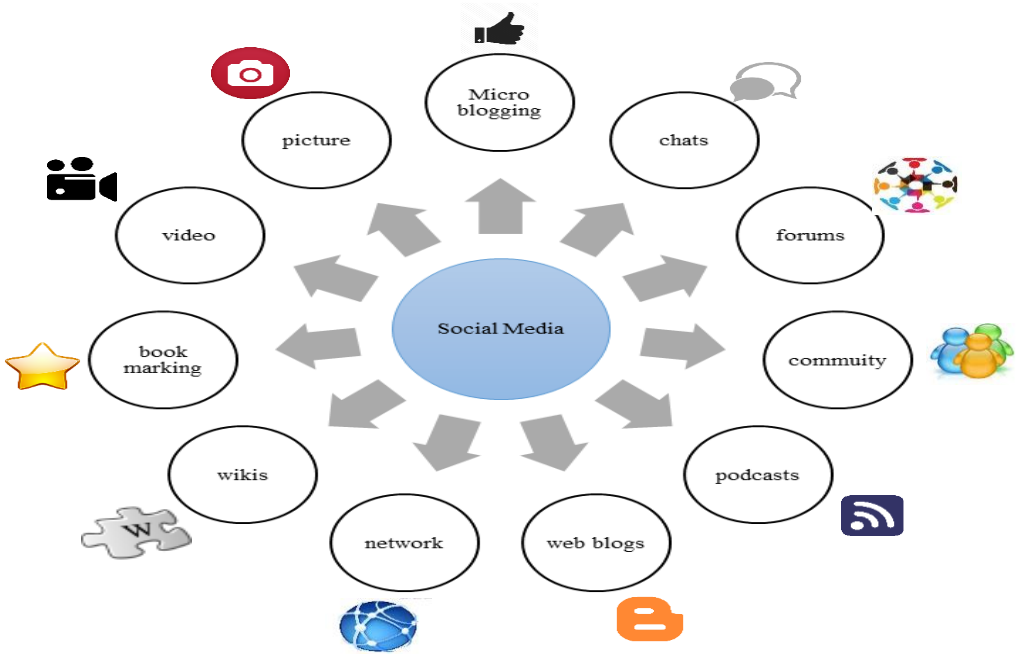

Fig.3. The social media features.

Table 3. Cumulative Analysis of Goal 1.

Q. No.	Strongly Disagreed	Disagreed	Neutral	Agreed	Strongly Agreed
1	0	1	6	12	34
2	1	6	13	20	13
3	1	2	6	22	21
4	0	1	5	27	20
Total	2	10	30	81	88
Avg.	0.5	2.5	7.5	20.25	22

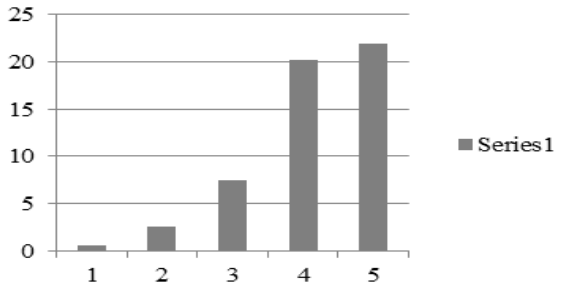

Fig.4. Cumulative analysis of goal 1.

Table 4. Cumulative Analysis of Goal 2.

Q. No.	Strongly Disagreed	Disagreed	Neutral	Agreed	Strongly Agreed
1	3	2	5	27	16
2	2	1	13	27	10
3	2	2	12	18	19
4	1	7	12	17	16
Total	8	12	42	89	61
Avg.	2	3	10.5	22.25	15.25

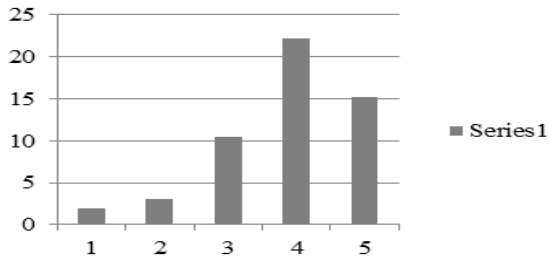

Fig.5. Cumulative analysis of goal 2

B. Goal 2- Teach the SECourse Using Cooperative Mastery Strategy

The questions set in this goal to evaluate the effectiveness of the use of cooperative mastery strategy to enhance student knowledge and the skills required in SE. Table 4 shows that 22.25% are agreed, and 15.25% are strongly agreed to use cooperative mastery strategy. 10.5% are responding neutrally. In addition, 3% are disagreed, and 2% have strongly disagreed. Table 4 results are presented in fig. 5.

C. Goal 3 - Effect of Using Social Media to Teach SE Course.

The questions in this goal for evaluating the role social media usage for teaching software engineering can improve software development skills for a student. Table 5 shows that 19.25% of the respondents are agreed, and 18.25% are strongly agreed. In addition, 9.25% are responding neutrally whereas 4.75% are disagreed and 1.5% of the respondents are strongly disagreed. Table 5 shows the result graphically in fig. 6.

Table 5. Cumulative Analysis of Goal 3.

Q. No.	Strongly Disagreed	Disagreed	Neutral	Agreed	Strongly Agreed
1	0	2	9	24	18
2	0	2	5	19	27
3	0	4	11	20	18
4	6	11	12	14	10
Total	6	19	37	77	73
Avg.	1.5	4.75	9.25	19.25	18.25

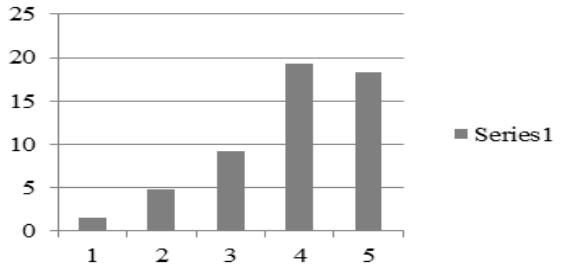

Fig. 6. Cumulative analysis of goal 3.

D. Goal 4- Establish the OSES Industry Experts to Improve Quality and Fill the Gap Between Industry and Academics

Four questions measure the effect of hiring industry experts to improve teaching SE. The results are displayed in Table 6. Table 6 shows that 21.75% of the participants are agreed In addition; the table is showed that 21.75% are strongly agreed whereas 6.25% of the respondents are responding neutrally. However, 2% of the participants are disagreed In addition, 1% of the respondents have strongly disagreed. Fig. 7 is presented results of goal 4.

Table 6. Cumulative Analysis of Goal 4.

Q. No	Strongly Disagreed	Disagreed	Neutral	Agreed	Strongly Agreed
1	0	3	8	19	23
2	0	3	5	30	15
3	2	1	6	21	22
4	2	1	6	17	27
Total	4	8	25	87	87
Avg.	1	2	6.25	21.75	21.75

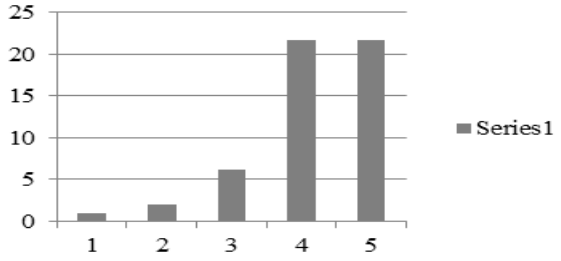

Fig.7. Cumulative analysis of goal 4.

VI. CONCLUSION

Most current methods of teaching SE are based on the theoretical education more than on practices. The majority of students are having a difficulty in the software industry and they lack the required skills to be recruited in good companies. In this paper, we proposed novel methods to address the previous issues based on four goals. First, we identify the objectives and the required skills of the SE course. Second, teaching the software engineering course is through using cooperative mastery strategy. These two goals will help to improve the quality of teaching software engineering, enhance student knowledge and the skills required to achieve determined course learning objectives (CLOs) in our approaches. Thirdly, using social media to teach SE course enhance awareness, learning and knowledge sharing. Finally, recruiting industry experts is recommended to prepare students for software development in industry and to fill the gap between industry and academics. The result of the survey indicated that most responders agreed with our proposed solution to improve the quality of SE teaching, and training students to prepare them for their future careers.

ACKNOWLEDGEMENTS

First, I am thankful to Allah to have completed this research. I am also thankful to my family for the support I repeatedly received. Special thanks to my supervisor Dr. Rizwan. He guided and supported to me throughout the journey of this paper. Least but not last, I am thankful to all those people who did help me to complete this research.

REFERENCES

[1] "Reducing the Gap between Software Engineering Curricula and Software Industry in Jordan - ProQuest." Available:http://search.proquest.com/openview/aaf510b1963600f7ae5cca23bcd890af/1?pq-origsite=gscholar. [Accessed: 14-Apr-2016].

[2] M. B. Blake, "A student-enacted simulation approach to software engineering education," *IEEE Trans. Educ.*, vol. 46, no. 1, pp. 124–132, Feb. 2003.

[3] D. Dahiya, "Teaching Software Engineering: A Practical Approach," *SIGSOFT Softw Eng Notes*, vol. 35, no. 2, pp. 1–5, Mar. 2010.

[4] L. Zhang and A. Hazeyama, "A Collaborative Learning Support System for Software Engineering Education," in *Proc. 6th Conference on Collaboration Technologies (Collabtech2012)*, 2012, pp. 134–137.

[5] "A Collaborative Approach to Software Engineering Education - ProQuest." [Online]. Available: http://search.proquest.com/openview/37066bc349be792bf2f185ad9689220a/1?pq-origsite=gscholar. [Accessed: 14-Apr-2016].

[6] P. Desai and G. H. Joshi, "Activity based teaching learning in software engineering - An experience," 2012, pp. 1–6.

[7] E. Choi, "Applying Inverted Classroom to Software Engineering Education," *Int. J. E-Educ. E-Bus. E-Manag. E-Learn.*, 2013.

[8] J. Ding, "A framework for global collaboration in teaching software engineering," in *Collaborative Teaching of Globally Distributed Software Development (CTGDSD), 2013 3rd International Workshop on*, 2013, pp. 30–34.
[9] K. Claypool and M. Claypool, "Teaching software engineering through game design," *ACM SIGCSE Bull.*, vol. 37, no. 3, pp. 123–127, 2005.
[10] O. Shata, "Teaching Software Engineering: A Critical Path Method," *Int. J. Comput. Commun. Eng.*, vol. 1, no. 2, p. 151, 2012.
[11] E. S. Grant and W. I. Helps, "Enhancing Teaching & Learning of Software Engineering in an International Environment," *GSTF J. Comput. JoC*, vol. 2, no. 1, 2014.
[12] T. Xie, N. Tillmann, and J. de Halleux, "Educational Software Engineering: Where Software Engineering, Education, and Gaming Meet," in *Proceedings of the 3rd International Workshop on Games and Software Engineering: Engineering Computer Games to Enable Positive, Progressive Change*, Piscataway, NJ, USA, 2013, pp. 36–39.
[13] T. Xiao-jun1, L. Ying1, and L. Na1, "Exploration and reformation of teaching methods for "Software Engineering," *Exploration and reformation of teaching methods for "Software Engineering" Course*, International Conference on Advanced Information and Communication Technology for Education), p. 4, 2014.
[14] M. Sepahkar, F. Hendessi, and A. Nabiollahi, "Defining Project Based Learning steps and evaluation method for software engineering students," *Int. J. Comput. Sci. Inf. Secur.*, vol. 13, no. 10, p. 48, 2015.
[15] A. Knutas, J. Ikonen, D. Maggiorini, L. Ripamonti, and J. Porras, "Creating software engineering student interaction profiles for discovering gamification approaches to improve collaboration," 2014, pp. 378–385.
[16] K. Tian, K. Cooper, and K. Zhang, "Improving Software Engineering Education through Enhanced Practical Experiences," 2011, pp. 292–297.
[17] A. Dagnino, "Increasing the effectiveness of teaching software engineering: A university and industry partnership," in *Software Engineering Education and Training (CSEE&T), 2014 IEEE 27th Conference on*, 2014, pp. 49–54.
[18] K. Berkling and C. Thomas, "Gamification of a Software Engineering course and a detailed analysis of the factors that lead to it's failure," in *Interactive Collaborative Learning (ICL), 2013 International Conference on*, 2013, pp. 525–530.
[19] M.-A. Storey, C. Treude, A. van Deursen, and L.-T. Cheng, "The impact of social media on software engineering practices and tools," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*, 2010, pp. 359–364.

Authors' Profiles

Dr. M. Rizwan Jameel Qureshi received his Ph.D. degree from National College of Business Administration & Economics, Pakistan 2009. He is currently working as an Associate Professor in the Department of IT, King Abdulaziz University, Jeddah, Saudi Arabia. This author is the best researcher awardees from the Department of Information Technology, King Abdulaziz University in 2013 and the Department of Computer Science, COMSATS Institute of Information Technology, Lahore, Pakistan in 2008.

Afra Abdullah Alabbadi is currently doing master in IT from Faculty of Computing & Information Technology, King Abdulaziz University, Saudi Arabia. She did graduation in Information System from Faculty of Computer & Information Technology Taif University, Taif.

How to cite this paper: Afra A. Alabbadi, Rizwan J. Qureshi,"The Proposed Methods to Improve Teaching of Software Engineering", International Journal of Modern Education and Computer Science(IJMECS), Vol.8, No.7, pp.13-21, 2016.DOI: 10.5815/ijmecs.2016.07.02