JOURNAL OF AERONAUTICS AND SPACE TECHNOLOGIES
(ISSN : 1304-0448)
July 2021 Volume 14 Number 2
jast.hezarfen.msu.edu.tr

Research Survey

A General Framework for Teaching Software Engineering and Improving Collaboration Skills in Multidisciplinary Teams

Dionysis GOULARAS [1], Mert ÖZKAYA[2*], Tacha SERIF [3], Sezer GÖREN[4]

[1] *Yeditepe University, Department of Computer Engineering, 34755 Ataşehir, Istanbul, Turkey*, goularas@cse.yeditepe.edu.tr, http://www.orcid.org/0000-0002-4802-2802
[2] *Yeditepe University, Department of Computer Engineering, 34755 Ataşehir, Istanbul, Turkey*, mozkaya@cse.yeditepe.edu.tr, http://www.orcid.org/0000-0002-3464-6489
[3] *Yeditepe University, Department of Computer Engineering, 34755 Ataşehir, Istanbul, Turkey*, tserif@cse.yeditepe.edu.tr, http://www.orcid.org/0000-0003-1819-4926
[4] *Yeditepe University, Department of Computer Engineering, 34755 Ataşehir, Istanbul, Turkey*, sgoren@cse.yeditepe.edu.tr, http://www.orcid.org/0000-0002-3688-5280
* *Corresponding Author*

Article Info

Received: *October 29, 2020*
Accepted: *June 1, 2021*
Online: *July 26, 2021*

Keywords: Software Engineering, Multidisciplinary projects, education.

Abstract

International Engineering Accreditation (IEA) organization measure and improve the quality of the engineering education in universities. This paper presents a framework for Software Engineering Education in line with the IEA criteria. A multidisciplinary project between computer engineers and designers is presented aiming to improve students' understanding of software processes and increase their communication skills in multidisciplinary teams. Computer engineering students follow specific software engineering processes, attend meetings and communicate via web-based project management tools. Their evaluation is made according to the IEA outcomes. As the results show, a multidisciplinary project structured according to specific IEA outcomes ensures students' skills in software processes and give them an overall satisfaction. The methods and procedures for developing collaboration skills in multidisciplinary projects are necessary. The creation of a profession-learning seminar improved student's communication skills thus showing that additional methods towards this direction need to be further designed and tested.

To Cite This Article: Dionysis Goularas, Mert Özkaya, Tacha Serif and Sezer Gören "A General Framework for Teaching Software Engineering and Improving Collaboration Skills in Multidisciplinary Teams", Journal of Aeronautics and Space Technologies, Vol. 14, No. 2, pp. 155-167, July 2021.

Yazılım Mühendisliğinin Etkin Öğrenimine ve Çok Disiplinli Takımlar İçinde İş-birliği Yeteneğinin Gelişimine Yönelik Genel Çerçeve

Makale Bilgisi

Geliş: *29 Ekim 2020*
Kabul: *1 Haziran 2021*
Yayın: *26 Temmuz 2021*

Anahtar Kelimeler: Yazılım Mühendisliği, Çok disiplinli projeler, eğitim

Özet

Enternasyonal Mühendislik Akreditasyon (IEA) organizasyonları, birçok üniversitede mühendislik eğitiminin ölçülüp kalitesinin onaylanmasına destek olurlar. Bu makalede, IEA kriterlerine göre etkin yazılım mühendisliği eğitimini hedef alan bir çerçeve önerilmektedir. Öğrencilerin yazılım süreçleri üzerine bilgilerini iyileştirmek ve çok disiplinli takımlarda iletişim becerilerini artırmak için, bilgisayar mühendisliği ile tasarım öğrencilerinin çok disiplinli proje kapsamında beraberce çalışması hedeflenmektedir. Bilgisayar mühendisliği öğrencileri çok disiplinli takımı kurarlar, yazılım mühendisliği süreçlerini takip ederler, toplantılara katılıp gerekli bilgi ve verilerin paylaşımını sağlarlar. Öğrencilerin performansı, IEA kriterlerine göre değerlendirilir. Sonuçlara göre, IEA kriterlerine göre yapılandırılan çok disiplinli proje ile, öğrencilerin yazılım süreçleri üzerine yetenekleri gelişmektedir. Takım içinde iş-birliğinin geliştirilmesi için bir takım metot ve prosedürlerin uygulanması gereklidir. Meslek eğitim seminerlerinin öğrencilerin iletişim beceresini iyileştirdiği ve böylece bu yönde metotların tasarlanıp test edilmesine gerek olduğu gözlemlenmiştir

1. INTRODUCTION

The ever-increasing demands of the customers and evolution of technology lead to the large and complex software systems developments that cannot be handled without applying the principles of engineering. In this context, Software Engineering education plays an important role for understanding the modern needs and adapting to the challenges of today's software. Because of the variety, the complexity and the constant change of software tools, the education of Software Engineering has attracted the attention of scholars who proposed adequate methods and procedures (e.g., [1], [2], and [3]) together with solutions for the adaptation to the modern business environment (e.g., [4]).

It is generally accepted that the program outcomes of the Software Engineering course cannot be properly achieved without an external collaboration, which either will expose the students to a business environment (e.g., [5]) or will include another or more than one department or an external organization. This collaboration usually happens in the framework of a multidisciplinary project and there are many studies that conducted research about this subject. As it will be discussed in the next section, those works have implemented a number of different methodologies and grading procedures thus showing that still doesn't exist a generally accepted and applied framework and grading method. Moreover, even if most of those studies have found that there are communication problems that could rise between the multidisciplinary teams, to the best of our knowledge no concrete steps have been taken in order to face this problem, apart from some general guidelines that encourage frequent meetings and further communication between teams.
As a result, the first motivation of this study is to show that multidisciplinary projects in software engineering help achieve the program outcomes of software engineering courses, thus encouraging all universities to include a multidisciplinary project in their software engineering syllabus. The second motivation is to propose a general framework for a software engineering course that includes a multidisciplinary project. This framework is based on the experience collected by the previous studies and also the collected experience of the authors. For this purpose, one of the key elements is the establishment of a standardized grading procedure. In this work we are proposing a grading method based on international accreditation criteria. Accreditation is a procedure allowing to measure, ensure and improve the quality of the education. It is carried out by national and international organizations and agreements all over the world. Despite the different standards available worldwide, all accreditation programs have a set of common goals and similar criteria. The Accreditation procedure guarantees the quality of programs and institutions and provide motivation and tools for continuous improvement. Including those criteria in the grading system will introduce a standard that can be easily followed by all universities worldwide.

Finally, the third motivation is to improve the collaboration skills of students when working in multidisciplinary teams. As it was mentioned above, the communication between teams was always a crucial element for the success of a multidisciplinary project. In order to achieve this goal, surveys should be conducted and concrete steps aiming to improve the communication between teams should be proposed.
In summary, this work will attempt to contribute to the literature by: *(i)* emphasizing the importance of including multidisciplinary projects in software engineering education for achieving the program outcomes; *(ii)* providing a complete road map for the organization of multidisciplinary projects that would include international accreditation criteria for grading; and *(iii)* establishing procedures aiming to improve the communication between multidisciplinary teams.
This study is divided into the following sections: Section 2 presents the background concerning the research on multidisciplinary projects in the field of software engineering. Section 3 presents the necessity and the accreditation framework used in this study for the evaluation of students based on specific program outcomes. Section 4 shows the methodology that is applied for the multidisciplinary project. Section 5 presents the organization of the project and section 6 displays the findings of this work. In Section 7 the results are discussed and finally, Section 8 concludes this study.

2. RELATED WORK

In a business environment, software application development requires the collaboration with professionals from different disciplines such as artists, designers, data scientists, content developers, etc. As a result, the software engineering course can teach a wide range of skills and talents [6] following different teaching methods (e.g., [7] and [8]) thus making it a very suitable candidate for including a multidisciplinary project in its curriculum, as it is also demonstrated in various studies such as [9, 10].

Collaboration between computer engineering and design students in multidisciplinary projects has been successfully achieved in the past. In [11] the authors emphasize on the cultural differences between the two disciplines and the respect for each other's expertise which is considered as the most important learning objective of this course. Respect as a quality and essential building block for a fruitful collaboration between computer engineers and designers have also been reported in [12]. The importance of integration of usability and user requirements with agile development was shown in [13]. Trust in virtual environments is also an important aspect as presented in a systematic review in [14]. In another study [15] the authors analyze the effects of different relations between the teachers in

interdisciplinary teams from health professions. In a recent study [16] a comparative literature review is presented in order to illustrate the differences in multidisciplinary collaboration studies. Finally, it is worth to mention the importance of multidisciplinary teams in the domain of Artificial Intelligence [17].

In general, the use of game development has been successful for multidisciplinary projects in software engineering. Actually, as it has been shown in [18] and [19], implementing a game is an attractive subject where all aspects of a software lifecycle can be represented.

There are a number of studies related with multidisciplinary projects in the domain of Software Engineering. In [20], Yuen describes the design of a summer capstone course made through cross-departmental collaboration using an agile development method (SCRUM). The course used a service learning approach where 10 projects were selected in collaboration with the educational faculty after the formation of ten teams with five students each. Sprint planning was organized followed by a final poster style presentation including project demos. The author focused on the difficulties of developing the projects in the shortened summer time period. Grading was made based on burndown chart and conditions of satisfaction. Working with the educational faculty had the advantage of having more feasible projects and the disadvantage of missing the collaboration with an industry client. In this paper the collaboration department plays the role of a client rather than a collaborating partner. In another work [21], Bidarra *et al.* describe the development of a game application between students of a department of Computer Engineering and students pursuing a degree of Game Design and Development in a School of Arts. An agile development process has been used, and the requirements are predefined in order to guarantee that specific type of graphics and AI techniques were to be included. An easy to use custom graphic engine was developed helping to overcome technical difficulties that could arise. The grading takes into account both programming and collaboration performance. The paper focuses on the difficulties encountered in the collaboration and they have proposed tools for facilitating the communication between students. This study involves second year students and it intends to prepare students for software process activities rather than integrate those projects into the scope of a software engineering course. In [22], Dorner and Spierling describe the development of serious game in the scope of an interdisciplinary teaching environment. The creation of four serious games is presented with the participation of students from computer engineering, media design, media informatics and media management departments. The steps under taken during the development of these games were supervised by the professors and the Scrum framework was followed. According to the authors, the design of a serious game is fruitful and at the same time demanding. As a result, in this study, the formation of big teams (20 persons) was required together with the management by professors and a possible participation of external stakeholders thus making it difficult to include it in the framework of a multidisciplinary project inside a software engineering course.

In [23], Maxim and Ridgway present the results of interdisciplinary projects in game development between students of a computer and information science department and an animation and digital media department using a two-semester course sequence about game design. The projects were organized in order to satisfy the ABET requirements for interdisciplinary work, but a grading procedure based on program outcomes was not applied. Students are organized in teams of 8-10 from both departments and milestones were defined for the completion of the project. In the students observations it was mentioned among others that good coordination and communication and face-to-face meetings together with proper planning is the key for successful projects. This study, as expected, was mainly focused on the challenges of the different phases of a game development rather than on a generalized perspective on the software engineering processes. In [24] Schaetter *et al.* incorporate the development of large projects in the curriculum of a department of Industrial Engineering and Information technology aiming to prepare students to real world conditions. Separate teams of 3-4 students supervised by professors are collaborating by taking different phases of a project (requirements, analysis, development, etc.). In this work the interdisciplinary collaboration includes only students from the engineering faculty. In [25], Vicente *et al.* focus on the collaboration of software engineering students with students from different educational backgrounds. In this study, collaboration between the programs of Computer Science, Business Management and Product design was established to work on projects with real clients. This work used the case method approach and Scrum methodology was applied. 10 teams of 10 students from the three departments were formed and the syllabus of three courses from the above departments was aligned in order to create interdisciplinary projects in a collaborative framework. Public ports and terminal organizations were invited to be the clients. In this study, the assessment was done only by peer evaluation between student members and by surveys. Another work [26] addresses the problem of collaboration between students of IT and other disciplines (accounting, art, etc.). In the scope of software engineering education, three teams' projects are compulsory in different semesters. The first two are oriented on software engineering processes and the third on interdisciplinarity goals. For three years 30 students from IT and Art disciplines are divided into 6 teams of 5. Students based on a general theme are

allowed to choose their project subject and they had to deliver a product report and a process report counting 60% and 40% respectively. Every year different clients were chosen and code was developed according to the number of CS/SE students in the class. The findings show that interdisciplinary skills and self-insight are developed more compared with a traditional software engineering team project. As those projects are part of a general interdisciplinary course including 1500 students from other disciplines a general aspect of interdisciplinarity is given where the final product depends on the number of CS/SE students that participate on it. It should be emphasized at the point that there are also studies that have different perspectives on Software Engineering – such as the work presented in [27], which entails the development of large-scale systems with teams from all over the world in an Internet-based groupware environment or giving special emphasis on project meetings [28]. From the above studies, it can be seen that in this domain a significant work has been done with important contributions. A summary of those works in terms of methodological issues like software process, teams, grading method etc. can be found in the results section (Table 4) when the current study will be compared with the above ones. Despite the high standards and the quality of those studies, a complete framework in terms of methodology, organization, and grading that could also be served as a general roadmap for multidisciplinary projects is missing, partly because every study focuses on different aspects. Moreover, even if most of the studies above emphasized the need for more communication in the team members, in most of cases no concrete solutions are proposed with the exception of some general guidelines or some names of communication tools. This study, based on the experience gained during 3 consecutive years of software engineering course, contributes to the aforementioned subjects by proposing a solid multidisciplinary project framework and performs a concrete step for enhancing the communication between multidisciplinary teams. The next section presents the accreditation framework used in this work and its necessity for evaluating the students' performance.

3. ACCREDITATION FRAMEWORK

Today, due to the important number of colleges and universities worldwide, as mentioned before, there is a constant need for the program administrators to measure, ensure and improve the quality of the education given in the engineering faculties. This overall procedure is called accreditation and is carried out by national and international organizations and agreements all over the world. Despite different standards that exist worldwide, accreditation programs have all common goals and similar criteria and they are one of the methods that guarantee the quality of programs and institutions and provides motivation and tools for continuous improvement.

The accreditation process has a long history and it is applied successfully worldwide. In United States, the Accreditation Board for Engineering and Technology (ABET) program, established in 1932 after its predecessor, the Engineers' Council for Professional Development (EPCD), aims to provide the quality assurance in engineering education [29]. ABET is used worldwide and has accredited colleges and universities in many countries. In Europe, the European Network for Accreditation of Engineering Education (ENAEE) founded in 2006 ensures and supports the EUR-ACE system [30]. At the same time, international agreements between bodies, like the Washington accord signed in 1989, allow the mutual recognition between signatories and the collaboration between countries [31]. In Turkey, the accreditation for engineering programs is carried out via the Association for Evaluation and Accreditation of Engineering Program (MUDEK), a non-governmental organization [32]. MUDEK aimed to comply with both American and European accreditation criteria. It was established in 2003 and today is authorized by ENAEE to grant EUR-ACE label. At the same time is a member of both ENAEE and Washington accord, thus making MUDEK label satisfying both US and European accreditation standards.

The implication of engineers in various and heterogeneous domains requires their collaboration with professionals from different fields. Therefore, the ability to cooperate in different multidisciplinary contexts is considered as an important element of their education. Consequently, multidisciplinary projects are included for ABET and EUR-ACE accreditation and in the past specific models were proposed for the evaluation of multidisciplinary projects in their guidelines [33]. In this framework, an academic curriculum that conforms to the above accreditation criteria requires at least one course to integrate a multidisciplinary project in its syllabus. Therefore, as MUDEK complies with the above requirements, in Turkish universities multidisciplinary projects should be included in the curriculum of the engineering departments. For a computer engineering department, as is been demonstrated before, the software engineering course is one of the best candidates.

4. METHODOLOGY

The multidisciplinary project in software engineering course was conducted for three consecutive years (2017, 2018, and 2019). Two departments have participated, namely the Department of Computer Engineering (CSE) and the Department of Visual Communication Design (VCD). The methodology in terms of data collection, tasks, software process, communication, roles, game application and project reporting is described below.

4.1. Data Collection

The data were collected by: *(i)* the students through

surveys that were conducted in the end of the semester; *(ii)* their interaction with web-based communication tools; *(iii)* personal discussion; *(iv)* their reports; and *(v)* the final application games, presentations and grades.

4.2. Tasks

For the creation of the game app, each VCD group – which consists of 2-3 students - gives the original idea to create the scenario of the game and prepares the graphics of the game. Then, each VCD group is matched with a CSE group – which consists of 4-6 students - whose responsibility is to develop the game app using the description and graphics provided by their partnering VCD group..

4.3. Software Development Process

After the initial specifications are set and planning is made, students applied a particular software process. More specifically, because of the short life span and small size of the project in terms of time and size, the agile development was selected for all of the projects, which is iterative and incremental model/approach.

Following this approach, the students then could start with small prototypes and revise their initial specifications and planning. Gantt charts also were used and revised even if their use in such a developing environment is debatable.

4.4. Communication

The communication between members of the CSE and VCD department has been facilitated with a web-based project management application [34] and complemented with e-mails and smartphone messaging tools. In every group the CSE students, after collaborating with the members of the VCD department, had to define the specifications of the game, create a Gantt chart and record the progress of the project.

For the CSE students the use of the web-based project management tool application is mandatory. This is because apart from establishing the specifications of the project, its planning and the communication between the members of the team, it allows the instructors to follow the project and the level of participation of team members.

4.5. Roles

The CSE students' roles were specified as: project manager, programmer, writing the report or liaising with the VCD members via messaging, email or phone calls. Every two weeks a meeting between the VCD and CSE members of each group is organized in order to share information, ideas and other subjects related with the project.

4.6. Game Application

The final product of the project is a game application. The game application is required to run on a smartphone. In every team 2-3 developers are responsible for writing the code and testing the application. The choice of the developing platform is selected by the CSE team members. Finally, a CSE member is responsible for communicating with VCD team and take the graphic elements that will be integrated to the game.

4.7. Reports

The reports are a critical part of the project and in general constitute an important part in the professional activities of an engineer. Each CSE group is expected to submit two reports throughout the course, which require them to document the requirement and design specifications of their game applications. Unified Modeling Language (UML) [35] is employed by the CSE students for creating requirements and design models. Note that the course instructors use UML as a medium for teaching CSE students how to specify requirements and design models. Firstly, an analysis report is submitted in week 6, which describes the requirements of the game application in a systematic and organized way. CSE groups initially analyse their domain with which the game application is relevant and draw a UML class diagram for specifying the domain concepts and their relationships. The CSE groups use UML's use-case diagram and use-case specifications to specify the functional requirements. Also, the CSE groups use *Volere* [37] template to specify the non-functional requirements in a precise and measurable way. Following that, a design report is submitted in week 11, in which the CSE groups specify how to implement the requirements from different viewpoints such as structure, behavior, and interaction. CSE groups use several UML diagrams for design purposes, such as the UML class component diagram for the structural modeling, UML sequence and activity diagrams for the interaction modeling, and UML state diagram for the behavior modeling.

5. PROJECT ORGANIZATION

The software engineer course is offered once every year and the challenge is to organize around 100 students of two different departments that may have different objectives but are sharing a common goal. The organization requires groups to be formed, a kick-off meeting to be organized and milestones to be defined:

5.1. Groups

In every term 10 groups are created and each of them is composed with 4-6 CSE students and 2-3 VCD students (Figure 1). As is been mentioned in the methodology part, in every team, the CSE students are assigned with distinct responsibilities such as project managers, programmers, report writers, etc.

5.2. Kick-off Meeting

In the beginning of the semester a kick-off meeting is organized in order to allow the formation of multidisciplinary teams. The formation of the

multidisciplinary teams is done during this meeting, which is held in a casual atmosphere. CSE and VCD members have the opportunity to meet each other and share ideas and discuss about the project.

At this stage, CSE and VCD students are already grouped within their corresponding departments and ready to discuss, exchange ideas and views with their counterparts to match and form/create a multidisciplinary team. The result of the kickoff meeting is the creation of 10 teams with a game application idea for each of them as it is displayed in Figure 1.

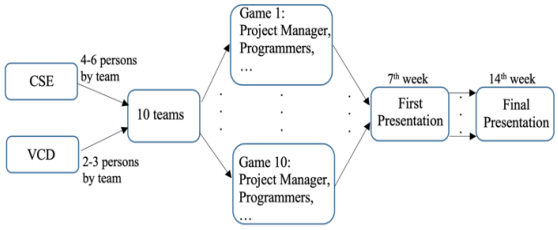

Figure 1. Organization of the multidisciplinary projects.

5.3. Milestones

Two milestones are set; one is in the middle of the semester and another one at the end as shown in Figure 1. In the first, every group is required to present their ideas orally in front of the instructors and the other teams.

The CSE and VCD departments form 10 teams where the CSE members have distinctive responsibilities. For each team, the objective is to create a computer game for smartphones. Their work is evaluated with the international accreditation outcomes. Two milestones namely First Presentation and Final Presentation complete the procedure.

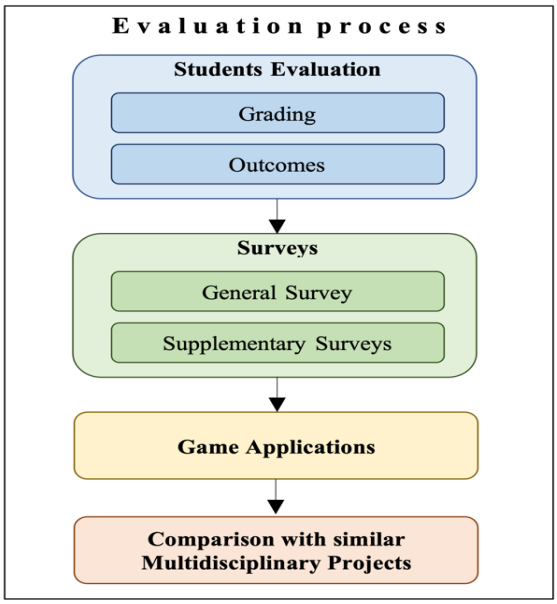

Figure 2. Diagrammatic representation of the evaluation process.

During this presentation, the VCD members should present the basic scenario and some drawings related with the characters and the scene of the game. The CSE students are required to present the specifications, the planning and the analysis of the project using text, Gantt charts, flowcharts and UML diagrams. In the second and final milestone, a final presentation in a large lecture theatre is made, that gives the opportunity to the teams to present every aspect of their work – including design, implementation, graphics and final application demo – to all teams, students and academics.

6. EVALUATION AND RESULTS

This section presents the findings of this study with regards to the outcomes, gradings, surveys, final applications and comparison with related studies. In general, most of the game applications were satisfactory and the gradings revealed that the students performed well and applied with success the software engineering skills. Finally, the surveys showed the general satisfaction for the project procedures and put the light on communication issues between multidisciplinary teams. Figure 2 illustrates the different evaluation steps.

6.1. Students Evaluation

The student evaluation was based on IEA outcome criteria and individual grading when this was possible, as it will be explained below.

6.1.1. Outcomes

The accreditation process requires the satisfaction of specific outcomes, which is related with the ability of the students to acquire a particular knowledge through the curriculum of the course. These outcomes are covering the activity of the whole course as they are fulfilled also by the assignments, midterm and final exam. Nevertheless, an important part of them is related with the multidisciplinary project. Table 1 presents the IEA learning outcomes of the software engineering course. IEA outcomes number 8 and 9 are not relevant with this course and for this reason are not mentioned in the table. We can note that for the software engineering course program outcomes, the multidisciplinary project is related with five out of the eight IEA outcomes. In brief, the project improves the ability to design a complex system, use modern techniques and tools, gather data, design and conduct experiments, communicate effectively orally and writing, work individually and inside a team and prepare design and production reports.

6.1.2. Grading

An important part of the accreditation procedure is the individual grading for each IEA outcome. Bearing in mind this, every student should be graded according to these outcomes. However, the biggest issue encountered during this procedure was the difficulty to

Table 1. The international engineering accreditation learning outcomes of the software engineering course as defined by MUDEK.

Program Learning Outcomes	**Related action in Software Engineering Course**
1. Adequate knowledge in mathematics, science and engineering subjects pertaining to the relevant discipline; ability to use theoretical and applied knowledge in these areas in complex engineering problems.	Assignment
2. Ability to identify, formulate, and solve complex engineering problems; ability to select and apply proper analysis and modeling methods for this purpose.	Midterm question
3. Ability to design a complex system, process, device or product under realistic constraints and conditions, in such a way as to meet the desired result; ability to apply modern design methods for this purpose	***Project design and project implementation***
4. Ability to devise, select, and use modern techniques and tools needed for analyzing and solving complex problems encountered in engineering practice; ability to employ information technologies effectively.	***Collaboration tools / Web-based project management tools.***
5. Ability to design and conduct experiments, gather data, analyze and interpret results for investigating complex engineering problems or discipline specific research questions.	***Project tests & experiments***
6. Ability to work efficiently in intra-disciplinary and multi-disciplinary teams; ability to work individually.	***Project collaboration***
7. Ability to communicate effectively in Turkish, both orally and in writing; knowledge of a minimum of one foreign language; ability to write effective reports and comprehend written reports, prepare design and production reports, make effective presentations, and give and receive clear and intelligible instructions.	***Project report***
10. Information about business life practices such as project management, risk management, and change management; awareness of entrepreneurship, innovation, and knowledge about sustainable development.	Lectures and final questions

*Outcome actions related with the multidisciplinary project are marked with bold italic.

evaluate the individual performance of each student inside a team.

As applying the outcomes to each student separately is an IEA requirement and needs time and effort, one of the solutions proposed to this matter, together with the average performance of the team, was to take into account their particular role inside the project: report writing, liaising between teams, programming, etc.

Figure 3 presents the grades of the program outcomes 3-7 related with the multidisciplinary project that are described in Table 1. As it can be shown, in the three terms the students performed above average. As the study has more than two samples, a Kruskal-Wallis test was performed for calculating the H value (equation 1). In this equation N is the total number of samples, n_i is the number of samples for each group k, and (r_i)¯ is the average rank for each group i. The results showed that there is no statistical difference between the threeyears grades (Chi square = 2.41, p = .48, df = 14). We can notice from figure 3 that students' performance was high in project design, collaboration, implementation and tests and slightly lower in the use of collaboration tools and writing the reports.

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} n_i \bar{r}_i^2 - 3(N+1) \qquad (1)$$

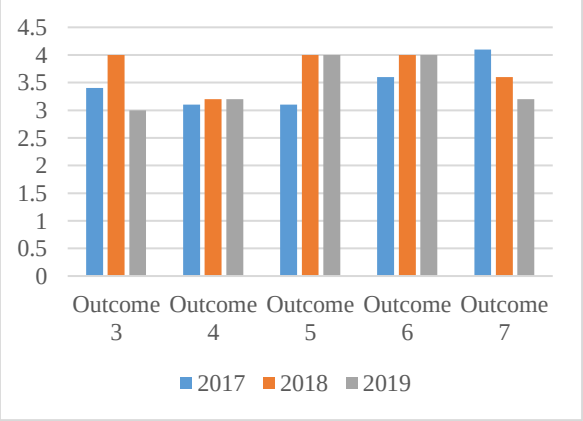

Figure 3. Average grades of the accreditation outcomes 3-7 related with the multidisciplinary project.

6.2. Surveys

In order to get a feedback from the students concerning their multidisciplinary experience, a series of questions were asked using the Likert scale [36]. The aim of the questions was to investigate if the project was beneficial in terms of software processes learning andalso to examine the collaboration of CSE students with students of the VCD department. The answers concerning the communications between multidisciplinary teams attracted our interest.

Table 2. Findings of a survey conducted after the end of each term.

Students Questionnaire			
	Mean (2017)	**Mean (2018)**	**Mean (2019)**
Overall Project experience			
1. Multidisciplinary project was a new experience for me	4.3	4.1	4.2
2. I would like to take part in a multidisciplinary project again in the future	4.4	3.8	4.1
3. I have experienced and learnt new methodologies to solve problems throughout a project lifecycle	4.5	4.2	4.0
4. This experience improved my understanding of project management	4.5	4.5	4.2
5. This experience improved my understanding of possible project development issues	4.1	4.5	4.5
Software processes			
6. I am happy with the initial project plan (Gantt Chart) of our project	3.8	3.8	3.8
7. I am happy with the software analysis of our project.	4.1	3.8	4.1
8. I am happy with the software design of our project.	4.1	3.9	4.3
9. I am happy with the toolset that I have learnt in the development our project.	4.5	4.0	4.4
10. After taking part in this project, I am confident in evaluating and testing software systems using unit and usability testing techniques.	3.8	3.8	3.9
Intra and multidisciplinary issues			
11. The planning and development of this project improved my interpersonal communication skills.	3.9	3.9	**4.2**
12. This was one of the worst experiences that I have ever encountered.	2.1	2.5	**1.7**
13. Your collaboration with VCD was fruitful.	2.8	3.0	**3.5**
14. We as (the CSE team) had communication problems.	2.2	2.5	**2.0**
15. We as (the multidisciplinary team) had communication problems.	3.0	3.0	**2.4**

*****The results are the mean values of the answers of all students in likert scale.

As a result, in the second term supplementary questions were asked aiming to investigate further the relation between students of the two departments. The findings revealed the need for a profession profile-learning seminar which was conducted the third year. According to the findings of the surveys conducted in the third year, this seminar was accepted well and evaluated as very positive by the students as it will be explained in detail below.

6.2.1. General Survey

Table 2 displays the results of a survey given to students during the three terms. A Kruskall-Wallis test shows that there is no statistical difference between the answers of students between the three terms (Chi square = 26.5, p = .97, df = 44). Moreover, the answer results concerning the project design, development and understanding of software processes were high and in accordance with the outcomes grade results. According to the answers of CSE students, it is clear that they perceived the overall project experience very positively. They affirmed that they have learnt new methodologies and acquired an understanding in project management. At the same time they were satisfied by the way they applied the required software processes. One other element that arises from the survey and personal discussions is the lack of motivation in some VCD groups. The explanation given was that the tasks and the amount of work of VCD students were not in balance with the requirements that CSE students had. From the point of view of CSE students, a successful multidisciplinary project organization requires a balanced workload, grading system and penalties in case of project failure between students of different departments.

6.2.2. Supplementary Surveys Related With VCD Profession

Concerning their relation of CSE students with the VCD department, as we have observed that their level of satisfaction was lower compared to the other answers, a supplementary survey has been conducted in the two consecutive years in order to investigate this issue.

Table 3. Supplementary survey (2018, 2019) related with multidisciplinary issues given only in the second consecutive term.

Investigation about a profession profile learning seminar	**Mean (2018)**
1. After having finished the multidisciplinary project, would you like to learn more about the profession of your collaborators of the VCD department?	3.2
2. Would you wish to know more about the profession of your collaborators before the multidisciplinary project starts?	3.7
3. Do you think that knowing more about their profession could increase the quality or the success of the multidisciplinary project?	4.1
4. Do you believe that knowing their level of computer knowledge could facilitate more the communication with them?	4.0
5. Do you think that it would be useful that your collaborators should know more about your way of working for an multidisciplinary project?	4.2
6. Are you positive about establishing a profession profile learning of your collaborators before an multidisciplinary project starts?	3.9
Profession profile learning seminar results	**Mean (2019)**
1. The profession profile learning seminar in the beginning of the semester was useful for the project?	4.3

Table 3 presents the results of this supplementary survey made in 2018 and 2019. From the first part in 2018 it can be seen that CSE students didn't know enough about the functionalities of the profession of a designer. Moreover, they were positive about the idea of getting this missing information before the multidisciplinary project started. For this reason, in the third year, a one-day profile-learning seminar was held in order to orient students about the nature of the design profession, the problems that could arise and the ways to be solved. Special emphasis was given to the relationships and the behavior that engineers should have in terms of respect and understanding. After the end of the third year and the end of the project, students were asked to evaluate the utility of this seminar. Their answers were very positive as it can be seen in Table 3. Moreover, as we can see in the previous table, Table 2, in the survey of 2019 concerning the answers related with the intra and multidisciplinary issues (marked with bold blue numbers), the communication between CSE and VCD teams is improved compared to the two previous years thus showing the importance of the profession learning seminar held in the beginning of the project.

Figure 4. Term project welcome screen images

6.3. Game Applications

According to the multidisciplinary project organization the game applications are presented during a final presentation in front of all teams. During the two terms the games had different genres: action, survival, vehicle simulation, educational. Most of them were created with the help of game development platforms (e.g., Unity) and some others were developed using some common programming languages (e.g., Java and C#). The overall quality was satisfactory in terms of scenario, game complexity and design, but given the restricted time period during which students had to finalize the work, they should be considered more like the game term projects rather than usual commercial games. Figure 4 shows some welcome screens from these games.

6.4. Comparison with Similar Multidisciplinary Projects

In order to further evaluate the current work, we did a comparison with other similar studies according to criteria like the grading system, the type of the multidisciplinary teams, the software process, the duration of the project courses and the management of the communication issues among the teams. According to the findings, this study presents two major advantages: first, it applies a grading system compatible with ABET and EU-ACE criteria based onspecific outcomes that attempts also to grade individually the students. Second, it includes a profile-learning seminar aiming to present the other discipline that students will collaborate in order to reduce the communication problems that may occur, due to lack of information about the 'other'. As it was mentioned in many of the studies presented in Table 4, one of the major challenges was to establish a communication between teams.

Table 4. Comparison of the current study with similar multidisciplinary works

	This study	**Yuen** [16]	**Bidarra *et al.*** [17]	**Dörner & Spierling** [18]	**Maxim & Ridway** [19]	**Schaetter *et al.*** [20]	**Vicente *et al.*** [21]	**Jaccheri & Sindre** [22]
Outcomes/ Grading based on outcomes	Yes	No [1]	No [2]	No	No	No	No [3]	No
Multidisciplinary collaborative teams	Yes	No [4]	Yes [5]	No [6]	Yes	No [7]	Yes	Yes
Software process	Agile	Scrum	Agile	Scrum	-	Spiral	Scrum	- [8]
Game application	Yes	No	Yes	Yes	Yes	No	No	No
Information/Education about the other team	Yes	No	No [9]	No [9]	No [9]	No	No	No
3 or more consecutive years of project course	Yes	No	No	No	No	Yes	No	No

[1] Burdown chart, conditions of satisfaction, [2]programming collaboration performance, [3] peer evaluation, [4] One team was acting as a client, [5]but 2nd year students, [6] all teams are supervised by professors, [7] only from engineering faculty, [8] No specific process is imposed to the teams [9] the need for communication between teams is mentioned

7. DISCUSSION

The program outcomes of a software engineering multidisciplinary project aim to improve the following: (i) the ability to apply the software processes when collaborating with people from another discipline and; (ii) the improvement of multidisciplinary communication and their capacity to exchange information successfully. In this scope, the experience of three years allowed to have some interesting findings and learn important lessons.

7.1. Summary of Findings

Multidisciplinary framework works well for learning software processes: From students' surveys and grades, this framework guaranties that the students are learning the program outcomes. This was expected according to the literature and through this study the importance of a multidisciplinary project in software engineering is confirmed one more time.

Individual grading for international accreditation outcomes is not always easy: For some program outcomes the assignments, midterm and final grades help to evaluate student's performance. But the outcomes where students are acting together, like the communication with the other team or programming tasks are more difficult to evaluate.

The lack of balance in workload between departments creates some unease: From personal discussions CSE students frequently reported about the low motivation of some of VCD students and the high workload that the CSE students had. The CSE students said that the penalties and the motives in general should be balanced in order to have uniform groups in terms of motivation.

Communication between multidisciplinary teams is the main challenge: According to surveys, the main issues are related with the communication between multidisciplinary teams. This finding is also highlighted in previous studies and some authors emphasize the need for more communication.

7.2. Lessons Learned

Balancing the workload between departments: This task requires a particular effort to the part of the instructors of different departments in order to adapt their curriculum accordingly. As the goal is common, a communication and conversation between instructors is necessary.

Adapting the grading procedure for individual grading: The international accreditation program outcomes provide a guarantee for the measurement of the students' ability to satisfy the related criteria. When individual grading is difficult, the challenge is to avoid increasing the complexity of the grading system by taking concrete and simple actions. For example, in the outcome 7 of the Table 1 related with the project report, individual grading can be done by assigning specific chapter(s) to be written by a single student. But for other outcomes like the outcome 6 related with the multidisciplinary performance, an evaluation based on team's performance might be followed.

Seminars about the 'other' and continuous information are necessary for improving the communication between multidisciplinary teams: The second survey focusing on communications issues showed that CSE students wished to have a priori information about the other team before the project starts. As a result, in the

beginning of the third year of the multidisciplinary project, the seminar course that was held created positive reactions. The survey after the end of the term showed that they felt much more prepared for their first contact with the VCD teammates. Eventually, the most importance lesson to be drawn from this experience is that informative seminars about the 'other' should be considered for the curriculum of courses including a multidisciplinary project.

8. CONCLUSION

In this study, we presented the findings of a multidisciplinary project between students of Computer Engineering Department and Visual Communication and Design Department at Yeditepe University. This project was conducted during three consecutive terms as part of the Software Engineering course.

The study was focused mainly on CSE students and for the evaluation of students an international accreditation system was applied, which is compatible with ABET and EUR-ACE accreditation standards. The results showed that the multidisciplinary project in combination with the accreditation procedure is beneficial for the students for acquiring the knowledge of software engineering processes because they provide a concrete framework that guarantees the learning and evaluation procedure.

On the other hand, the findings also showed that further effort should be given in order to improve the effectiveness and the communication between students of different disciplines and also organize an equally balanced project between different departments in terms of work load, motivation and penalties. Furthermore, according to the comments and suggestions of students, a special attention should be given to inform about the profession of the 'other' discipline before the multidisciplinary project starts. Entering this information into the curriculum of a course that contains multidisciplinary projects could prepare students for real life condition projects because it will introduce a useful professional attitude. This behavior could be beneficial because it will make them more open to other professions, and behaviors in general, that an engineer doesn't see when he is not involved in multidisciplinary projects. Being more communicative, aware of the other and open-minded was and continues to be a major challenge for engineers and as it was proven to be true as a result of this study too. Those findings should be guideline in tailoring and shaping up the new educational aims and objectives.

In summary, the contribution of this study compared to similar ones lies on the proposal of a complete framework, including elements like the aforementioned grading procedure and the information about the other disciplines where their importance is still not highlighted enough.

9. ACKNOWLEDGMENTS

The authors would like to thank Assoc. Prof. Dr. Neda Ucer and Yasemin Ulgen Muluk of the Visual Communication and Design department for their overall collaboration, suggestions and very useful feedback that have provided us during the three terms of this multidisciplinary project.

10. REFERENCES

[1] M. Shaw, "Software Engineering Education: A Roadmap", in *Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland June 4-11, 2000*, pp. 371–380.

[2] C. Ghezzi, and D. Mandrioli, "The challenges of software engineering education", in *Proc. - 27th Int. Conf. Softw. Eng. ICSE05, St. Louis, Missouri, USA, May 15-21, 2005*, pp. 637–638.

[3] M. Jazayeri, "The Education of a Software Engineer", in *Proceedings of the 19th IEEE İnternational Conference on Automated Software Engineering, Linz, Austria, September 24-24, 2004*, pp. 18–xxvii.

[4] N. M. Devadiga, "Software Engineering Education: Converging with the Startup Industry", in *Proceedings - 30th IEEE Conference on Software Engineering Education and Training, CSEE and T 2017, Savannah, GA, USA, November 7-9, 2017*, pp. 192–196.

[5] T. J. Reichlmay, "Collaborating with industry: strategies for an undergraduate software engineering program", *in Proc. 2006 Int. Work. Summit Softw. Eng. Educ., Shanghai, China, May 20, 2006*, pp. 13–16.

[6] M. Gnatz, L. Kof, F. Prilmeier and T. Seifert, "A practical approach of teaching Software Engineering", in *Software Engineering Education Conference, Proceedings, Madrid, Spain, March 20-22, 2003*, pp. 120–128.

[7] D. Dahiya, "Teaching Software Engineering : A Practical Approach", *ACM SIGSOFT Softw. Eng. Notes*, vol. 35, pp. 1–5, March 2010.

[8] A. Sureka, M. Gupta, D. Sarkar and V. Chaudhary, "A Case-Study on Teaching Undergraduate-Level Software Engineering Course Using Inverted-Classroom, Large-Group, Real-Client and Studio-Based Instruction Model", *www.arxiv.org*, September 2, 2013 [Online].Available: https://arxiv.org /abs/1309.0714 [Accessed: Sept. 29, 2020]

[9] A. Zeidmane, and S. Cernajeva, "Interdisciplinary approach in engineering education", *in 2011 IEEE Global Engineering Education Conference, EDUCON 2011, Amman, Jordan, April 4-6, 2011*, pp. 1096–1101.

[10] D. Rosca, "Developing teamwork and communication skills in a multidisciplinary experiment", in *Frontiers in Education Conference,*

FIE, 2003, Westminster, CO, USA, November 5-8, 2003 pp. S4C-14-17.

[11] S. Martin and M. Fred, "A dual-major course emphasizing computer science and graphic design", *J. Comput. Sci. Coll.*, vol. 30, pp. 24–31, June 2015.

[12] P. L.Li, A. J. Ko and A. Begel, "Cross-disciplinary perspectives on collaborations with software engineers", in *Proceedings of the 10th International Workshop on Cooperative and Human Aspects of Software Engineering, Buenos Aires, Argentina, May 23, 2017*, pp. 2–8.

[13] K. Curcio, R. Santana, S. Reinehr and Malucelli, "Usability in agile software development: A tertiary study", *Computer Standards & Interfaces*, vol. 64, pp. 61-77, May 2019.

[14] J.V. Hacker, M. Johnson, C. Saunders and A.L. Thaye, "Trust in virtual teams: A multidisciplinary review and integration", Australasian Journal of Information Systems, vol. 23, January 2019.

[15] Q. T. Nguyen, and C. Mougenot, "Dimensions of multidisciplinary collaboration: a comparative literature review within design context". *In Proceedings of the Design Society: DESIGN Conference, May 2020* , pp. 1335-1344.

[16] S. Meeuwissen, W. H. Gijselaers, I. Wolfhagen, and M. Oude Egbrink, "How teachers meet in interdisciplinary teams: hangouts, distribution centers, and melting pots", *Academic Medicine*, vol. 95, pp. 1265-1273, August 2020.

[17] R. Kusters, D. Misevic, H. Berry, A. Cully, Y. Le Cunff, L. Dandoy, N. Díaz-Rodríguez, M. Ficher, J. Grizou, A. Othmani, and T. Palpanas, "Interdisciplinary Research in Artificial Intelligence: Challenges and Opportunities", *Frontiers in Big Data*, vol. 3, pp. 45-51, November 2020.

[18] K. Claypool and M. Claypool, "Teaching software engineering through game design", *ACM SIGCSE Bull.*, vol. 37, pp. 123–127, June 2005.

[19] I. Miladinovic and S. Schefer-Wenzl, "Pawing the Way for Inter-disciplinary Learning in Software Development Education", in *Proceedings of the 16th World Conference on Mobile and Contextual, Learning, Larnaca, Cyprus, October 30-November 1 2017*, pp. 1-2.

[20] T. T. Yuen, "Scrumming with educators: Cross-departmental collaboration for a summer software engineering capstone", in *Proceedings - 2015 International Conference on Learning and Teaching in Computing and Engineering, LaTiCE 2015, Taipei, Taiwan, April 9-12, 2015*, pp. 124–127.

[21] R. Bidarra, J. Boers, J. Dobbe and R. Huijser, "Bringing a pioneer games project to the next level", in *Proceedings of the 3rd İnternational Conference on Game Development in Computer Science Education. GDCSE 2008, Miami, Florida, USA, February 27-March 3, 2008*, pp. 11–15.

[22] R. Dörner and U. Spierling, "Serious Games Development as a Vehicle for Teaching Entertainment Technology and Interdisciplinary Teamwork", in *2014 ACM International Workshop on Serious Games, Orlando, Florida, USA, November 7, 2014*, pp. 3–8.

[23] B. R. Maxim and B. Ridgway, "Use of interdisciplinary teams in game development", in *Proceedings - Frontiers in Education Conference, FIE 2007, Milwaukee, WI, USA, October 10-13, 2007.*

[24] A. Schaetter, H.-G. Koeglmayr, K. Blankenbach, and N. Markus, "Interdisciplinary Approach to Software Engineering Education", *J. Syst. Cybern. Informatics*, vol. 7, pp. 29–36, 2009.

[25] A. Vicente, T. A. G. Tan and A. R. O. Yu, "Collaborative Approach in Software Engineering Education: An Interdisciplinary Case", *J. Inf. Technol. Educ. Innov. Pract.*, vol. 17, pp. 127–152, January 2018.

[26] L. Jaccheri and G. Sindre, "Software engineering students meet interdisciplinary project work and art", in *Proceedings of the International Conference on Information Visualisation, Zurich, Switzerland, July 4-6, 2007*.

[27] J. Favela and F. Peña-Mora, "An experience in collaborative software engineering education", *IEEE Softw.*, vol. 18, pp. 47–53, April 2001.

[28] C. Y. Chen and P. P. Chong, "Software engineering education: A study on conducting collaborative senior project development", *Journal of Systems and Software*, vol. 84, pp. 479-491, March 2011**.**

[29] W. M. Phillips, G. D. Peterson and K. B. Aberle, "Quality Assurance for Engineering Education in a Changing World", *Int. J. Eng. Educ.*, vol. 16, pp. 97–103, January 2000.

[30] A. Giuliano, "Accreditation of engineering programmes: European Perspectives and Challenges in a Global Context." *Eur. J. Eng. Educ.*, vol. 32, pp. 273–283, June 2007.

[31] "Washington Accord." *www.ieagreements.org* [Online]. Available: http://www.ieagreements.org/accords/washington/. [Accessed : 20-Sep-2001]**.**

[32] MUDEK, "Mudek Briefly," *www.mudek.org.tr* [Online]. Available: http://www.mudek.org.tr/en/hak/kisaca.shtm. [Accessed: 15-Jan-2019].

[33] J. R. Rowland, "A fine-grid model for evaluating multidisciplinary team experiences", in *Frontiers in Education Conference. 31st Annual. IEEE 2001, Reno, NV, USA, October 10-13, 2001,* pp. T4A-5-8.

[34] A. Kaur, "App Review: Trello", *J. Hosp. Librarianship*, vol. 18, no. 1, pp. 95–101, February 2018.

[35] J. Rumbaugh, I. Jacobson, and G. Booch, *Unified Modeling Language Reference Manual, the (2Nd Edition)*, Pearson Higher Education, 2004.

[36] R. Likert, "A Technique for the Measurement of Attitudes", *Arch. Psychol.*, vol. 22, pp. 5–55, 1932.

[37] S. Robertson, J. Robertson, *Mastering the Requirements Process: Getting Requirements Right*, 3rd ed. Addison-Wesley Professional.

VITAE

Dionysis GOULARAS received his PhD degree in computer science from the University of Paris-SUD, Paris, France, in 2005. He holds bachelor and master degree in computer science. He is currently working as an Assistant Professor in the Dept. of Computer Engineering, Yeditepe University, Istanbul, Turkey. His research interests include artificial intelligence, image processing, computer graphics, computer vision, virtual reality, medical imaging and 3D reconstruction.

Mert ÖZKAYA is an Associate Professor of Computer Engineering at Yeditepe University and leads the software engineering research group at Yeditepe. Dr. Özkaya conducts active researches on model-driven software development, software architectures, formal verification, and empirical software engineering. Dr. Özkaya also conducts researches on improving the academia-industry collaborations on software engineering, software engineering education, and improving productivity in software engineering. Dr. Özkaya contributes and contributed to several national and international research projects on software engineering and published many article and conference papers on model-driven software engineering and empirical software engineering. Dr. Özkaya acts as a technical/academic consultant of many software development companies in Turkey.

Tacha SERIF is an Assistant Professor in the Dept. of Computer Engineering, Yeditepe University, Istanbul, Turkey. Prior to this post, he was a consultant at HP UK&I and prior to that a fellow at Brunel University London as part of an EU FP7 project. Dr. Serif's research interests include mobile computing, context and location-aware systems, networking, distributed systems, Internet of Things (IoT), healthcare monitoring systems and e-learning technologies. At his current post, Dr. Serif is responsible for undergraduate and postgraduate Data Communications and Computer Networking, Mobile Application Development and Object-Oriented Programming courses.

Sezer GÖREN received the BS and MS degrees in EE from Bogazici University and the PhD degree in Computer Engineering from the University of California, Santa Cruz. She was a senior engineer in Silicon Valley from 1998 to 2004 at Syntest, Cadence, Apple, PMC-Sierra, and Aarohi Communications. She is currently a full Professor of the Dept. of Computer Engineering, Yeditepe University. Her research interests include reconfigurable computing, design automation, design verification, test, computer arithmetic, vehicular technologies, and embedded systems design.