

Article

# The MOSAICS Model of Educational Approaches for Teaching the Practice of Software Project Management

Costin-Anton Boiangiu <sup>1,\*</sup> and Iulia-Cristina Stănică <sup>1,2</sup>

<sup>1</sup> Department of Computer Science and Engineering, University POLITEHNICA of Bucharest, Splaiul Independenței 313, RO-060042 Bucharest, Romania; iulia.stanica@gmail.com

<sup>2</sup> Department of Engineering in Foreign Languages, University POLITEHNICA of Bucharest, Splaiul Independenței 313, RO-060042 Bucharest, Romania

\* Correspondence: costin.boiangiu@cs.pub.ro

Received: 20 December 2018; Accepted: 23 January 2019; Published: 27 January 2019



**Abstract:** Maybe you heard the line “managing programmers is like herding cats”, and if you consider there is some truth behind this, then you should, perhaps, think how it is to teach people to perform this job. As we know from the research literature, there is no such thing as a consensus about the most suitable teaching method of a software project management course targeted to information technology students. Moreover, the majority of publications focus on the theoretical aspects of the course, thus leaving little details about the application of the theory, or how to experience the practical side. The paper at hand proposes an abstract model of educational approaches, suggestively named MOSAICS, which may be used in teaching the practical side of a software project management course.

**Keywords:** software project management; educational model; teaching strategies; teaching practical side; MOSAICS; cooperation; computer science teaching approaches; grade data alignment; skewness; kurtosis

---

## 1. Introduction

### 1.1. What We Try to Present and Accomplish

Software projects are based on complex and well-defined procedures for creating software applications, which include numerous phases—from establishing the requirements to testing and launching the final product. The development of software products is a relatively recent domain and its process is limited by three constraints: Time, quality and cost. Each one of these three factors can have a huge impact on the software development process, therefore, the existence of an appropriate software project management approach is essential. Teaching software project management (SPM) to information technology (IT) students is an important step in their formation, as they should be trained for what is expected of them in the IT industry. There is little research on the matter and most publications focus on the theoretical aspects, which must be taught. Our proposed model of educational approaches focuses on the practical side of teaching software project management. Each approach is centered on a certain idea, a starting point, which is developed and applied in various aspects, such as the project theme, evaluation criteria, milestones, or team structure. Our seven proposed approaches are: Independent (based on cooperation), synergistic (based on collaboration), synchronous (rigid), anarchic (focusing on rivalry), competitive, original (flexible), and mixed (a combination of the previous ones).

## 1.2. Previous Work

Cooperation has been thoroughly analyzed since 1980, in order to observe its effects when being applied to small-group teaching [1]. Peer-tutoring and group-investigation are two methods which involve cooperation: The first one requires a student to become the tutor of the group (or, in a software project, the manager), while the second method is more student-centered, with four essential elements—the four “I”s (interaction, investigation, interpretation, and intrinsic motivation) [2]. In software project development, some approaches, which involve teamwork and cooperation, include the participation of specialists from the software industry to give students an accurate insight regarding the process of developing a software project. After several experiments, teachers from the software project management course of the University of Maryland, USA concluded that communication, good management, and teamwork are the essential aspects of this approach. The specialists from the software industry were the supervisors, guiding the project managers (chosen from the students) in their process. Even though the most experienced students were chosen as team managers, team members have mentioned that having a responsible and competent leader is mandatory for assuring the success of a project. The feedback and results were positive, preparing students for real-life situations, and making them learn a lot from their realizations or failures: “(The class) provided real world experience without the cost of getting fired from the many mistakes I made” [3].

The “Agile” method has become more and more popular, focusing on cooperation between all actors involved in the project (team members, product owner, and other stakeholders). The individual factor is also essential, as each person should remain motivated and self-organized, while maintaining harmonic interaction with the teammates, during all stages of the development cycle. The development stages or iterations of an Agile method have a relatively short duration, and they are usually called “sprints”. The Agile method has started to be introduced in education and it can revolutionize teaching methods: “Agile is fundamentally about learning, people, and change—three things we struggle with in education and handle poorly at the present time” (Peha, founder of “Teaching That Makes Sense”) [4]. Thus, it comes naturally to try to integrate the Agile educational approach into teaching software project management courses. Anarchy elements are present in an Agile method: Each member should be capable of being their own organizer, as there is no real manager—the Scrum master is just responsible for facilitating meetings and communication, not for giving orders and distributing tasks. Various researches [5,6] showed good results when using the Agile method for teaching software project management. Devedzic and Milenkovic [6] have practiced the agile software development method in various software-related courses, for both undergraduate and graduate students. The results were the most effective when the interaction between team members was maximized, thus focusing on the cooperative aspect. In addition, Agile characteristics, such as daily meetings and short iterations, were essential for assuring the success of obtaining high-quality products at the end of the course.

Collaborative learning represents a shift from the teacher-centered learning to the one focusing on students and their interaction. It can be achieved through wikis, writing groups, workshops, discussion groups, or learning communities [7]. Chu et al. [8] studied the effectiveness of using wikis for collaborative learning in various disciplines, including English language, mechanical engineering, and information management. A wiki is a website (and also a database), where users can share their knowledge and experience by creating or editing its content. The students were asked to create a case analysis regarding the information management of a real organization. Among the above-mentioned disciplines, the teams working on the information management projects obtained great results, in terms of overall learning, motivation, and group interaction. This can mean that wikis are efficient for collaborative work related to IT management fields. However, the projects were based on theory rather than software development. Therefore, if we consider the fact that students did not perceive wikis as being efficient in terms of knowledge management and that technical aspects were not primordial, we can conclude that wikis must be combined with other practical collaborative methods to assure pure software project management. Another example of the collaborative approach used for teaching software-related courses was used at the National Central University of Taipei, Taiwan. Chen et al. [9]

mention that students often lack communication and collaborative skills, mandatory in software project management. Their proposed project, called “Project Issues Monitoring Information System (PIMIS)”, aimed to help students develop software project management competencies and prepare them for facing real-life problems. Industry experts were also involved and students had to work in teams to create prototypes of revolutionary IT products for a company. The PIMIS project uses a cyclic software development strategy, focusing on a meetings-flow approach: Teamwork and collaborative development are essential for establishing functional meetings and assuring interconnectivity between them. The results were positive from both qualitative and quantitative perspectives [9].

Competitive approaches can keep students motivated and assure an objective evaluation system. Recent research shows the tendency of combining e-learning platforms, competitive approaches, and social networks [10]. Even though software engineering refers just to the technical aspects of a software project, thus involving a less complex process than software project management, the two courses are strongly related to one another. Competition in software-related courses usually manifests through games, where students must compete against one another to become victorious. Sheth et al. [11] tested this approach at the Columbia University of New York. Two different competitions were launched. The first one was a Battleship tournament, where students had to implement a player capable of defeating the teaching staff’s artificial intelligence. The necessity of building an efficient strategy, the competitive aspect, and the bonus credit points, which could be won, made the students very dedicated and active. The second competition was based on gamification principles using a platform named HALO (“Highly Addictive socially Optimized Software Engineering”). HALO used game mechanics, such as quests, motifs from famous games and movies, or rewards, to keep students motivated and engaged in the software engineering course. The quests had the role of presenting important software techniques (such as white box or black box testing), thus hiding not-so-pleasant software activities under the appearance of game stages. Similar to real games, students receive various rewards, such as titles, experience points, or achievements—aspect which can increase motivation, as each student wants to earn more rewards than their competitors. If at the beginning, the assignments were individual, in the end, students were allowed to work in teams and use their imagination in order to create their own HALO-based projects. The students’ feedback was in general positive, finding the approach fun and beneficial for learning good software practices.

Several educational approaches have attempted to deviate from the rigidity of traditional teaching methods, using creativity, flexibility, and originality as their main principles. Gamification aspects are becoming more and more frequent, as serious games represent a viable alternative for teaching and educating students. Such approaches have become popular also for teaching software project management [12]. For instance, SimSE is a serious game used for teaching software engineering project management, developed using several game design patterns. The web-based game puts the player in the role of a project manager—they must make decisions, communicate with their “employees”, and stay on budget while assuring the success of the virtual project [13]. To analyze the effectiveness of using this game as a teaching approach, a study was conducted to see how students behave in an SPM-related serious game [12]. Pre- and post-game questionnaires show that the SPM related knowledge of the students improved after playing the game.

Furthermore, games can be used as a context for students to work in teams in order to develop their software project management skills [14,15]. Simsoft is a game using both virtual and real game elements, such as a Java-based dashboard, a printed game board, and plastic coins. Students have to work in teams and all of them must assume the role of a project manager. After receiving a base scenario in the Simsoft game, they must use the information from the game and their previous knowledge in order to complete the project successfully, without exceeding their budget. The results of the approach were encouraging, successfully addressing various aspects of a research project, such as analysis, synthesis, and evaluation. The human-centered criteria were also met, focusing on empowering users, making them understand essential SPM notions and putting their problem-solving capabilities to test [14]. Another example is ProDec, a serious game, which simulates all the stages of a software

project's lifecycle, putting the students' decision-making skills at test. The game must be played in teams and students must have previously acquired theoretical knowledge about software project management [15].

There are some approaches that use unique rules and project themes, tending to be on the rigid side of educational models. Such examples can include learning by simulations and experiential learning, both of them being used in software project management courses. The University of North Dakota [16] used experiential education principles for their SPM course, giving up on traditional lectures. Students took part in class discussions, testing their knowledge and offering feedback to their colleagues. On the practical side, they had to work in teams to develop useful software for a spacecraft program called OpenOrbiter. The goal of the project was to create a CubeSat, a miniaturized, low-cost satellite. Students became part of already-existing teams from the program and they had to accomplish multiple project management tasks, such as documentation, diagrams, and a risk management plan, focusing on intra-team communication and providing feedback. They received some evaluation tests before and after the course to evaluate their progress. Even though the students' learning was demonstrated, the comparison of the proposed approach with other ones from the literature is difficult, in the absence of a unique measurement. Another example of an approach, which does not have traditional lectures, is the one based on simulations for teaching SPM [17]. By using a system called AMEISE, based on a simulation engine named SESAM (Software Engineering Simulated by Animated Models), students were trained through experimental learning principles. The system's simulations aim to train different project management competencies, as trainees can learn from their failures. Simulations bring the advantage of creating worlds, which are otherwise unavailable for students, but they also have some drawbacks, as they present a particular model, thus ignoring other possible aspects that can appear in real-life situations.

Mixed approaches are often preferred in education, as they are more flexible and succeed in taking the advantages of different models. The rigidity and fixed characteristics of using a singular approach can thus be avoided. In teaching and research, there are various paradigms that can guide the steps of an educational approach. Mixed-methods have been proposed in both language teaching [18] and software project management [19], based on a mix of different approaches or paradigms, such as positivism and constructivism. Both previously mentioned teaching fields are interdisciplinary, so it comes naturally to adopt mixed approaches for teaching them. Positivism refers to the philosophical approach, which considers that knowledge is gained through experiences [20]—this can be applied in both evaluating and teaching students, thus focusing on the practical side of a course, not on theory. Constructivism, also known as interpretivism, emphasizes the idea that each person can construct their own reality based on their own interpretation. Constructivists think there is no universally acceptable reality, there are only consensuses or conventions [20,21]. This can be represented through original and independent approaches, where each student is given a certain degree of liberty when tackling their projects. The need for a certain degree of independence in a teaching method is reflected in the individualized learning approach [22]. This technique refers to adapting the content, the learning pace, goals, and curriculum to the personality and profile of each student. Research and reports [23] show that individualized teaching and learning can bring many benefits for students of all ages, as it emphasizes students' individual strengths, motivation, and expectations. This technique has been shown to also improve school attendance, self-efficacy, and career planning activities [24].

A. De Lucia [19] proposed a mixed approach for teaching two related disciplines, software engineering and software project management, which took place in the same semester, but were taught to bachelor's students and master's students, respectively. The mixed scenario is reflected through the team organization, where bachelor's students received development roles, while the master's students were the managers. This approach of mixing the two different generations, with different levels of knowledge and experience, in the same team, can simulate accurately a real-life scenario. Their research showed that imposing leaders with more experience can be in concordance with the natural-leaders' well-known aspects. Several constraints were imposed: Milestones that must

be respected, team organization and preferences, and evaluation responsibilities for the managers. The research was conducted during seven years and the results are encouraging—there were no project failures, the quality of the projects was elevated, there were very few abandonments, and the overall effort was reduced. The students' feedback was also positive: They felt more responsible, organized, and felt that the course was contributing to their professional training. The educational approaches presented in De Lucia's research [19] are used also across multiple disciplines, for example, in the "Dreams and Teams" international program by the British Council, which focuses on developing the leadership skills of young people from all over the world [25].

## 2. Objectives

In order to describe each proposed approach, our paper presents a model called MOSAICS, where a representative term was chosen for each term. These notions already exist for general-purpose education, therefore, we will describe their meanings, in concordance with their use in our software project management approaches.

Independent learning is probably one of the most widely spread processes in the actual educational system. Known also as self-regulated learning [26], independent learning focuses on concepts, such as personalization and ownership, making the students assimilate the learning process as being a result of their own work [27]. "Spoon feeding, in the long run, teaches us nothing but the shape of the spoon." (E.M. Forster) - this quote is completely relatable in the educational field, as entire generations of students with no true identity are created. The new teaching methods should encourage independent learning, where the teacher provides an idea, a certain topic, and the necessary resources, which must be later developed and researched based on one's abilities, skills, and originality. Independent should not be mistaken for individual, as independent learning can still be applied in teams, where each person has a well-defined role based on their skills and previous experience [28,29]. The independent approach, at the team level, takes into account the idea of cooperation, as "the learner, in conjunction with relevant others, can make the decisions necessary to meet the learner's own learning needs." (Cyril Kesten).

The term "anarchy" is usually used with a pejorative meaning, being defined in the Oxford Dictionary as "the absence of government or control; disorder, confusion" [30]. As a matter of fact, anarchy has a positive side and does not refer necessarily to an "anti-authority" attitude; it concentrates on the idea of showing suspicion towards any type of hierarchy, also being described as a "theory of decentralization" (J.P. Clark) [31]. This aspect is also relevant for the anarchic approach in education, as its main goal is to get away from formal learning, hierarchies, and rigid patterns. A modern Spartan principle refers to the fact that the person who is the strongest, the most skillful, and the wisest is also the one who becomes successful. However, today's anarchic approach focuses on diversity, as each student must gain their own knowledge, do research, and seek opportunities on their own without waiting for advice or indications from their peers. The anarchic approach has been promoted by big companies, such as Valve Corporation [32] and Google, who are famous for using a decentralized leadership principle, with their employees being free to establish their goals, tasks, and make their own decisions. It comes naturally then to try to combine the anarchy principles from company management with those useful in education, thus creating a teaching method for the software project management course.

We hear the term "Collaboration" everywhere, but what is the true meaning of it? Collaboration refers to the idea of creating a unitary community, with members who have a common purpose and are willing to share their precious resources, knowledge, experience, and skills, to the benefit of the team. Collaborative learning is a modern educational approach, where the classical "teacher-centered" method is replaced by one based on team-work and students' collaboration. The teachers become mediators and mentors, transmitting their own knowledge and past experiences to their "disciples". This educational approach can be expressed through various ways, such as class discussions, research teams, and projects, which require analysis and response to other people's work and ideas [7].

Collaboration and cooperation are not synonyms, as collaboration implies “To work with another or others on a joint project”, while cooperation means “To be of assistance, or willing to help” [33]. Perhaps the best way to tackle the subtle differences in collaboration and cooperation is to follow the descriptions provided by Jesse Lyn Stoner [34]. Collaboration is based on the shared vision of the product that is created with a combined effort. Cooperation, on the other hand, is about exchanging resources and information to mutual benefit, but not necessarily for a shared goal; in this case, any product that arises will be mainly the result of the individual, and not of the team, effort. Therefore, when collaborating, people share a common goal, they must make decisions together and find solutions to problems. On the other hand, cooperation does not imply the same level of trust or teamwork. Each individual wants to accomplish their own goals while being willing to provide assistance or offer help for their mutual benefit [33].

Competition in education has existed since ancient times, as famous Roman teachers are known to have used competitive approaches in the teaching process [35]. “The power of [competition’s] effect makes its use very tempting. Little else gets a group of young people more energized than competition.” (John Shindler). However, the competitive approach can have its upsides and downsides. Therefore, a debate has risen between two types of theoreticians: Firstly, there are those who view this approach as being beneficial, as it motivates students and prepares them for real-life competition; on the other hand, competition may be associated with stress and lead to frustration and conflict—thus, it can be considered counter-productive, as it is opposed to collaboration.

The original approach in education aims to develop the creative part of each student, as creativity can later lead to discoveries and spectacular achievements. In recent years, traditional educational approaches have started to be replaced with modern ones. There is less of a focus on memorizing and reproducing theoretical aspects of the lessons, as the main goal of education is understanding and acquiring useful knowledge while progressing as an individual. The originality of this approach manifests through various aspects, encouraging debates, questioning, experimentation, and personal development [36]. Creativity manifests differently based on the teaching subject—it comes more naturally for arts and foreign languages, but it can be integrated successfully in various areas, such as science, information technology, or business studies. Some important advantages come with this approach, independent of the teaching subject: Students have a sense of personal achievement, they gain confidence by putting into practice and presenting their own ideas, and they reflect critically on their own ideas or those of their peers. The original approach focuses on flexibility, distancing itself from the rigidity of traditional approaches.

The term “synchronous” refers to an event or phenomenon, which “exists or occurs at the same time”. It is from the same word family as “synchronize”, which means, in the computing field, the action of “causing a set of data or files to remain identical in more than one location” [30]. In the educational field, the term, synchronous learning, exists and it defines forms of education where students and teachers are involved in the learning process simultaneously, but not necessarily in the same place [37]. It refers to the way communication takes place in a lecture and it exists in both face-to-face (“in-person learning”) and online teaching (web conferences, chat sessions, etc.) [38].

### 3. Method

#### 3.1. MOSAICS Defined and Briefly Commented

Software project management is a tough domain to teach. It is not clear what path to follow so that students obtain the desired interlacing of hard and soft skills necessary to perform in the software industry.

There are a lot of possibilities, in terms of how to:

- Enable students to manage a project team formed by their colleagues;
- interact and respond when managed by a colleague;
- calibrate grading for correct balancing of the individual and the team effort; and

- motivate the students to integrate into the team and not get lazy and leave the job to other colleagues.

There is no such thing as a perfect solution, but our aim is to offer a formalization for a model of educational approaches that may be successfully used in this research area. In order to do so, let us build an analogy between software project management and the more mathematically-rigorous domain of data clustering and cluster analysis.

The “cluster analysis” research area has found across time numerous and various utilizations. Like Klaus Krippendorff [39] stated, since 1980: “Clustering originated in anthropology and in psychology in response to the need for empirically based typologies of cultures and of individuals. Computational problems hindered the initial development of these ideas. But by the early 1960s clustering techniques emerged in a variety of other disciplines, including biology. Applications are now so numerous that references to them would fill a book.” Additionally, as time passes, a lot of other domains have benefited from clustering analysis and validation techniques: Signal and image processing, computer networks, machine learning, and so on.

The current paper generalizes and adapts some basic concepts of the cluster analysis to the specific task of teaching, in a university, a course of software project management. In fact, what we need to deal with is the concept of a team of students (analogy for a cluster of objects) that are functioning along with other teams (other clusters on the same level) on a project. We may analyze their intra-team and inter-team relations as intra-cluster and inter-cluster measures between objects and clusters [39]. To further formalize the relations between students inside their team or the relation between teams, we may continue the analogy with the interpretation and validation of data clusters using a technique called silhouette factor introduced by Peter J. Rousseeuw [40]. This is accomplished by computing two elements of the candidate clustering: The cohesion and the separation. The cohesion is an intra-cluster measure, which tries to identify how close the relation between the cluster’s objects is. The separation, on the other hand, is an inter-cluster measure, whose purpose is to examine how distant the relations between clusters are. However, things are not always clear in data clustering. The fuzzy clustering technique [41] allows us to make our next analogy by generalizing the fuzziness measure to project management, regarding this measure in respect to the balance between flexibility or rigidity towards the: Student’s original contribution importance, final product specifications, development framework strictness, completeness of interfaces between modules, strictness of user’s requests and acceptance procedures, milestone synchronization, intermediate deliverable definitions, and so on.

Taking all the above analogies into account, we are now ready to propose a model of educational approaches targeted at teaching the practical side of software project management. We will name the proposed model MOSAICS for two reasons:

- The first reason is that it represents the whole idea of project management, thus building a mosaic at an intra-team level from people, which must complement perfectly to enhance the team performance and combine the resulted mosaics at the inter-team level to complete the project; and
- the second reason is that the name is constructed from the initials of the seven proposed educational approaches, which are spread evenly across the proposed model, to fully specify both its origin and boundaries: Mixed, Original, Synergistic, Anarchic, Independent, Competitive, and Synchronous.

### 3.2. The Axis—Defined and Commented

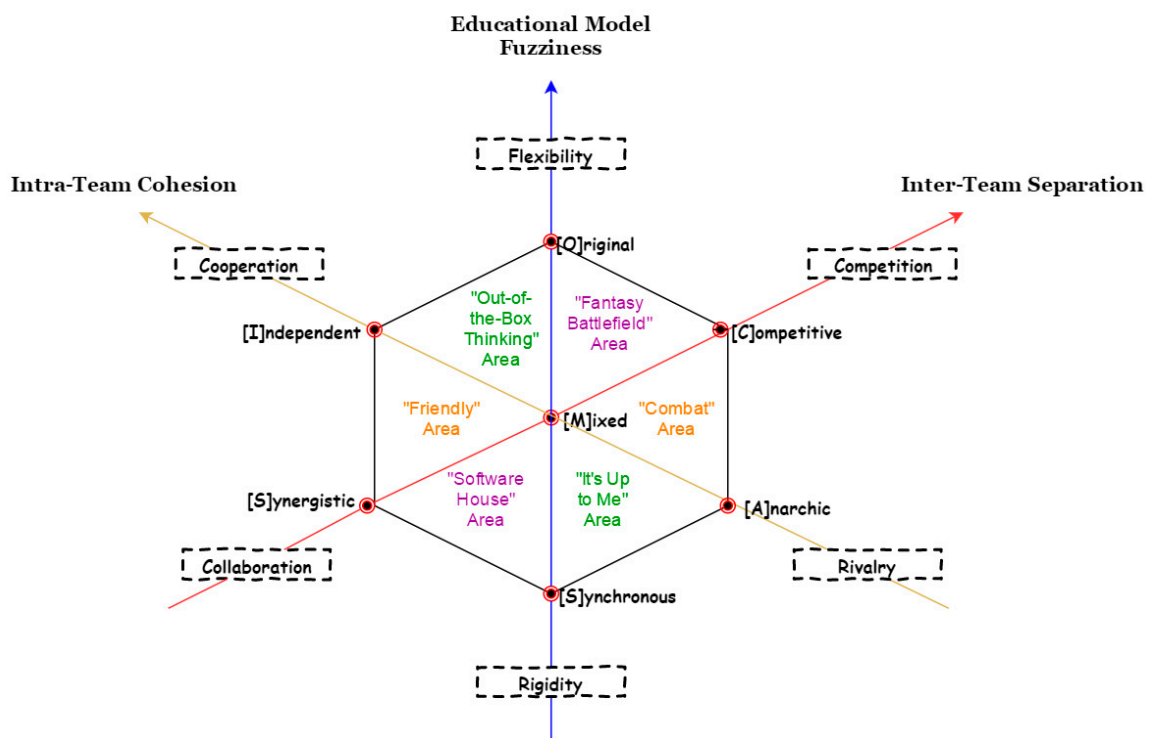
In an analogy with data clustering techniques, we will further specify our model by adding three axes: cohesion, separation, and fuzziness:

- The cohesion axis: Measures the intra-team relations, in other words, the balance between team members’ rivalry and cooperation. We will denote this as the Intra-Team Cohesion axis (ITC);

- the separation axis: Measures the inter-team relations, more precisely the balance between a team’s competition and collaboration. We will denote this as the Intra-Team Separation axis (ITS); and
- the fuzziness axis: Measures the characteristics of an educational approach to balance between the flexibility and rigidity of the overall system, or more precisely in terms of product characteristics, development framework, interfaces, requests, acceptance, milestones, deliverables, etc. We will denote this as the Educational Model Fuzziness axis (EMF).

Having now defined our educational approaches model, it is time to specify its boundaries on every axis and the relation between them. The most suitable graphical representation is presented in Figure 1.

The MOSAICS model axes are not fully independent of each other, as in the case of a Cartesian coordinate system with three axes, so we may regard the ITC, ITS, and EMS as slightly dependent variables, exhibiting clear inter-dependency monotonicities. Imagining the representation as a planar one, Figure 1 also reveals the impact of any variation across any axis.



**Figure 1.** The proposed model of the educational approaches for teaching the practice of software project management.

### 3.3. The Areas Commented

As a result of the axes definitions and their boundaries, we may observe that the proposed educational approaches model is composed of six adjacent areas, each having its own specific features and unique capabilities, both derived from blending the educational approaches that define their extremities:

- Anarchic-Competitive-Mixed, the “Combat” Area: Intense competition for both inter and intra-team members;
- Competitive-Original-Mixed, the “Fantasy Battlefield” Area: An inspirational/imaginational battleground, where fantasy works together with both software development and project management;



- Original-Independent-Mixed, the “Out-of-the-Box Thinking” Area: This is the realm where the requirements are just the start of the story;
- Independent-Synergistic-Mixed, the “Friendly” Area: Here, both the inter and intra-team relations are meant to be of collaboration and cooperation;
- Synergistic-Synchronous-Mixed, the “Software-House” Area: This is, practically, the real-world approach towards managing software development; and
- Synchronous-Anarchic-Mixed—the “It’s Up to Me” Area: A place where personal interest is better put in use, due to the increased rigidity of the product characteristics, development framework, interfaces, requests, acceptance, milestones, and deliverables in conjunction with the lower possibility of real differentiation between teams.

### 3.4. *The Transitions Commented*

Now, it is time to finish the MOSAICS model description by specifying the last remaining component, the meaning of the transitions between the educational approaches situated in proximity to one of another:

- Anarchic-Competitive, the “Intra-Inter Team Competition” balance: This transition correlates the idea of competition between teams and the rivalries between individuals;
- Competitive-Original, the “Fixed-Flexible Competition Rules” balance: It finds the perfect match between strict rules, specific to a classical competition, and the suppleness of the original approach;
- Original-Independent, the “Fixed-Flexible Product Requirements” balance: This transition can be assimilated with a software product request, which is given to the team, with specific requirements, but also having a certain amount of flexibility;
- Independent-Synergistic, the “Link vs. Separation Between Modules” balance: It comes at different intensities: Zero, weak, strong link strength;
- Synergistic-Synchronous, the “Usefulness/Risk vs. Redundancy/Robustness” balance: A transition from collaboration to rigidity, simulating real-life situation; and
- Synchronous-Anarchic, the “Hierarchy Power” balance: It manifests through strong, weak, zero continuity/predictability in team management.

## 4. **The Proposed Approaches: One-By-One, Pieces of a Giant Educational Puzzle**

Our proposed MOSAICS model includes seven distinct approaches, which will be described one-by-one, by taking into consideration several aspects: The main idea, the end product, their detailed description, the evaluation process, as well as observed advantages and disadvantages, from the authors’ and the students’ comments.

### 4.1. *Independent Approach*

Our proposed model includes the use of the independent approach for teaching the software project management course. The practical side of project management cannot be taught individually, therefore, the students will have to work in teams, where each person will acquire their own knowledge (through independent learning) and apply their own skills for a mutual benefit. Different project themes are proposed (with different goals, varying from games to serious applications) and the students must use their curriculum vitae to apply to one or more projects, based on their preferences and skills. After the establishment of the teams, each student must determine on their own which role they want to accomplish to establish a valid project management hierarchy. The idea of customizing the project theme for each team of students can constitute a huge advantage, as it can be personalized based on the students’ preferences or interests. Homogenous teams can be constituted, which can improve the intra-team communication. Therefore, our proposed project themes included image analysis software, traffic and circuit simulators, e-commerce or educational platforms, as well as various games.

Since each team of students receives a different project theme, the need to communicate with the members of different teams is minimized: “They have something else to do, they can’t help us”. In this case, the process of demanding or receiving helpful information would have to occur unconditionally between teams, without expecting anything in return. This phenomenon is very unlikely to occur (helping others from pure altruism), thus making the team effort extremely autonomous and keeping the interaction between teams to a minimum. This can constitute an advantage for the idea of independent learning, as students will have to put in their own effort to acquire the necessary knowledge, as they cannot rely on the help and opinions of their peers.

In addition, the competition between teams is diminished, as different project themes simulating different scenarios, requiring different competencies or approaches from the team members, can make the subjects focus only on their work in the early stages of the project. This can constitute an advantage, as it minimizes the exterior distractions and the team members can work better on their own ideas and solutions. The competition factor can be introduced and observed later, when the teams must present and “sell” their applications. They should now focus on presenting their own idea as being the more attractive, original, well implemented, and efficient one.

The evaluation part can be simplified at the intra-team level. The students are evaluated individually (during the semester) and at the team level (at the end of the semester), the final grade being calculated as a weighted average. Therefore, each student should focus on both self-learning and teamwork, as both aspects will have an impact on their final grade. Usual inconsistencies, which appear between the efforts of people from the same team, can, therefore, be diminished, as no one can expect that their work will be handled by their teammates.

On the other hand, it is very difficult to find a relatively high number of different project themes that have the same level of difficulty and require the same amount of time to complete. Therefore, a serious disadvantage can occur: The grading system can become unbalanced, as subjectivity tendencies can be observed for a project with an increased level of difficulty. In addition, different themes can desynchronize the milestones, as specific steps can have different completion times, based on the type and difficulty of the project.

#### *4.2. Anarchic Approach*

In our anarchic approach, the students were divided into teams and they all received the same project: A board game, where the teams must compete against each other to obtain victory after several rounds. Each student has a pawn, which must be programmed to assure their team’s success, but the individual benefit should also be considered—at the end of each round, a pawn can end up in another team, so the students must be capable of adjusting their strategy on the spot. However, the algorithm of the team structure generation guarantees that each individual will spend  $50\% + 1$  rounds in the initial team, so the team-level strategy remains very important. Two grades are considered: One based on the individual success (the number of wins of the temporary team) and one based on the team success (the number of wins of the original team).

To apply pure anarchic principles, the teams have no actual leader: Neither the teacher nor any of the students have the right to impose ideas or restrictions. The teacher can provide guidance or advice, while a team manager exists, but only as a mediator between team members. Each individual must think of their own strategy, establishing goals and milestones, which must be respected. This can represent an advantage, as students may train their self-knowledge and self-management abilities, thus becoming more competent and self-aware. Each member of a team must perform the same amount of work in order to achieve the desired result, which can balance the amount of effort given at the team level. Furthermore, it can lead to an efficient observation of the individual performance, balancing eventually the grading system, based on the evolution of the team-individual duo.

As opposed to the independent approach, where the individual interest and the team interest are strictly interdependent, in the anarchic approach, they can be completely opposite. Initially, one may think that their strategy should be positive, constructive, and towards the benefit of the

team, as their final contribution in the competition will be evaluated. Instead, they should never forget that they can switch teams at any time, passing from a collaborative model to a competitive one or to pure, uncontrollable anarchy. The anarchic approach can be regarded as an accurate simulation of inter-human relations, as each individual must try to assure an efficient intra-team and inter-team communication.

Unfortunately, there are some disadvantages of the anarchic approach, which are almost impossible to avoid. Some students might become frustrated, as it can become extremely difficult to find the optimal balance between a secret individual strategy and the benefit of the team. Furthermore, efficient communication is difficult even in hierarchic models, so a decentralized approach would aggravate even more the inter-human interactions. Some individuals can even become hostile towards the rival teams or even towards their teammates in an attempt to emphasize their self-centered benefit.

#### *4.3. Synergistic (Collaborative) Approach*

In software project management, collaboration is essential in order to assure the flawless development of the project. Therefore, when using the collaborative approach to teach the practical aspect of SPM, several aspects must be taken into consideration: Team-work must be encouraged, students' work must be carefully guided, and collaborative effort should be encouraged continuously during the project's phases [9]. Our proposed collaborative approach was tested during the software project management course, when the students were asked to contribute to the development of an "Automated Data Analysis and Extraction System". The goal of the project is to create a solution for transforming written documents (such as books, newspapers, or maps) into electronic documents, in a standard format, accessible on a great variety of devices. To put into practice the collaborative approach, the project is divided in several sub-projects and modules, which are assigned to different teams. The sub-projects are digital image enhancing, image segmentation, entity hierarchy generation, and optical character recognition (OCR), with the last one being the most complex and having to be divided into five separate modules. All these modules, which must be implemented by different teams, must communicate through eXtensible Markup Language (XML) files, thus encouraging team-work and inter-team communication. For the evaluation part, two grades are considered: One for the individual work (individual grade) and one for the teamwork (project grade). An interesting aspect is introduced to emphasize the importance of the individuals' participation in the team project: The project's grade is multiplied by a factor calculated based on the individual grade, so that if the contribution of the member to the team is non-existent, their final grade will be zero.

Probably the most significant advantage of the synergistic approach is the capacity of simulating as accurately as possible the process of developing a product, by using several independent teams. Each team works on a different part of the project, having a well-established role, and each student is successfully integrated into the common effort of creating a complex software product. The roles inside a team are not predefined, as the members are encouraged to choose a manager who would coordinate their entire activity. The evaluation process mentioned above has the goal of encouraging both individual and teamwork. Furthermore, the results obtained by each team will be cumulated in a final product, therefore, the need to communicate with the other teams is vital. This aspect emphasizes the collaboration process, not the competition between teams, as inter-team separation must be kept to a minimum.

At the same time, the synergistic approach has a series of disadvantages. Firstly, the fact that each team has a different purpose can lead to unbalanced difficulty levels of their tasks. This can result in unsynchronized milestones or discrepancy and discontent regarding the final grades. Furthermore, as each team works on a different module, the testing process of these modules becomes difficult, which can eventually lead to a faulty final product. If the module created by a specific team has a failing component, it can cause frustration among all students, as the final product will not work properly or not work at all. In conclusion, the synergistic approach focuses mostly on collaboration, not on quality, which can be extremely harmful for a software project.

#### 4.4. Competitive Approach

Based on the famous Olympic Games, the term “Olympic education” has appeared in sports education, promoting the educational principles from ancient times and highlighting the quintessence of competitive spirit [35]. Our project proposal for testing the competitive approach consisted in the creation of a software system, which simulates a Formula 1 competition. The students worked on teams and each one of them developed the same project: Creating a circuit map and programming two Formula 1 cars, which must finish the race as fast as possible. The project has four important milestones, which are evaluated, as they represent essential parts of the final result: Establishing the project specifications, creating the viewer (the application for viewing the Formula 1 maps and cars), implementing the application server, and, finally, the artificial intelligence engine. After each milestone, the best team will be chosen and their code will be shared to all students, in order to continue their projects. Each team must establish a role for each one of its members and choose a project manager, who must present a CV and be accepted by the teaching assistant. In the end, a competition takes place, with multiple races where all teams compete against each other using both cars. The teams must remember during the whole project development process that they are in a competition and in the end, they have to win the car races. The evaluation process is based on the same principle: Two grades are considered - one individual grade and one for the teamwork, with the same interesting aspect, with the latter multiplied by a factor calculated based on the individual grade. In addition, more pressure is added on the shoulders of the project manager (PM), in order to assure their dedication and implication. The PM has to evaluate their team members periodically and give them grades, which will contribute to their final evaluation. Furthermore, this grade is multiplied by a factor based on the project manager’s grade, which is given by the teaching assistant. No team member can, therefore, receive a higher grade than the one of their project manager.

Probably the most important advantage of the competitive approach is the capacity of stimulating the team’s work in an efficient way, preparing them for a real-life competitive environment. The idea of a competition successfully combines both work and fun—multiple rounds (races) with final rankings based on points can encourage all teams to become combative and adopt offensive strategies. The final phase of the project, the presentation of the resulted products and the competition itself, can be extremely interactive and full of suspense. Thus, the desire of impressing others and performing better than them can be extremely stimulating and foster ambition. In the end, we can be assured that it is an approach which guarantees the development of a high-quality product.

Again, the roles inside a team are not predefined, as the members are encouraged to choose their own roles and designate a manager who coordinates their entire activity. A bitter competition can be observed also between the managers, as their responsibilities are increased: They must coordinate, organize, and evaluate their team’s members to keep them motivated and assure the team’s victory. This has another advantage: It simulates accurately the scenario of real-world competition between software project managers. In our proposed competitive approach, the evaluation process assures an objective and correct evaluation, as it is done by multiple people (team manager, teacher, teaching assistant) and it encourages both individual and teamwork.

Unfortunately, there are some disadvantages of the competitive approach. Some students might become frustrated if they lose the competition or even if they do not obtain the best results at a preliminary milestone. This can discourage them and diminish their motivation, leading to less implications and weaker performances. Another aspect which can lead to frustration is represented by the team’s structure—unbalanced teams in terms of knowledge and implication may have small chances of winning the competition. Therefore, it is essential to establish balanced teams, with an appropriate number of members and complementary competencies. In addition, communication between teams is not very significant—it occurs only at the end of a certain milestone, when all teams must use the solution of the winner team and continue their project based on that particular implementation.

#### 4.5. Original Approach

Originality defines the characteristic of an object, person, or creation as being unique, special, and unrepeatable. Software project management combines both information technology and business studies so one would believe there is no room left for originality. Our proposed project, however, has the goal of proving the contrary—“Van’Gogu” (a reference to van Gogh, since “Gogu” is used in Romanian as “dude”) is an original theme, which has the goal of letting students express freely their creativity. Each team of students creates a software application, which generates an “artistic” image, using procedural generation, fractal geometry, and basic algorithms. Inspired by the creations of van Gogh, all students must combine their technical knowledge and their artistic spirit in order to obtain beautiful creations. The students should create images that must be obtained algorithmically, as each geometrical structure or texture must be generated using mathematical formulas, procedural programming, and fractals. Inventiveness, artistic spirit, and math knowledge are required, and the students have 60 seconds to run the program to obtain an original creation. Creativity and originality are encouraged, as students are not allowed to use external resources or ideas which are not their own. In addition, at each execution of the application, the generation of slightly different images is recommended, so the teams must integrate a certain degree of randomness in their work.

The evaluation process is subjective, based on the artistic value of each resulting image. All Software Project Management teachers form a jury, which evaluates each team’s image based on their personal preferences and criteria, as “beauty lies in the eye of the beholder”. The winning team, apart from receiving the maximum grade for the SPM subject, gets a symbolic “Van’Gogu” trophy and the pride of having their creation displayed on the SPM official page.

The main advantage of the original approach is its flexibility. By distancing itself from the rigidity of traditional education, it encourages creativity, originality, and artistic spirit. It can show students that great, original ideas can lead to extraordinary results, even in a technological field, characterized by precision and rigorousness. All teams have a high degree of freedom: They receive a common theme, but they are free to use their imagination and any suitable technology to accomplish the project. The roles inside each team are not designated by teachers—each member is free to choose their own role, with those with the most original ideas having the opportunity of step up and take responsibility. In addition, our proposed original approach encourages competition between students: Each team wants to create the most beautiful image and defeat their competitors. In the end, they can also win the “Van’Gogu” trophy, which can make them extremely proud and boost their ego.

One of the disadvantages of the approach is represented by the fact that the result does not necessarily reflect the effort of the team: A technically complicated approach does not guarantee the attainment of a visually pleasing image. On the contrary, a good idea can lead to appropriate results, even if it is not technically challenging. This can lead to frustration for those who put in a lot of effort into their work and do not get the result they expected. Another disadvantage is represented by the subjective evaluation process; maybe the members of the team have other artistic preferences and an image which they consider beautiful can receive poor appreciation from the jury. This disadvantage is, however, minimized by including a larger number of people in the jury, with different tastes and artistic choices.

#### 4.6. Synchrononus Approach

Our unique proposed approach is related to the other definition of the “synchronize” term, the one used in the computing field. Therefore, we concentrate on the “identity of the data”, the uniqueness of a project theme, rules, and requirements. We thus defined a synchronous approach, which can also be applied to software project management. Our proposed project theme requested the creation of a game called “Letters carousel”—a game with rather strict, specific rules. Each team of students must develop a game where the players must use the letters rolled on some virtual dice to create many valid words which are at least four letters long. The game takes place in multiple timed rounds and takes into consideration the existing words of a certain language’s dictionary.

In comparison with other approaches, the teams do not have the same amount of time available for developing their projects: They must finish as possible, in a limited amount of time. Team members are encouraged to concentrate on the base functionality, as in “First make it run, then make it run right”. Therefore, the first steps of the project should involve the conception of a fun game, by designing the architecture, the user interface and, afterwards, the actual implementation. Additional functionalities are optional and they bring bonuses to the team’s grade. These functionalities might include: Conceiving a configurable scoring system with an emphasis on the additional points brought by long words, analyzing the frequency of letters’ occurrences in a certain language and configuring the dice based on this frequency, creating a multiplayer feature, creating unique player identifiers, score history, and top scores, and developing an achievements system. The idea that “we shouldn’t reinvent the wheel” is highly encouraged, as long as the teams do not share code between them or use already existing full games and present them as their own. Each team might choose if they want to leave the project open source or make it private. Git version control system should be used for code sharing and the SPM teachers will receive access to the repository to be able to follow the project’s evolution and facilitate the evaluation process.

All students receive the same project theme, with the same characteristics, detailed specifications, and strict milestones. The rigidity and strictness of the synchronous approach might be viewed as a double-edged sword: On one hand, it can be considered an advantage, as it simulates quite accurately the software industry when a rigid project proposal is given by a client. On the other hand, it can be viewed as a disadvantage, as it cancels completely the creativity and liberty of the students. However, even if the rules related to the project theme are unique and rigid for all students, some aspects are left to each team’s choice: They can use any programming language, any type of framework, library, or code fragments, and the game can run on any platform (mobile, web, desktop, etc.). This can be an advantage, as it allows the development of completely different products and lets each team put into practice their knowledge.

Even if it is difficult to evaluate many projects which have the same theme, the advantage of the evaluation process of the synchronous approach is objectivity. The rules are clearly established, the available bonuses are listed, and the teachers have access to the repository with the source code, thus facilitating the evaluation.

Unfortunately, interaction between teams is almost non-existent. Even if they have the same project theme, none of the project requirements specifies the need for inter-team communication. Each team has the freedom of using their chosen programming language, framework, or development program, so they might not feel the need to exchange ideas with other teams. Furthermore, not even the bonus multiplayer has to work between the implementations of different teams: It functions only at the intra-team level.

#### *4.7. Mixed Approach*

We previously described six different educational approaches for teaching software project management. Most of the mixed approaches from the existing literature are based on creating a balance between competitive and collaborative educational models. We propose a more advanced approach, by taking and combining several characteristics from our six different models, thus forming a new, mixed approach. Our software project management proposition for the mixed approach was the creation of an image processing system based on a voting strategy. More precisely, a system which receives a color image (for instance, resulted after scanning a book/newspaper) as input and transforms it into a binary image (black and white) to improve the legibility of the text. Optical character recognition (OCR) is applied afterwards in order to convert the binary image to digital text. The algorithm that is capable of recognizing the biggest number of characters is considered the best.

The organization of students in teams is more special than the one used in other approaches. The students are divided into subgroups and each subgroup contains four teams or “cells”, each one with a reduced number of students. Three of these teams are responsible for developing three different

conversion algorithms (“Binarization Algorithm Module—BAM”), while the fourth team must combine the three algorithms in a new one, better than all the initial ones (“Voting Binarization Algorithm Module”—VBAM). Several characteristics of distinct approaches that we mentioned previously are combined. For instance, individual roles must be covered by members of the same team: Research, development, and testing. They are encouraged to choose an appropriate project manager, who is capable of organizing weekly meetings and distributing tasks according to the team members’ competencies. Mixed competition—collaboration aspects are also present: Students must collaborate to obtain the best BAMs and VBAMs, respectively, which must work well together. On the other hand, there is permanent competition between teams, given the fact that all BAMs (and VBAMs, respectively) from all the subgroups must compete against each other. The pressure is extremely elevated, as the selected BAMs and VBAM will be used in a prestigious international competition related to document image binarization.

The evaluation process is specific to a mixed approach, where both individual and teamwork are essential. Therefore, two grades are considered: One is the individual grade and the other one is the project grade, given at the team level. The project grade is given at the team level and it considers both the performance of the individual conversion algorithm (BAM) and that of the VBAM obtained by the subgroup. The importance of the individual’s participation in the team project is emphasized, as the project grade is multiplied by a factor calculated based on the individual grade.

The mixed approach can bring a lot of advantages, by combining the positive aspects of previously tested models. Therefore, the collaborative effort and the competitive motivation can be extremely stimulating. The team organization is also innovative and efficient, as the reduced number of members in a team can lead to better organization, synchronization, and unitary management. The evaluation process is also correct and objective and can hardly be contested. It is reflected through concrete performances of the algorithms, accentuating the amount of work spent at the individual and team level. All teams receive the same project theme and the best final products are chosen in the end—this means that the failure of a certain team does not lead to the failure of the whole didactic project.

On the other hand, the idea of mixing opposite characteristics from several approaches can restrict the area of available project themes. They cannot always be in concordance with the students’ preferences or knowledge. Even if the didactical outcome is satisfactory, students might not perceive the application as being very useful and believe they are just being part of an educational need/experiment. The idea that the winning solutions will be presented to a prestigious international competition can represent an advantage or a disadvantage, based on each person’s character: Some of the students might feel extremely motivated, while others may not be used to handling such pressure.

## 5. Comparison and Results of the Proposed Approaches

### 5.1. Statistical Processing Across Multiple Years, How to Compare Apples with Oranges

One of this main paper’s objective is to offer an in-depth view of the proposed educational approaches, viewed from both sides of the software project management course in our university: Teachers and students. There is, however, a measurement that interests both parts in the evaluation process: The grades.

Since we are talking about an activity that spanned across 10 years, involving almost 900 students, 26 teaching assistants with different evaluation styles, and seven totally different practical approaches with different methods of grade computation, trying to compare the results obtained for the practical side, without any additional processing, is like comparing apples with oranges. In order to make the comparison of apples and oranges feasible, the data needs to be processed so that most of the subjective factors of evaluation, encountered during the years, will be reduced as much as possible. For this purpose, we make use of the observation that the most plausible grade distribution is a normal (Gaussian) one. This means that the data must be measured for symmetry and shape accuracy, and processed independently at the student batch level to align it as much as possible, within certain limits,

to the desired normal distribution. A batch of students is constructed from the collection of teams that perform under the same project requirements and are evaluated by the same teaching assistant.

In the statistical processing phase, the following notations and measures are used:

- The grades inside a batch of students:  $\{X_1 \dots X_N\}$ , where  $N$  is the number of students in the batch. The grades are positive real values in the range 0 (nothing) to 10 (perfect solution).
- The mean measures the average performance of a batch of students:

$$\text{Average} = \frac{\sum X_i}{N}, \text{ furtherly denoted } \bar{X}. \quad (1)$$

It is considered that the bigger the average is, the better the performance of the students.

- The standard deviation measures the amount of variation around the mean (which may be considered as an expected value), inside the students' batch:

$$\text{Standard Deviation} = \sqrt{\frac{\sum (X_i - \bar{X})^2}{(n - 1)}}, \text{ furtherly denoted } s. \quad (2)$$

The interpretation of the standard deviation is important regarding the outcome of one educational approach because we consider that bigger standard deviations imply better student separation in the grading process, thus resulting a more precise method of evaluation.

- The skewness measures the asymmetry of the distribution of grades, around the mean value, inside one batch:

$$\text{Skewness} = \frac{n}{(n - 1)(n - 2)} \sum \left( \frac{X_i - \bar{X}}{s} \right)^3 \quad (3)$$

Ideally, the skew must be zero, meaning that the grades inside a batch are perfectly balanced around the mean. However, this unlikely so it is desirable that the skew should be as low as possible, with negative values indicating the "left tail" is longer and the gross of the distribution is focused at the right, while positive values indicate the exact opposite.

- The kurtosis measures the peakedness (or flatness) of the grade distribution inside a batch, compared with the normal distribution. In fact, for the current processing, excess kurtosis is chosen instead because this is the measure that behaves like a distance indicator between the real (grade) distribution and the ideal (normal) one:

$$\text{Kurtosis} = \frac{n(n + 1)}{(n - 1)(n - 2)(n - 3)} \sum \left( \frac{X_i - \bar{X}}{s} \right)^4 - \frac{3(n - 1)^2}{(n - 2)(n - 3)} \quad (4)$$

Like in the case of skewness, the excess kurtosis should, ideally, be zero, thus indicating that the grades inside the batch follow the shape of the normal distribution. A negative value indicates a distribution of data that is called "platykurtic", or broadly shaped with thinner tails, while positive values indicate a so-called "leptokurtic" distribution, having a slender shape with fatter tails [42].

The standard deviation, skewness, and excess kurtosis statistical functions may have slightly different formulas, depending on their usage or the data size. Some variations are provided to serve as unbiased variants, however, in the aforementioned definitions, the Microsoft's Excel versions are used, because this is, perhaps, the most common software solution for table-structured data processing.

To statistically process the grades to ensure the inter-batches' relevance as much as possible, it is necessary to match as closely as possible the intra-batch distribution to resemble the normal one. For this purpose, we apply three processing steps:

- A linear preprocessing (translation-scaling) operation:  $Grade = Grade \times PreprocessScale + PreprocessTranslation$
- A power-law correction applied to the grading distribution:  $Grade = Grade^{CorrectionPower}$



- A linear post-processing (translation-scaling) operation:  $Grade = Grade \times PostprocessScale + PostprocessTranslation$

The processing steps are performed with very small variations around their neutral point: Both scales around 1.0, both translations around 0.0, and power around 1.0, using a brute-force solution searching in the multi-dimensional (*PreprocessScale*, *PreprocessTranslation*, *CorrectionPower*, *PostprocessScale*, *PostprocessTranslation*) space. The result is a per-batch tuple of values that minimizes both skewness and excess kurtosis, which ensures the closest resemblance of the intra-batch grade distribution with the normal distribution. The ranges of variation for each of the processing steps' parameters were determined experimentally, as a mean of the natural variations obtained from evaluating the same projects with the same grading schemes, but applied at different moments by the same teaching assistant, or at the same moment by different teaching assistants who were communicating with each other.

## 5.2. The Students' Results

Based on the previously explained statistical measures, we will analyze their meaning and impact on the students' results and performance. Therefore, we will draw some conclusions related to the efficiency of each approach. For each measure, we will consider which approach was the best, the second best, the worst, or the second worst.

For the independent approach, where each team received a different project theme, the following relevant statistical measures can be analyzed:

- The mean value of 7.09 was the worst of all seven approaches—the students' performance can, therefore, be interpreted as being the poorest.
- The grade range is the second best, varying from 1.4 to a perfect 10. A larger grade range can illustrate more accurately the differences between the knowledge acquired by the students, making the evaluation process more precise.
- The standard deviation is the second best, having a relatively high value. Combined with the large grade range, it can contribute to the easier separation of students in the grading process. We do not want to obtain a low standard deviation, as it can be very difficult to separate grades between very similar students, with close levels of knowledge. The value obtained for the standard deviation is encouraging, as one of our main concerns was the difficulty of applying an objective, balanced grading system.

In the anarchic approach, where students had to balance both individual and team strategy, the statistical measures for analyzing the grades can help us draw the following conclusions:

- The skewness has the second best value, which means that the values are relatively symmetrically distributed around the mean value.
- The grade range is the second worst, varying from 4.46 to 9.94. A narrower grade range cannot illustrate accurately the differences between the students' performances, making the evaluation process much more difficult.
- The kurtosis also has the second worst value, having a negative value. This means that the grade distribution is flat, "platykurtic", as opposed to the ideal, normal distribution. Even if the skewness indicated that the grades are distributed symmetrically around the mean value, they are insufficiently concentrated in a limited domain—they are concentrated on an interval which covers most of the grade range.

The mean value was 8.18, which represents the second best from all the approaches. This means that most students performed well, as individuals and at the team level.

- The standard deviation is the second worst, having a relatively low value. This means that the students' grades were distributed relatively close to the average value, which can make the evaluation process difficult, as no relevant differences between students can be established.

- The skewness has the worst value, which means that the values are not symmetrically distributed around the mean value. The high negative value of this measure indicates the fact that the gross distribution is concentrated on the right, with the majority of the grades being above the mean.
- The kurtosis also has the worst value, having a high positive value. This means that the grade distribution is steep, “leptokurtic”, as opposed to the ideal, normal distribution.
- The bad values of the skewness and kurtosis measures indicate that the evaluation process became more complicated and less accurate.

In the competitive approach, the separation between teams became maximal, a fact which can also be observed in the statistical measures. Among all the proposed approaches, the competitive one stands out in a large number of categories:

- The grade range is the best, varying from 1.02 to a perfect 10. This can make the evaluation process easier and more precise, as it illustrates more accurately the students’ performance.
- The standard deviation is also the best. This measure and the large grade range prove the effectiveness of the competitive approach in terms of objective and effective evaluation.
- The kurtosis has the best value, having a slightly negative value. The grades are almost perfectly concentrated in a limited domain, very close to the ideal, normal distribution.
- The mean value was 7.41, which represents the second worst value. This means that most of the students’ performances are not very good. A probable cause might be represented by the final rankings of the competition, based on points—there are considerable differences between first place and the others in terms of points’ allocation.
- The skewness was also the second worst, as the values are not symmetrically distributed around the mean. The high negative value of this measure means that the gross distribution is concentrated on the right, with more grades being above the mean.

The original approach concentrates on the creativity and artistry of each team. Even if it is a creative approach, we used the same statistical measures in order to draw some relevant conclusions regarding its effectiveness:

- The mean value is 8.54, which represents the best from all the approaches. This underlines the fact that most students performed great and they managed to use their creativity efficiently.
- The kurtosis is the second best, having a relatively low positive value. The grades are concentrated in a limited domain, very close to the normal distribution.
- The standard deviation is the worst. Its low value indicates the fact that most grades are distributed close to the mean value, which can harden the evaluation process, making it more subjective.
- The grade range is also the worst, varying from 5.94 to 10. A narrow grade range can make the evaluation process even more difficult and subjective.

In the synchronous approach, the accent is put on uniqueness and rigidity. For this approach, the statistical measures have shown medium results in almost all categories (neither the best, nor the worst). The skewness is the only parameter which stands out, being the best, which means that the grades’ values are symmetrically distributed around the mean value.

The mixed approach combines some characteristics taken from all the previously mentioned approaches. Amazingly, for the mixed approach, all the statistical measures show medium results in all categories. It does not stand out (but it is not the worst either) in any category, which means that the mixed approach can be placed in the center, as in our proposed MOSAICS model.

Figure 2 presents a comparison between all approaches in terms of skewness and kurtosis. In can easily be observed that the worst values (highest) are obtained for the synergistic approach, while the best ones are for the synchronous approach and competitive approach (kurtosis). Figure 3 shows a comparison between the mean values and grade ranges for all approaches.

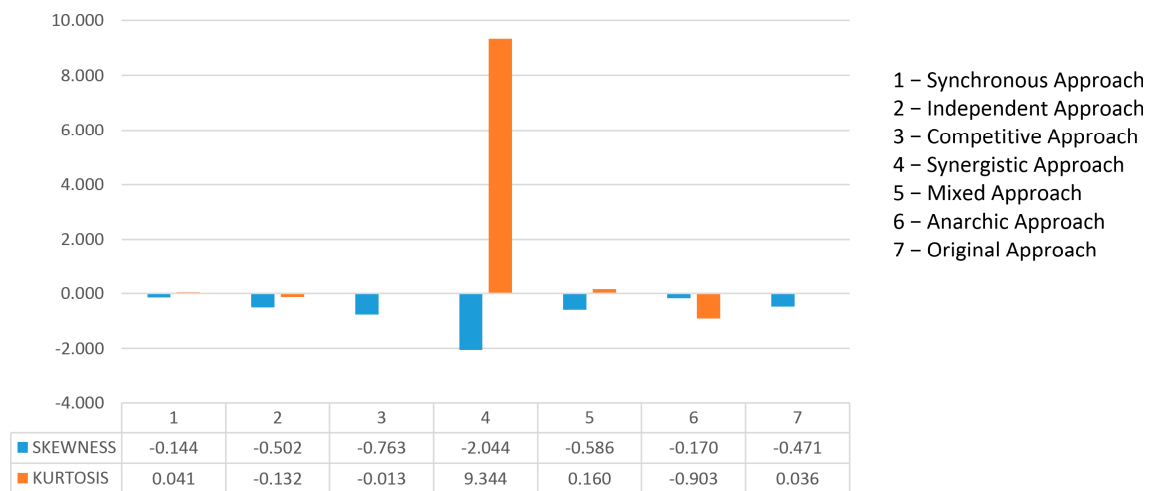


Figure 2. Skewness and kurtosis values.

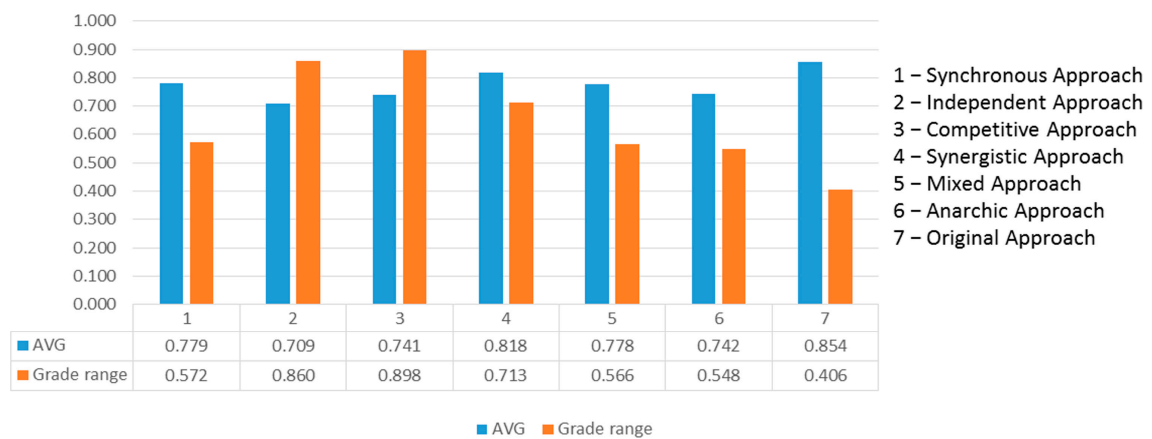


Figure 3. Mean and grade ranges.

## 6. Discussion

### 6.1. Peer-To-Peer Advantages and Disadvantages. The Quest for Perfection

To emphasize the advantages and disadvantages of each approach, we compare two approaches, which are situated on the same axis, on the opposite sides. Relevant criteria were selected, including also the previously mentioned statistical measures.

Firstly, we compare the independent and anarchic approaches, situated on the “Intra-team cohesion axis” (Table 1).

Table 1. Independent approach vs. anarchic approach.

| Criteria                 | Independent  | Anarchic   |
|--------------------------|--|--|
| Intra-team cohesion axis | oriented towards “Cooperation”   | oriented towards “Rivalry”   |
| Team structure           | organized teams, where each person has a well-determined role  | flexible teams, as members might be asked to switch teams during the competition                       |
| Hierarchy                | creates a valid project management hierarchy   | decentralized, no actual hierarchy; a team manager exists, but only as a mediator between team members |
| Project scope            | each team receives a different project theme, simulating various scenarios, requiring different competencies or approaches | all the teams receive the same project   |
| Project Allocation       | personalized project themes based on the students’ preferences or interests  | one common project theme doesn’t take into account the students’ preferences or skills                 |

Table 1. Cont.

| Criteria                   | Independent  | Anarchic   |
|----------------------------|--|--|
| Competitiveness            | the competition between teams is diminished (different project themes) and delayed (introduced later, when the teams must present and “sell” their applications)   | bitter competition between teams and individuals   |
| Team-Individual Connection | significant, as the individual interest and the team interest are strictly interdependent  | individual interest and team interest can be completely opposite (changes can occur from a collaborative model, to a competitive one or to uncontrollable anarchy)   |
| Team-Individual Relation   | each member has to acquire their <b>own</b> knowledge (through independent learning) and apply their own skills, for a <b>mutual</b> benefit   | each member must find an optimal balance between a secret individual strategy and the benefit of the team  |
| Communication              | <b>improved</b> intra-team communication (homogenous teams); <b>minimized</b> inter-team communication   | tries to assure better intra-team and inter-team communication, but the result is chaotic (because of the decentralized approach)  |
| Evaluation                 | individually (during the semester) and at the team level (at the end of the semester)  | based on the individual success (the number of wins of the temporary team) and based on the team success (the number of wins of the original team).  |
| Statistical Measures       | worst <b>mean value</b> (7.09) means poorest students’ performance<br>good <b>standard deviation means</b> easy and efficient separation of students in the grading process<br>the large <b>grade range</b> (1.4–10) illustrates accurately the differences between students, makes the evaluation easier<br>medium values for <b>skewness</b> and <b>kurtosis</b> | low <b>mean value</b> (7.42) means poor students’ performance<br>insignificant <b>standard deviation</b><br><b>the narrow grade range</b> (4.46–9.94) cannot illustrate too accurately the differences between students, makes the evaluation more difficult<br>good <b>skewness</b> means grades are relatively symmetrically distributed around the mean value; bad <b>kurtosis</b> (grades insufficiently concentrated in a limited domain) |

If we compare the synergistic and competitive approaches, we can observe the advantages and disadvantages of two approaches situated on the “Inter-team separation” axis (Table 2).

Table 2. Synergistic approach vs. competitive approach.

| Criteria                   | Synergistic   | Competitive  |
|----------------------------|---|--|
| Inter-team separation axis | oriented towards “Collaboration”  | oriented towards “Competition”   |
| Team structure             | organized teams, where each member chooses a specific role  | organized teams, where each member chooses a specific role   |
| Hierarchy                  | each team is encouraged to choose a manager to coordinate their activity  | each team must choose a project manager—his authority and competency are extremely important   |
| Project scope              | each team works on a different part of the same project   | each team receives the same project and comes up with different solutions  |
| Competitiveness            | emphasizes the collaboration process, not the competition between teams, which is minimum   | permanent competition between teams: during the project development and at the end, when the best final products win   |
| Team-Individual Connection | significant, as the individual interest and the team interest are strictly interdependent   | significant, as the individual interest and the team interest are strictly interdependent  |
| Communication              | both intra-team and inter-team communication are very important and well represented  | the intra-team communication is well represented, while the inter-team communication is almost non-existent  |
| Evaluation                 | individually and at team level, encourages each person to be implicated; the evaluation is also influenced by the work of other teams which makes it complicated and less accurate  | individually, at team level and based on the team manager’s appreciation; objective and precise (based on points, rankings)  |
| Statistical Measures       | second best <b>mean value</b> (8.18) represents good students’ performance<br>bad <b>standard deviation</b> represents that the evaluation process is difficult<br>worst <b>skewness</b> and <b>kurtosis</b> represents that the evaluation process was more complicated and less accurate.<br>more narrow <b>grade range</b> (2.87–10) represents a more difficult process to compare the students’ performances | low <b>mean value</b> (7.41) represents poor students’ performance<br>great <b>standard deviation</b> represents an objective and effective evaluation<br>bad <b>skewness</b> , but good <b>kurtosis</b> represents that more grades were above mean, but closer to the normal distribution<br>very large <b>grade range</b> (1.02–10) illustrates accurately the differences between students |

The original and synchronous approaches must be compared based on the educational model fuzziness axis, confronting creative and rigid educational models (Table 3).

**Table 3.** Original approach vs. synchronous approach.

| Criteria                         | Synergistic  | Competitive   |
|----------------------------------|--|---|
| Educational model fuzziness axis | oriented towards “Flexibility”   | oriented towards “Rigidity”   |
| Team structure                   | organized teams, but roles are not designated by teachers  | organized teams, where each member has a specific role  |
| Hierarchy                        | those with the most original ideas have the opportunity of stepping up and taking the responsibility of leadership   | strict hierarchy, with pre-established roles  |
| Project scope                    | each team receives the same project theme, but with a high degree of freedom   | each team receives the same project and must focus on “First make it run, then make it run right”   |
| Duration                         | long, during the whole semester (14 weeks)   | short, until the middle of the semester (7 weeks)   |
| Competitiveness                  | highly encouraged; the winning team gets the Van’Gogu trophy and the pride of having their creation displayed on the SPM official page   | not significant for this approach   |
| Creativity                       | encourages creativity, originality and artistic spirit<br>students are not allowed to use external resources or ideas<br>integrate a certain degree of randomness                          | diminished by rigidity and strictness<br>“we shouldn’t reinvent the wheel” (any programming language, any type of framework, library or code fragments, any platform are permitted) |
| Communication                    | intra-team communication is very important and well represented (organization, filtering creative ideas)   | the intra-team communication is at a medium level; interaction between teams is almost non-existent (not even for the multiplayer)  |
| Technical difficulty             | a technically complicated approach doesn’t guarantee the attainment of a visually pleasing image<br>a good idea can lead to appropriate results, even if it is not technically challenging | challenging: obtaining a working game, as fast as possible<br>additional functionalities are encouraged leading to bonuses  |

All approaches are compared based on various criteria, relevant for most of the educational approaches of our MOSAICS model (see Table A1 in Appendix A).

### 6.2. The Students’ Feedback

The MOSAICS approaches were evaluated and interpreted by the students who took part in them. By receiving their positive and negative comments, we can take some characteristics of each approach and put them together in order to create the most suitable one for the students’ preferences.

In the beginning, the competitive approach was well received by most of the students, who considered that it was a good simulation of a real-world working environment, a fact which kept them motivated. On the other hand, some characteristics of the approach were not appreciated by the students. They thought there were too many members in a team, as compared to the project’s difficulty. Furthermore, the idea of having the best team’s code shared to all teams after a milestone felt unfair and discouraging, as this can encourage lazy students or lead to difficult bugs in the code. The students believed that a project manager was not necessary for this type of project, as their self-organization would have been more successful.

The synergistic approach showed positive changes from the previous, competitive one. The results got better, students felt more motivated, and contributed greater effort, led by the desire to make their teammates proud. Students thought that collaborative work helps them acquire both technical and social skills, essential in software project management.

The anarchic approach led to low final grades, which displeased students. They did not think that the anarchic principles were the main reason of their low results, as some unexpected technical difficulties influenced the outcome of their projects. The difficulty of testing the final projects in a virtual environment decreased the students’ motivation. This can demonstrate that working for an attractive project can be more stimulating than working for receiving a better grade. In addition, the anarchic principles should be applied from younger ages, in order to help people become accustomed

with this type of teamwork. As there is no project manager, teams should be made of people who know each other well, in order to facilitate communication and collaboration.

The independent approach received a lot of positive feedback, as each student was able to choose a suitable theme for their taste. There were, however, some mixed opinions, as some of the students stayed motivated and worked hard, while others lost their enthusiasm, maybe because they were going through a very busy time period.

For the mixed approach, a student's words speak for themselves: "It was a new and great experience to work side by side with other colleagues in developing an algorithm. The fact that we had the change to embrace collaboration, competition and gamification in the same project was more than rewarding."

## 7. Conclusions

### 7.1. The Relevance of the Study

The authors consider the presented study relevant and reasonably accurate and reproducible, due to the following facts:

- Our study for creating the MOSAICS educational model spanned 10 years of software project management courses, from 2008 until 2018;
- For each proposed approach, a significant number of students were involved, interesting project themes were elaborated, and general rules and specifications were created. In total, almost 900 students from different years, inherently having different backgrounds and using multiple technologies, contributed to the study;
- A total of 26 different teaching assistants were employed, with an average of seven per proposed educational approach. They, obviously, had totally distinct teaching and evaluation styles, thus decreasing the overall subjectivity of the study;
- To compare (with a meaning) totally different approaches and situations in the calibration process, we used the controlled alignment of the most relevant statistical measures in grade distribution: Standard deviation, skewness, and kurtosis.

In the end, the knowledge and observations gained through this experiment led to a complete and general educational model, called MOSAICS.

### 7.2. Authors' Recommendations

Since all the proposed educational approaches were proven to possess certain strong points, but also several weaknesses, and all of them appeared to complement each other nicely in MOSAICS, it appears that a complete training in the practical side of software project management will have to switch certain projects developed in each of the proposed model areas. This is not an easy task because every proposed educational approach, to fully develop its core essentials, must be extended across a significant period of time. Since any course (and "Software Project Management" is no exception) is limited to a specific amount of time (usually one semester), the main difficulty will be to formulate seven specific projects, each of them belonging to the seven distinct educational approaches consistent with the MOSAICS model's origin and boundaries, with clearly-defined outcomes and time-boxing constraints. It is not impossible, but it will prove a solid challenge to any teacher's imagination, to spawn seven projects' descriptions, which will both appeal to students and also make strong, balanced use of their soft and hard skills.

### 7.3. Future Work

Future efforts will be directed towards changing the structure of the "Software Project Management" course in the practical side. In the previous 10 years, the seven aforementioned educational approaches have been tried for one or more semesters, and now it will be the appropriate

time to implement them in seven mini-projects chained across one-semester. This will allow the students to tackle all facets of interaction in both inter and intra-team form, and to experience different degrees of fuzziness in product characteristics, development frameworks, interfaces, requests, acceptance, milestones, and deliverables, thus completely and extensively preparing them for their future real-world jobs.

**Author Contributions:** Conceptualization, C.-A.B.; Data curation, C.-A.B.; Investigation, C.-A.B. and I.-C.S.; Methodology, C.-A.B.; Project administration, C.-A.B.; Resources, I.-C.S.; Software, C.-A.B.; Validation, C.-A.B.; Writing—original draft, C.-A.B. and I.-C.S.; Writing—review & editing, C.-A.B. and I.-C.S.

**Funding:** This research received no external funding.

**Acknowledgments:** The first author, “Software Project Management” course coordinator, would like to thank teaching assistant team leader Răzvan Deaconescu, teaching assistants: Adrian Cătu, Cătălina Manolache, Mihai Zaharescu, Daniel Rosner, Ciprian Coman, Andrei Maruseac, Octavian Saca, Alexandra Ciortea, Mihai Soare, Mihai Zamfirescu, Lucian Grijincu, Alexandru Moşoi, Monica Bugeanu, Alexandru Calotoiu, Andreea Florescu, Ştefan Apăteanu, Costin Speciac, Raluca Dobroiu, Oana Căplescu, Matei Neagu, Laura Antonache, Alex Brebu, Carina Mogoş, Giorgiana Vlăsceanu for their great support, feedback and assistance.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

Table A1. General comparison of MOSAICS approaches.

| Criteria                   | Independent  | Anarchic   | Synergistic  | Competitive   | Original   | Synchronous  | Mixed  |
|----------------------------|--|--|--|---|--|--|--|
| Team structure             | organized teams, where each person has a well-determined role  | flexible teams, as members might be asked to switch teams during the competition   | organized teams, where each member chooses a specific role   | organized teams, where each member chooses a specific role  | organized teams, but roles are not designated by teachers  | organized teams, where each member has a specific role   | subgroups and each subgroup contains four teams  |
| Hierarchy                  | creates a valid project management hierarchy   | decentralized, no actual hierarchy; a team manager exists, but only as a mediator between team members   | each team is encouraged to choose a manager to coordinate their activity   | each team must choose a project manager—his authority and competency are extremely important                                | those with the most original ideas have the opportunity of stepping up and taking the responsibility of leadership                     | strict hierarchy, with pre-established roles   | each team has a project manager which is capable of organizing the project according to the team members' competencies |
| Project scope              | each team receives a different project theme, simulating various scenarios, requiring different competencies or approaches                                       | all the teams receive the same project   | each team works on a different part of the same project  | each team receives the same project and comes up with different solutions   | each team receives the same project theme, but with a high degree of freedom   | each team receives the same project and must focus on "First make it run, then make it run right".                                 | three teams responsible for conversion algorithms; the fourth team must combine the three algorithms in a new one      |
| Competitiveness            | the competition between teams is diminished (different project themes) and delayed (introduced later, when the teams must present and "sell" their applications) | bitter competition between teams and individuals   | emphasizes the collaboration process, not the competition between teams, which is minimum  | permanent competition between teams: during the project development and at the end, when the best final products win        | highly encouraged; the winning team gets the Van Gogh trophy and the pride of having their creation displayed on the SPM official page | not significant for this approach  | highly encouraged; the winning solutions will be presented to a prestigious international competition                  |
| Team-Individual Connection | significant, as the individual interest and the team interest are strictly interdependent  | individual interest and team interest can be completely opposite (changes can occur from a collaborative model, to a competitive one or to uncontrollable anarchy) | significant, as the individual interest and the team interest are strictly interdependent  | significant, as the individual interest and the team interest are strictly interdependent                                   | not significant for this approach  | not significant for this approach  | significant, as the individual interest and the team interest are strictly interdependent                              |
| Communication              | <b>improved</b> intra-team communication (homogenous teams); <b>minimized</b> inter-team communication   | tries to assure better intra-team and inter-team communication, but the result is chaotic (because of the decentralized approach)                                  | both intra-team and inter-team communication are very important and well represented   | the intra-team communication is well represented, while the inter-team communication is almost non-existent                 | intra-team communication is very important and well represented (organization, filtering creative ideas)                               | the intra-team communication is at a medium level; interaction between teams is almost non-existent (not even for the multiplayer) | essential, at both team and subgroup level   |
| Evaluation                 | individually (during the semester) and at team level (at the end of the semester)  | based on the individual success (the number of wins of the temporary team) and based on the team success (the number of wins of the original team).                | individually and at team level, encourages each person to be implicated; the evaluation is also influenced by the work of other teams, meaning it is complicated and less accurate | individually, at team level and based on the team manager's appreciation; objective and precise (based on points, rankings) | subjective, based on the artistic value of each resulting image ("Beauty lies in the eye of the beholder")<br>a jury of SPM teachers   | objective (rules clearly established, bonuses are listed, teachers have access to the repository with the source code)             | correct, objective, can hardly be contested; reflected through concrete performances of the algorithms                 |



## References

1. Sharan, S. Cooperative Learning in Small Groups: Recent Methods and Effects on Achievement, Attitudes, and Ethnic Relations. *Rev. Educ. Res.* **2009**, *50*, 241–271. [[CrossRef](#)]
2. Seifert, K.; Fenster, A.; Dilts, J.A.; Temple, L. An Investigative, Cooperative Learning Approach to the General Microbiology Laboratory. *CBE Life Sci. Educ.* **2009**, *8*, 147–153. [[CrossRef](#)] [[PubMed](#)]
3. Tahmoush, D.; Fouché, S.; McMaster, S.; Purtilo, J. Enhancing Software Project Management Courses with Industry Participation, Case Study in Cooperative Learning through Industrial Partnership. In Proceedings of the FECS'09: Computer Science and Computer Engineering, Las Vegas, NV, USA, 13–16 July 2009.
4. Agile Based Learning: What Is It and How Can It Change Education? Available online: <http://www.opencolleges.edu.au/> (accessed on 20 December 2018).
5. Tan, C.-H.; Tan, W.-K.; Teo, H.-H. Training Students to Be Agile Information Systems Developers: A Pedagogical Approach. In Proceedings of the Sigmis CPR Conference on Computer Personnel Research, Charlottesville, VA, USA, 3–5 April 2008.
6. Devedzic, V.; Milenkovic, S.R. Teaching Agile Software Development: A Case Study. *IEEE Trans. Educ.* **2011**, *54*, 273–278. [[CrossRef](#)]
7. Smith, B.L.; MacGregor, J.T. *What Is Collaborative Learning?* Washington Center for Improving the Quality of Undergraduate Education: Washington, DC, USA, 1992.
8. Chu, S.K.W.; Zhang, Y.; Chen, K.; Chan, C.K.; Lee, C.W.Y.; Zou, E.; Lau, W. The effectiveness of wikis for project-based learning in different disciplines in higher education. *Internet Higher Educ.* **2017**, *33*, 49–60. [[CrossRef](#)]
9. Chen, C.-Y.; Chong, P. Software engineering education: A study on conducting collaborative senior project development. *J. Syst. Softw.* **2011**, *84*, 479–491. [[CrossRef](#)]
10. Radygin, V.Y.; Lukyanova, N.V.; Kupriyanov, Y. LMS in university for in-class education: Synergy of free software, competitive approach and social networks technology. In Proceedings of the AIP Conference, Moscow, Russia, 7–8 December 2015.
11. Sheth, S.; Bell, J.; Kaiser, G. *A Competitive-Collaborative Approach for Introducing Software Engineering in a CS2 Class*; Department of Computer Science, Columbia University: New York, NY, USA, 2011.
12. Flores, N.H.; Paiva, A.C.R.; Letra, P. Software Engineering Management Education through Game Design Patterns. *Procedia Soc. Behav. Sci.* **2016**, *228*, 436–442. [[CrossRef](#)]
13. Navarro, E. *SimSE: A Software Engineering Simulation Environment*; University of California: Oakland, CA, USA, 2006.
14. Caulfield, C.; Veal, D.; Maj, S.P. Teaching Software Engineering Project Management—A Novel Approach for Software Engineering Programs. *Mod. Appl. Sci.* **2011**, *5*. [[CrossRef](#)]
15. Calderón, A.; Ruiz, M. Bringing Real-life Practice in Software Project Management Training Through a Simulation-based Serious Game. *CSEDU* **2014**, *2*, 117–124.
16. Straub, J.; Kerlin, S. An Experiential Education Approach to Teaching Software Project Management. In Proceedings of the Midwest Instruction and Computing Symposium, Verona, WI, USA, 25–26 April 2014.
17. Bollin, A.; Hochmuller, E.; Samuelis, L. Teaching Software Project Management using Simulations—The AMEISE Environment: From Concepts to Class Room Experience. In Proceedings of the IEEE 25th Conference on Software Engineering Education and Training, Nanjing, China, 17–19 April 2012; pp. 85–86.
18. Riazi, A.M.; Candlin, C.N. Mixed-methods research in language teaching and learning: Opportunities, issues and challenges. *Lang. Teach.* **2014**, *47*, 135–173. [[CrossRef](#)]
19. De Lucia, A. *Teaching Software Engineering and Software Project Management: An Integrated and Practical Approach*; Department of Management & Information Technology, University of Salerno: Fisciano, Italy, 2012.
20. So, C.; Ng, M.A. *Positivism & Constructivism, Epistemology and the Design of Learning Environments*; CSULA's EDIT 451; SlideShare Presentation: Los Angeles, CA, USA, 2010.
21. Barney, A. What Is Your Paradigm? eResearch Methods, University of Southampton. Available online: [http://www.erm.ecs.soton.ac.uk/theme2/what\\_is\\_your\\_paradigm.html](http://www.erm.ecs.soton.ac.uk/theme2/what_is_your_paradigm.html) (accessed on 20 December 2018).
22. Dreambox Learning, What Is Individualized Learning? Available online: <http://www.dreambox.com/individualized-learning> (accessed on 9 January 2019).
23. Rennie Center Education Research & Policy. *Charting a Path to the Future through Individualized Learning Plans*; Rennie Center Education Research & Policy: Boston, MA, USA, 2016.

24. Welsh, S. Goal-Oriented Personal Learning Plans and Their Effect on Student Aspirations Regarding Postsecondary Education. Bachelor's Thesis, Boston College, Chestnut Hill, MA, USA, 2005.
25. British Council Schools Online. *Dreams and Teams Education Pack*; Report; British Council Online: London, UK, 2015.
26. Meyer, B.; Haywood, N.; Sachdev, D.; Faraday, S. *What Is Independent Learning and What Are the Benefits for Students?* Research Report 051; Department for Children, Schools and Families: London, UK, 2008.
27. Meyer, W.R. Independent learning: A literature review and a new project. In Proceedings of the British Educational Research Association Annual Conference, London, UK, 1–4 September 2010.
28. DeLong, S. *Teaching Methods to Encourage Independent Learning and Thinking*; Center for Teaching Excellence, United States Military Academy: West Point, NY, USA, 2009.
29. Petty, G. Teaching students independent learning. *InTuition* **2013**, Winter 2013/14, 30–31.
30. Oxford English Dictionary. Available online: <https://en.oxforddictionaries.com/> (accessed on 20 December 2018).
31. Suissa, J. *Anarchism and Education: A Philosophical Perspective*; PM Press: Oakland, CA, USA, 2010.
32. Wagreich, S. A Billion Dollar Company with No Bosses. Yes, it Exists. Available online: <http://www.inc.com/> (accessed on 20 December 2018).
33. Australian Institute for Teaching and School Leadership AITSL. *The Essential Guide to Professional Learning: Collaboration*; AITSL: Melbourne, Australia, 2011.
34. Stoner, J.L. Let's Stop Confusing Cooperation and Teamwork with Collaboration, Jesse Lyn Stoner on Leadership Blog Articles. Seapoint Center for Collaborative Leadership. Available online: <http://seapointcenter.com/cooperation-teamwork-and-collaboration/> (accessed on 8 April 2016).
35. Verhoeff, T. *The Role of Competitions in Education*; Faculty of Mathematics and Computing Science: Eindhoven, The Netherlands, 1997.
36. The Office for Standards in Education. *Learning: Creative Approaches that Raise Standards*; Office for Standards in Education, Children's Services and Skills: London, UK, 2010.
37. The Glossary of Education Reform, Synchronous Learning. Available online: <https://www.edglossary.org> (accessed on 20 December 2018).
38. Schwier, R.A.; Balbar, S. The interplay of content and community in synchronous and asynchronous communication: Virtual communication in a graduate seminar. *Can. J. Learn. Technol.* **2002**, *28*, 2–6. [[CrossRef](#)]
39. Krippendorff, K. *Multivariate Techniques in Human Communication Research*; Monge, P.R., Cappella, J.N., Eds.; Academic Press: New York, NY, USA, 1980; pp. 259–308.
40. Rousseeuw, P.J. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Comput. Appl. Math.* **1987**, *20*, 53–65. [[CrossRef](#)]
41. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [[CrossRef](#)]
42. Joanes, D.N.; Gill, C.A. Comparing Measures of Sample Skewness and Kurtosis. *Statistician* **1998**, *47*, 183–189. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).