

Enhancing students' motivation to learn software engineering programming techniques: a collaborative and social interaction approach

Ricardo Rodrigues Nunes^{1,2}, Daniela Pedrosa^{1,3}, Benjamim Fonseca^{1,2}, Hugo Paredes^{1,2}, José Cravino^{1,3}, Leonel Morgado^{2,4}, Paulo Martins^{1,2}

¹ Universidade de Trás-os-Montes e Alto Douro (UTAD), Vila Real, Portugal
{rrnunes,dpedrosa,benjaf,hparedes,jcravino,pmartin}@utad.pt

² INESC TEC, Porto, Portugal

³ Research Centre Didactics and Technology in Education of Trainers (CIDTFF), Portugal

⁴ Universidade Aberta, Lisbon, Portugal

leonel.morgado@uab.pt

Abstract. To motivate students to study advanced programming techniques, including the use of architectural styles such as the model–view–controller pattern, we have conducted action research upon a project based-learning approach. In addition to collaboration, the approach includes students' searching and analysis of scientific documents and their involvement in communities of practice outside academia. In this paper, we report the findings of second action research cycle, which took place throughout the fourth semester of a six-semester program. As with the previous cycle during the previous academic year, students did not satisfactorily achieve expected learning outcomes. More groups completed the assigned activities, but results continue to reflect poor engagement in the communities of practice and very low performance in other learning tasks. From the collected data we have identified new approaches and recommendations for subsequent research.

Keywords: motivation; learning programming; collaboration; social interaction; communities of practice; project-based learning; problem-based learning.

1 Introduction

For students following a software engineering study program, learning object-oriented programming approaches for system development with well-structured coding is a complex challenge [1,2]. During introductory programming courses, typically students become able to develop small programs, as well as adapt and combine pieces of existing code, but they do not clearly understand the importance of writing well-structured code from pre-existing structures such as frameworks, libraries, and application programming interfaces (API) [3]. In more advanced programming situations - for example, involving the use of architectural styles such as model–view–controller (MVC) [4,5] - students need to develop a set of complex skills [3]. Furthermore, besides the programming skills required to apply such best prac-

tices during system development, students also need to develop social skills in order to collaborate with other developers as part of the teamwork-based process for developing large, complex software systems.

Literature addressing engineering education has reported that current learning approaches do not align with the professional practice required by the labor market [6,7]. These approaches are narrowly focused upon the acquisition of technical knowledge supported by heavy workloads and promote a meritocracy of difficulty-based belief system instead of prioritizing active learning and integrating knowledge, skills more aligned with professional realities [8,9,10,11].

The pedagogical context in which students learn influences their engagement and resolve to achieve learning outcomes [12,13], and much research has examined approaches to the above described problem employing project-based learning (PBL) and teamwork environments [14,15,16]. While engineering problems are designed so that multiple solutions of varying mathematical and scientific sophistication are possible, teamwork skills and both oral and written communication skills used to model these problems are essential to the success of future engineers [8].

2 Background

The PBL context may impact students' motivation regarding their development of feelings of autonomy, competence, and relatedness. Teachers who use PBL methodologies assist students to overcome difficulties that can negatively influence their motivation such as team composition or task difficulty [16]. PBL and its derivatives have been reported in literature addressing engineering education to constitute a methodological approach that can promote and maintain students' motivation [17,18,19], as well as develop their situational interests [20].

Pascual [21] has described an extension of project-oriented learning to increase the social development of knowledge and learning. His approach aims to maximize students' opportunities for sharing knowledge with professionals in order to unite academia and communities of practice. Such proposals are based on theories of motivation focused on enhancing students' intrinsic motivation by creating conditions that can meet their needs of relatedness. Several activities were developed during Pascual's study, including meetings between communities of students and maintenance engineers, scholastic and recreational activities both on and beyond campus, and the development of a web-based decision support system. The author hypothesized that this multimodal approach increases active learning and social relations, and results identified that enhanced intrinsic motivation, thereby confirming that communities of practice and relatedness needs are relevant factors for learning outcomes. For instance, students and professionals can meet in online environments (e.g., virtual worlds) and, therefore, students can receive constructive feedback that helps them to clarify their doubts [22]. Other psychological needs such as autonomy and competence are also related to PBL and other approaches, as well as favor social interaction to promote active learning and self-study with engineering coursework [23,24].

In this study, we posit that learning environments based on communities of practice can allow students to become motivated and take advantage of the expertise of experienced (professional) programmers in order to recognize the value of better code organization. From this

perspective, personal, behavioral, and environmental factors are related in a dynamic way that influences students' feelings regarding the skills necessary to overcoming challenges that arise during their learning [25,26]. In this process, we thus took into account students' social and cognitive factors.

3 The Course Context, Approach, and Assignments

3.1 Course context

The present action research effort was developed in the course Programming Methodologies III (PM3) which is part of the fourth semester of the undergraduate program in Informatics Engineering at Portugal's Universidade de Trás-os-Montes e Alto Douro (UTAD). Before reaching this stage of the undergraduate program, students participated in other courses, with coursework addressing programming techniques and concepts, including computational logic, basic procedural programming, structured procedural programming, object-oriented programming, assembly programming, and introductory concurrent programming. They also participated in two semester-long "laboratory" projects (i.e., tutored development of a project), based on structured programming and object-oriented programming techniques, respectively. Alongside PM3, students are attended a course concerning algorithms and developed a semester-long laboratory project where they should apply the concepts being learned in PM3 [27].

The goal of PM3 is to introduce large-scale programming concepts, which is a learning objective of the ACM / IEEE CSC [28]. In PM3, students progress toward working with the MVC architectural style, which essentially proposes a structural division of programs among three blocks: the model (i.e., program state), the view (i.e., output), and the controller (i.e., program flow). In PM3, the original proposal of the MVC style [4] which handles input in the controller, is contrasted with a more recent one [5], which handles input in the view.

When students arrive at this mid-program level, they are not entirely motivated to attain the long-term benefits of a more structured and manageable code organization [29]. This reality, if not tackled until graduation, would leave them unprepared for the labor market, for which such skills are essential [30]. Our research goal is to address this problem. We thus hypothesize that students are not motivated and do not recognize the importance of better code organization due to their inexperience with team-based approaches involved in long-term software development.

3.2 Approach

We conducted a blended-learning approach during the second semester of the 2010–2011 academic year within the PM3 course, in which students confronted problems they would have to solve in groups over 8 weeks. The first approach has been described in detail in an earlier paper [29]. In the first action research cycle, a course assignment on software architecture styles was used. Students had to study a problem, develop an approach, and discuss it online with programmers in communities of practice and / or social networks. The expectation was that students would find motivation for their studies, both because of their contact with the developer communities, and because of having to study and reflect on their problems

well enough to be able to discuss them with the members of these communities. Most students failed to achieve successful outcomes. Only 7 groups out of the 19 groups that participated showed some output during the various phases of the project, and only 4 groups had positive feedback from their involvement with professional developers and online communities of practice. The students indicated that their main difficulties were in understanding what was being asked and in finding professionals and communities in the field. In relation to the assignment, lack of motivation, lack of feedback on the development of the work and lack of time were identified as the main problems. Finally, most students considered the current assignment appropriate. We was changed based on the analysis of learning outcomes and observational data. Here we describe the resulting second approach.

The new approach was implemented during the second semester of the 2011–2012 academic year. The project length was increased to 13 weeks (i.e., the full semester minus the entry week), and learning activities were more time-structured (i.e., weekly tasks and check-points). Two tutors became available to support students via email, instant messaging (MSN and GTalk), Facebook, and Moodle fora. We also scheduled face-to-face meetings with students, either individually or in groups, in the case that they had difficulty with the tasks. We additionally changed the online environment (PBworks wiki platform) chosen in the first approach by Moodle. Moodle allowed us to separate all of the activities into modules over several weeks. This arrangement allowed the teacher and the tutors to better monitor the development of the assignment.

3.3 Assignments

The assignments presented to students involved solving a specific problem using a software architecture in order to stimulate and foster advanced programming skills in students via their participation in communities of practice and their analysis of scientific and technical documents. The assignments required students to develop of a written document explaining in detail the coding approaches used to apply an MVC-related architectural style involving different frameworks, libraries, and/or specific APIs. Our approach entailed providing a generic assignment framework or meta-assignment to be instantiated differently for each group of students (Table 1). The aim was for students to render into concrete terms (i.e., the coding approaches with specific libraries) the abstract concepts of the MVC style.

Table 1. Sample assignment instantiations

Groups	Description
G1	Write a detailed document explaining how to use the MVC architectural style to develop applications using OpenSimulator and/or Second Life virtual worlds as a user interface employing the libOpenMetaverse library. The document should include specific examples of implementations to illustrate the explanation.
G2	Write a detailed document explaining how to use the MVC architectural style to develop applications using the Windows Phone Application Platform employing the XNA framework. The document should include specific examples of implementations to illustrate the explanation.

G3	Identical to Group 2, but with the Silverlight framework.
-----------	---

With 95 students, 21 groups were formed, most of which consisted of five students, though two groups had four students, one group had three, and two groups had two. The assignment instantiations were made available via UTAD's Moodle e-learning platform. In addition to the assignments, set weekly tasks and documents to be completed by students were also available. To support students' development of the activities, we created and provided an online example as a guide for what was expected.

We created two Moodle fora: (1) questions and suggestions, including generic messages exchange; and (2) task-related notices. Students also had access to slides from classes with included audio explanations by the course professor.

Task submissions were made via the administrative teaching support information system, called SIDE [31]. Submissions were weekly for individual and group tasks, and activities were developed in three complementary phases (Table 2).

Table 2. The assignment phases

Phases	Tasks
P1	Search for literature addressing the assigned topic; take reading notes; contact professionals and communities of practice online related to the assigned theme; adapt knowledge gained from the context and professional styles.
P2	Become involved with professionals and communities of practice online; debate the assigned topic, either asynchronously or synchronously; devise tentative approaches for developing and solving the assigned topic; present, debate, and further develop assigned topics.
P3	Provide an online report of results and present them in class.

In P1, the objectives were to increase knowledge of the problem domain and encourage participation and discussion in communities of practice, albeit not yet discussions related to the assigned problem. P1 took place over 3 weeks (Weeks 1–3). During Week 1, two individual tasks were undertaken: taking reading notes about scientific and technical literature related to the assigned theme and getting in touch with professionals in communities of practice, in order to understand the communities' culture and present themselves. Week 2 also consisted of two individual tasks: summarizing other group members' notes and attempting to help on a generic problem posted by any member of the selected community of practice (not related to the assignment theme), in order to contribute to the community. Week 3 involved a group task and an individual task; while the former consisted of developing a summary of all scientific and technical literature found by the group about the assigned theme, the individual task involved each student's helping to solve another generic problem submitted by a member of the community of practice, primarily in order to strengthen his/her identity within the community.

In P2, students were expected to develop effective contact within communities of practice, now debating the assigned topic. This phase lasted for 5 weeks (Weeks 4–8). During Week 4, students as a group identified questions remaining after P1, after which each student posed

his/her question to the community of practice in order to promote discussion about the assigned topic. During Week 5, it was suggested that groups should expand their discussions with new questions and ideas, after which during Week 6 they as a group discussed results obtained in the communities and drafted a report addressing the assigned theme. Week 7 consisted of an individual task in which each student published his or her reflection based on the group's drafted report in order to again generate feedback within communities of practice. Lastly, during Week 8, each student and then each group had to finalize a report with the reflections generated during the project that explained in detail the process of negotiating the assigned problem, preferably with practical examples.

Lastly, in P3, we asked students to prepare a final report and a slideshow on an assigned theme. This phase was executed over the course of 5 weeks (Weeks 9–13). During Week 9, each student was asked to prepare an individual final report describing in detail the entire project process and including their reflections on the assigned theme. Each student also had to complete a self-assessment form. During Week 10, each group produced its final report. Week 11 involved teams' refinement of their final reports based on the analysis and feedback of individual and group reports made by tutors. During Weeks 12 and 13, each group delivered a presentation of their slideshow.

4 Results and Discussion

In this section, we present reflections formed as a product of data analysis and the tutoring of students during the semester. Compared with the previous action research cycle [29], we found that more groups actively participated in the tasks throughout the semester. In fact, 9 of the 21 groups that started the project participated regularly and obtained feedback in communities of practice online (Table 3). Some factors have been reported to be relevant to the development of the project, including flexibility, task deadlines, and tutors' feedback.

Table 3. Assignment development by each group

Group	Assignment development					Summary
	N° members	Contacts with tutors		Contact on communitis		
		Email	Face to Face	N° topics	N° messages	
G1	4	2	1	40	107	Literature and reading notes; posts in several forums; brief contact with colleges and members of communities of practice.
G2	5	3	1	14	44	Literature and reading notes; posts in a few forums and on Facebook; questions lacked focus and clarity; no significant results were generated.
G3	5	-	-	28	55	List of literature and a few reading notes; posts in a few forums;

						questions lacked focus and clarity; no significant results were generated.
G4	5	2	1	17	24	List of literature and a few reading notes; posts in a few forums; questions lacked focus and clarity; no significant results were generated.
G5	5	-	-	3	12	Only worked during Week 1.
G6	3	1	1	19	92	List of literature and a few reading notes; brief contact with colleges and members of communities of practice; questions lacked focus and clarity, particularly during contact via Facebook; no significant results were generated; the solution was unoriginal.
G7	5	1	-	16	91	List of literature and a few reading notes; posts in a few forums and use of Google groups; brief contact with colleges and members of communities of practice.
G8	5	3	1	9	28	Only one student completed all activities; questions lacked focus and clarity; no significant results were generated.
G9	5	-	-	13	78	List of literature and a few reading notes; brief contact with colleges and members of communities of practice; questions lacked focus and clarity; no significant results were generated.
G10	5	-	-	2	19	Only worked by 4 weeks.
G11	5	2	-	12	53	Only worked by 6 week.
G12	5	2	1	6	14	Did not work during the final weeks.
G13	4	1	-	4	169	List of literature and a few reading notes; brief contact with colleges and members of communities of practice; questions lacked focus and clarity; no significant results were generated.
G14	5	1	-	14	54	List of literature and a few reading notes; posts in a few forums; questions lacked focus and clarity;

						no significant results were generated.
G15	5	-	-	0	0	Only worked during Week 1.
G16	5	1	1	2	18	Only one student completed the activities yet did not work during the final weeks.
G17	5	-	-	9	43	The most of the group did not work during the final weeks; two students worked individually; without any significant results; use of an unoriginal solution.
G18	5	1	1	9	0	Did not work during the final weeks.
G19	5	-	-	2	11	Only one student completed the activities yet did not work during the final weeks.
G20	2	-	-	1	6	Did not work during the final weeks.
G21	2	-	1	2	13	Only one student completed the activities yet did not work during the final weeks.

Students did not satisfactorily achieve expected learning outcomes. Of all the groups that completed the project, only six students obtained satisfactory grades. We noted that even students involved in the community of practice and who performed all activities did not feel motivated. In addition, all students interviewed said that they did not study regularly but rather made intensive, last-minute efforts to meet task deadlines and study for tests. Their lack of motivation and time were considered to be the primary reasons for their lack of dedication to studying and performing tasks.

This lack of motivation was also reflected in the collaboration of peers in each group. In only two groups did all members actively participate throughout the project; in another seven groups, not all students took part in all tasks, which significantly compromised the quality of the project. Still in other groups, only one or two students participated in the tasks, and most students gave up between the second and third phase of the project. From this, it is clear that rigor and the requirement of deadlines may not always be positive, meaning that the negotiation and flexibility of task deadlines can be decisive factors for the success of learning outcomes.

Concerning the participation of students in online communities of practice, “Portugal a Programar”¹ and “MSDN”² were the communities that experienced the most interaction. Though students declared that they did not know how to address the assigned themes with professional developers in these communities, when students received constructive feedback that helped them to clarify their doubts, their interaction with such professionals was considered to be a primary factor of the appropriateness of the task in the course. Another positive

¹ <http://www.portugal-a-programar.pt/>

² <https://msdn.microsoft.com/pt-pt/default.aspx>

indicator was the participation in these communities of PM3 alumni who collaborated in students' discussions of assigned topics. There were also situations in which students collaborated with group members in these communities.

Some students were reprimanded by communities' members for using the same introduction text in communities and for inserting too many topics for the same subject. Most students simply reused the introduction text model given by the teaching staff instead of customizing it. Yet, even though the topics of questions included by students addressed the same subjects, the staff of the communities guided students in correcting them, which shows that students did not clearly know the social protocols developed by these communities. We will doubtlessly keep these factors in mind for future activities. We conclude that it is necessary to provide better guidance for students' interactions in communities of practice. In this sense, in subsequent research iterations we intend to create a community of practice with students, alumni, and programmer experts to improve interaction and student motivation.

Though we asked students to become involved by discussing concepts and ideas, most students viewed the communities as a simple source of information. The factors that influenced this result occurred largely due to the students' inexperience with participation in communities of practice. Students tended to state that they often did not know how to discuss their questions with more experienced developers, partly due to their difficulty in the theoretical domain (e.g., understanding the problem and the development of its resolution). Most students sought an exact answer or a "magic solution" to the assigned problem, and there was generally poor involvement and application of their knowledge; for example, no group developed a basic coding for discussing its ideas in the communities.

Regarding tutoring, subjects ranged from problems with group composition and task delivery, as well as with the activity itself. Due to problems with group composition, some students started their tasks two or three weeks late and were thus affected by not having started the literature search early enough and not initiating contact with communities of practice. Meanwhile, problems with delayed task delivery affected students' reading notes, for one assessment criterion was to not assign grades to delayed work. The quality of work was another concern of tutoring; many students claimed to have difficulty developing strategies for solving their problems, even with feedback from face-to-face meetings and class discussions.

At the end of P1, some students who had difficulty with executing tasks could not report the possible causes of their difficulty. In response, we developed a series of dynamic groups in subsequent classes. To discuss barriers identified during the learning process, and attempt to generate more informal participation, we conducted a talk show with students during class. One of the tutors took the presenter-interviewer role and asked some students to be interviewed, while the other students formed the audience and were encouraged to contribute during the interviews. From our analysis of data obtained through the talk show, we identified that students did not know how to address the professional communities about their assigned problems. We conclude that this circumstance occurred because the students had little knowledge of the problem domain, as also reflected in their search of technical papers. Another problem identified was students' difficulty with understanding how theoretical knowledge covered in the course related to the practice of programming. In this sense, the dynamic also served to meet students' weekly study routines. Two factors that caught our attention were that students did not continually meet to work on tasks and devoted little time for group meetings; instead, they preferred to exert great effort on the eve of dead-lines.

For the subsequent class, we invited an alumnus of the course in order to motivate students by offering them a personal view about the assignments given by the guest. The presentation addressed aspects of their academic routine, their leisure activities (e.g., video games), the difficulties encountered during the course, and how to overcome them. Students also presented an illustrative schematic of how to address the assigned problems and define steps toward solving the problem. As a result, we expected that students would adapt these ideas to their realities.

The third and last group dynamic developed aimed to examine the difficulties with working in a group. During the dynamic called “complete the music,” students were asked to form groups, and different colors were assigned to each group. While the music was played, the lyrics of the parts were displayed in different colors, and each group had to stand up and sing the part written in its assigned color. The dynamic occurred in a gradual, interwoven process toward the end of the song, at which point all groups sang together. The students stated that this activity was fun and allowed reflection on the difficulties with working in groups. Activities like this have a long history in management and business [32,33].

At the end of the project, we considered adopting other tutoring strategies and feedback so that students could achieve the learning outcomes in terms of motivation and engagement. Tutoring students is a process that requires a heavy workload for one teacher and two tutors in which the role of team leaders becomes essential to assisting the teaching staff, both in forming teams and motivating students.

At the end of the action research cycle, students were asked to complete an online survey addressing their personal information, opinions of the adequacy of the tools, and difficulties with developing the project. However, only nine of the 95 students fully responded to the survey. Given this low response rate, which fails to take into account data collected, we list it among the limitations of the investigation.

5 Conclusions and future work

In this paper, we have reported the outcomes of the second cycle of action re-search. As in the previous cycle, students did not satisfactorily achieve expected learning outcomes. However, more groups completed the assigned activities. Results nevertheless reflect their poor engagement in the communities of practice and dismal performance with the other learning tasks. From these results, we have identified several ideas and recommendations that we intend to apply in subsequent versions of the coursework. For example, creating a community of practice with students, alumni, and invited programmer experts, rather than asking students to participate in external communities right away, might improve interaction and student motivation. Also, selecting better project-management practices in order to identify problems at an earlier stage may enable better guidance from the teaching staff to support students’ needs and hence help them achieve better results. Consequently, we propose that interaction and pedagogic assessment strategies are reshaped to simulate a business-like environment, including project management methods, e.g., SCRUM [34]. It should also include other aspects of a professional environment, such as teamwork, coaching, continual feedback and/or self-assessment strategies. We can consider this proposal to be a simulation of programming in a business context, and plan to refer to it in the future as the SimProgramming approach.

6 Acknowledgments

Nunes, R. R. & Pedrosa D. thank the Fundação para a Ciência e Tecnologia (FCT), Portugal, for Ph.D. Grants SFRH/BD/91309/2012 and SFRH/BD/87815/2012.

7 References

1. Tappan, D. (2014). A holistic multidisciplinary approach to teaching software engineering through air traffic control. *Journal of Computing Sciences in Colleges*, 30(1), pp. 199-205.
2. Zschaler, S.; Demuth, B. & Schmitz, L. (2014). Salespoint: A Java framework for teaching object-oriented software development. *Science of Computer Programming*, Volume 79, pp. 189-203.
3. Jenkins, T. (2002) On the difficulty of learning to program, Proceedings of 3rd annual conference of the LTSN-ICS, 27-29 August 2002, Loughborough (University of Ulster, LTSN Centre for Information and Computer Sciences).
4. Krasner, G. E., & Pope, S. T. (1988). A description of the model view controller paradigm in the Smalltalk-80 system. *Journal of Object-Oriented Programming*, 1(3), 26-49.
5. Curry, E., & Grace, P. (2008). Flexible self-management using the model-view-controller pattern. *Software, IEEE*, 25(3), 84-90.
6. Duderstadt, J. J. (2010). *Engineering for a changing world Holistic Engineering Education* (pp. 17-35): Springer.
7. Sheppard, S. D., Macatangay, K., Colby, A., & Sullivan, W. M. (2008). *Educating engineers: Designing for the future of the field* (Vol. 2): Jossey-Bass.
8. Adams, R., Evangelou, D., English, L., De Figueiredo, A. D., Mousoulides, N., Pawley, A. L., Schiefellite, C., Stevens, R., Svinicki, M., Trenor, J. M. and Wilson, D. M. (2011), Multiple Perspectives on Engaging Future Engineers. *Journal of Engineering Education*, 100: 48-88. doi: 10.1002/j.2168-9830.2011.tb00004.x
9. Litzinger, T., Lattuca, L. R., Hadgraft, R., & Newstetter, W. (2011). Engineering Education and the Development of Expertise. *Journal of Engineering Education*, 100(1), 123-150. doi: 10.1002/j.2168-9830.2011.tb00006.x
10. Stevens, R., Amos, D., Jocuns, A., & Garrison, L. (2007). Engineering as lifestyle and a meritocracy of difficulty: Two pervasive beliefs among engineering students and their possible effects. Paper presented at the American Society for Engineering Education Annual Conference, Honolulu, Hawaii.
11. Stevens, R., O'Connor, K., Garrison, L., Jocuns, A., & Amos, D. M. (2008). Becoming an engineer: Toward a three dimensional view of engineering learning. *Journal of Engineering Education*, 97(3), 355-368.
12. Johri, A., & Olds, B. M. (2011). Situated engineering learning: Bridging engineering education research and the learning sciences. *Journal of Engineering Education*, 100(1), 151-185.
13. Patterson, E. A., Campbell, P. B., Busch-Vishniac, I., & Guillaume, D. W. (2011). The effect of context on student engagement in engineering. *European Journal of Engineering Education*, 36(3), 211-224.
14. Sancho-Thomas, P., Fuentes-Fernández, R., & Fernández-Manjón, B. (2009). Learning teamwork skills in university programming courses. *Computers & Education*, 53(2), 517-531.
15. Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2011). Improving teaching and learning of computer programming through the use of the Second Life virtual world. *British Journal of Educational Technology*, 42(4), 624-637.
16. Schaffer, S. P., Chen, X., Zhu, X., & Oakes, W. C. (2012). Self-Efficacy for Cross-Disciplinary Learning in Project-Based Teams. *Journal of Engineering Education*, 101(1), 82-94.
17. Bédard, D., Lison, C., Dalle, D., Côté, D., & Boutin, N. (2012). Problem-based and project-based learning in engineering and medicine: Determinants of students' engagement and persistence. *Interdisciplinary Journal of Problem-based Learning*, 6(2), 8.

18. Yadav, A., Subedi, D., Lundeberg, M. A., & Bunting, C. F. (2011). Problem-based Learning: Influence on Students' Learning in an Electrical Engineering Course. *Journal of Engineering Education*, 100(2), 253-280.
19. Zhou, C., Kolmos, A., & Nielsen, J. F. D. (2012). A problem and project-based learning (PBL) approach to motivate group creativity in engineering education. *International Journal of Engineering Education*, 28(1), 3-16.
20. Rotgans, J. I., & Schmidt, H. G. (2012). Problem-based learning and student motivation: The role of interest in learning and achievement One-Day, One-Problem (pp. 85-101): Springer.
21. Pascual, R. (2010). Enhancing project-oriented learning by joining communities of practice and opening spaces for relatedness. *European Journal of Engineering Education*, 35(1), 3-16.
22. Morgado, Leonel; Varajão, João; Coelho, Dalila; Rodrigues, Clara; Sancín, Chiara; Castello, Valentina. 2010. "The Attributes and Advantages of Virtual Worlds for Real World Training", *The Journal of Virtual Worlds and Education* 1, 1: 15 - 35.
23. Stolk, J., & Harari, J. (2014). Student motivations as predictors of high-level cognitions in project-based classrooms. *Active Learning in Higher Education*, 15(3), 231-247.
24. Vanasupa, L., Stolk, J., & Harding, T. (2010). Application of self-determination and self-regulation theories to course design: Planting the seeds for adaptive expertise. *International Journal of Engineering Education*, 26(4), 914.
25. Slavich, G. M., & Zimbardo, P. G. (2012). Transformational teaching: Theoretical underpinnings, basic principles, and core methods. *Educational Psychology Review*, 24(4), 569-608.
26. Pritchard, A. & Woollard, J. (2010). *A Psychology for the classroom: constructivism and social learning*. Oxford, UK: Routledge/David Fulton Education, 112pp.
27. Universidade de Trás-os-Montes e Alto Douro, "Despacho nº 14253/2011 - Regulamento do curso de Licenciatura em Engenharia Informática", *Diário da República*, 2ª série, nº. 202, October 20th, 2011.
28. Joint Task Force on Computing Curricula. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. New York, NY, USA: ACM.
29. Morgado, L.; Fonseca, B.; Martins, P.; Paredes, H.; Cruz, G.; Maia, A. M.; Nunes, R. R.; Santos, A. (2012). Social networks, microblogging, virtual worlds, and Web 2.0 in the teaching of programming techniques for software engineering: a trial combining collaboration and social interaction beyond college. In *IEEE Engineering Education 2012 - Collaborative Learning & New Pedagogic Approaches in Engineering Education - Proceedings*, Marrakesh. doi: 10.1109/EDUCON.2012.6201129.
30. Brandel, M. (2013). 8 hot IT skills for 2014. [Online] Available at: <http://www.computerworld.com/article/2484858/it-management/8-hot-it-skills-for-2014.html> [accessed on 29 September 2014].
31. Barbosa, L. F., Alves, P., & Barroso, J. (2011, June). SIDE - Teaching support information system. In *Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on* (pp. 1-6). IEEE.
32. Nilson, Carolyn (1993). *Team games for trainers*. New York: McGraw-Hill.
33. Pike, Robert W., and Solem, Lynn (2000). *50 creative training openers and energizers: innovative ways to start your training with a bang*. San Francisco: Jossey-Bass/Pfeiffer & Creative Training Techniques Press.
34. Schwaber, Ken (2004). *Agile project management with Scrum*. Microsoft Press.