






Co-regulated Learning in Computer Programming: Students Co-reflection About Learning Strategies Adopted During an Assignment

Daniela Pedrosa¹ , José Cravino^{1,2} , Leonel Morgado^{3,4} ,
and Carlos Barreira⁵

¹ School of Science and Technology,
Universidade de Trás-os-Montes e Alto Douro (UTAD), Vila Real, Portugal
{dpedrosa, jcravino}@utad.pt

² Research Centre “Didactics and Technology in Education of Trainers”,
Aveiro, Portugal

³ LE@D & CIAC, Universidade Aberta, Coimbra, Portugal
leonel.morgado@uab.pt

⁴ INESC TEC, Porto, Portugal

⁵ Faculdade de Psicologia e Ciências da Educação,
Universidade de Coimbra, Coimbra, Portugal
cabarreira@fpce.uc.pt

Abstract. Higher education students exhibit difficulties in learning computer programming, particularly transitioning from initial programming to advanced programming, so it’s necessary to develop effective teaching strategies.

We developed the SimProgramming approach to help students overcome learning difficulties transitioning from entry-level to advanced computer programming, by developing appropriate learning strategies. The students perform a specific set of tasks in a learning environment that simulates business operations, developing a problem-based learning assignment. One of those tasks is filling biweekly individual self-reflection and co-reflection forms.

This approach was implemented at the University of Trás-os-Montes e Alto Douro (Portugal), in a 4th semester course of two bachelor programmes: Informatics Engineering and ICT. The students provided 37 biweekly forms, on which we conducted thematic analysis to identify their strategies for co-regulation of learning during the assignment.

Students are adopting different strategies in each phase of the approach. Early phases are devoted to organization, planning, and transformation of information, and later phases focus on applying theoretical knowledge and hands-on programming.

We recommend including this type of pedagogical task (biweekly self-reflection and co-reflection forms) in educational practices, in view of their contribution to improving self- and co-regulation learning strategies.

Keywords: Co-regulation learning · Computer programming · SimProgramming approach · Reflection

1 Introduction

There is a growing need for skilled professionals in computer science [1]. However, Higher Education students exhibit difficulties learning computer programming, particularly transitioning from initial programming to advanced programming [2], with high rates of academic failure being common. Students' difficulty in learning to program is particularly noticeable in the transition from entry-level programming to advanced programming [2].

The reasons include the teaching approach and the attitudes/strategies used by students in computer programming [3], including the lack of motivation and involvement in study [2, 3]. Also, most students come to the job market lacking the necessary skills to meet the expectations of employers [4], such as teamwork and cooperation skills [5].

In advanced programming courses, the level of complexity is much greater than entry-level programming courses. Differences impacting students include large code sizes, the need to work in teams of varying size with associated communication issues, and a dynamic environment where regular changes to existing code become necessary. When applying techniques to address this complexity, such as architectural styles, e.g. Model–View–Controller (MVC) [6], students have difficulties grasping the rationale of the technique and other software engineering concepts [7]. There are also other complex programming skills [8] and social skills [5].

We seek to employ in this regard the concept of Self-Regulated Learning (SRL), a key element for success in higher education, which allows students to be proactive managing their learning and developing life skills [9]. The application of SRL strategies typically predicts high academic achievement [10] and can be improved with appropriate interventions [11]. Also, it is recommended that teachers promote and contribute towards students' development of metacognitive knowledge about academic work and task-specific strategies [12].

In computer science, students that apply SRL and metacognitive strategies have good performance [13]. However, most students are not aware of SRL and metacognitive strategies that can be used, so instilling them is important [14].

Co-Regulated Learning (CRL) also helps understand the process of regulating student learning [15]. In the context of computer programming, CRL helps students improve their programming skills [16], by providing a set of resources and skills for working with others [17].

We developed the SimProgramming approach [18–21] and applied it in the academic year 2013/2014 to the Programming Methods 4 (PM4) course, part of the third year of the bachelor programmes in Informatics Engineering (IE) and in Information & Communication Technologies (ICT) at the University of Trás-os-Montes e Alto Douro (UTAD), Portugal.

In the SimProgramming approach, the course syllabus requires students to develop a problem-based learning (PBL) assignment with a specific set of tasks based on the conceptual foundations of SimProgramming and is described further ahead in Sect. 3.

One of these tasks is filling out biweekly forms for self-reflection and co-reflection. These are handed out to each student's team and focus on their biweekly performance

developing the course assignment. We conducted thematic analysis of 37 biweekly forms seeking to identify the SRL and CRL strategies mentioned by students during the development of the assignment.

In previous studies, we described the SRL and CRL strategies used by individual students, collected via weekly forms [19] and semi-structured interviews [20, 21]. In this paper, we look at the team perspectives on co-regulation learning strategies, collected via biweekly forms.

2 Background

The cyclical model of SRL is based on the social cognitive theory proposed by Zimmerman, which has three phases for self-regulation: (1) forethought, which is the goal setting and planning before the assignment/study; (2) performance, which is when the students use various strategies, monitoring and controlling their learning; and (3) self-reflection, reflecting about the learning process after assignment/study [22–24].

In fact, SRL is considered a meta-process [23] in which students are proactive and have control over their cognition, behavior, emotions and motivation through the use of personal strategies to achieve their established personal goals [25].

Self-regulated learners are active participants in their learning and develop academic skills [26]. They are generally successful and have good academic performance [10], adopting various learning strategies [23] during an academic assignment, such as cognitive, behavioral and motivational strategies to improve and guide the learning process [27].

However, university students may find it difficult to regulate their learning [28], which in turn may lead them to abandon courses [29], or not finish them on time [24].

SRL provides students with opportunities for getting acquainted with effective practices/strategies for their study, such as: time management; resource management; environmental management; incorporating feedback; and management of learning objectives and results [26, 30]. Students construct their own meanings, goals, and strategies from the information available in the external environment and in their own minds [31].

SRL strategies (SRLS) are specific skills that are part of the SRL process and can be taught to students for them to apply in real contexts [22, 23], such as: goal setting and planning, organizing and transforming, seeking information, rehearsing and memorizing, environmental structuring, seeking social assistance, self-consequences, records and monitoring, reviewing records, and self-evaluation [23].

SRLS help students obtain and retain knowledge about the adoption of a methodological approach and structure their learning, affecting their results [10]. According to Wang et al. [32], the application SRLS is usually a predictor of good academic performance.

CRL is understood as a social regulation of learning, in which students temporarily regulate their cognition, behavior, motivation, and emotions in situations of temporary coordination of regulation with other people (teachers or peers) [24, 27, 33]. This interaction of the student with others allows him/her to internalize regulatory processes (ibid.).

Team regulation, social activities and learning co-regulation are important for successful collaboration, because team members contribute to building shared dynamic mental models, and awareness towards a successful decision-making process [34]. In addition, cognition and feelings can be encouraged by peers or teachers to improve and increase understanding of tasks [35]. CRL helps understand the processes of student regulation [15], when students co-regulate learning, they have the potential to improve their learning self-regulation skills [36].

In computer programming, CRL helps students improve their programming skills [16], as it provides students with a set of outwardly resources and skills (seeking social help, evaluating others' ideas, monitoring tasks) [17].

Reflective learning helps students become more aware of the learning process and their difficulties [37]. When students make effective self-reflection, they analyze how they learned, how they understood the objectives of the learning process and what is necessary to create conditions for success [38]. It also encourages students' critical thinking about their abilities, and reflects on improvement strategies for the learning process, making them aware of the learning advantages in the future, and helps them develop transversal skills [37]. The interaction between compromise, self-control, autonomy and students' self-discipline allows them to regulate their own actions to achieve their learning goals [39].

On the other hand, reflective learning provides feedback to teachers, enabling them to readjust their experiences and pedagogical tools [38]. The use of a reflective diary is a technique that reinforces and stimulates reflection on the theoretical and practical component of work (ibid.).

The pedagogical context contributes towards learners' engagement and resolve to achieve learning outcomes [40]. In higher education, it is important to prepare students for the challenges of later professional practice and provide students with opportunities to develop their self-regulation and co-regulation skills, through activities that improve collaborative and active learning [32].

In engineering education in particular, learning approaches are typically not aligned with the requirements of the labor market [41, 42], not prioritizing skills aligned with professional realities, such as active learning or integrating knowledge [43].

Role-playing pedagogical techniques stimulate students to learn about similar real-world situations, with problem-solving, and active learning, providing opportunities to learn by doing and collecting feedback for building new knowledge [44]. They also help develop professional identities [45].

3 The SimProgramming Approach

The SimProgramming approach [18–21] is based on four conceptual foundations (ibid.): (1) business-like learning environment, (2) SRL; (3) CRL, and (4) formative assessment.

The SimProgramming approach was developed through iterative application and refinement, between 2011 and 2014.

Through these conceptual foundations, teaching strategies are adopted to stimulate SRLS and CRLS. The learning activity process develops along four phases, and

students have specific tasks in each phase, with specific environment, roles, and deadlines during a course-long (i.e., semester-long) assignment [18–21].

Conceptual Foundation 1: Business-Like Learning Environment

SimProgramming stipulates the simulation of a business-like environment with the goal of promoting awareness of the professional reality and teamwork expectations, through role-playing, with each participant taking on a role. Also, Problem-Based Learning (PBL) is used to promote collaborative discovery for the resolution of the problem [46].

The course lecturer plays the role of CEO or general manager, taking responsibility for the course content and monitoring.

Other teaching staff of the course (e.g., course tutors or teaching assistants – henceforth “tutors”) play the role of project managers, providing close monitoring, mentoring, and feedback to students, based on the Scrum method for project management and agile software development [47].

Students play different roles as members of development teams and divide the work according to the role played by each member. One student acts as team leader and the remaining students handle subsets of work (“work packages”). The team leader facilitates the integration of information and guides the group [5], making sure that team members keep a global view of the project context and status, integrating knowledge. Other students have a specific role each, having to master their individual packages and cooperate with the team leader.

Conceptual Foundations 2: Self-regulated Learning (SRL)

The goal of conceptual foundation 2 is to promote students’ SRLS through active participation and engagement in meaningful activities before, during, and after completion of academic work [39].

The SimProgramming approach aims to promote students’ SRL strategies through active learning and engagement in meaningful activities before, during, and after completion of academic work [32]. Each student must solve his/her individual work packages and contribute to the overall perspective of the team problem. The team leader integrates research and exploration output of all members, reporting weekly at project management meetings. He/she also ensures the information flow within the team. Biweekly, each team completes a self-reflection form about his/her own work, pondering on what to do the following week, and reflecting upon the factors that prevented him/her from achieving the team and individual objectives. Students are encouraged to develop the concept of doing their work regularly and adopting study routines. This is done by creating a context where tasks are performed continuously, alongside feedback and monitoring support for self-reflection and self-regulation.

Conceptual Foundation 3: Co-regulated Learning (CRL)

The SimProgramming approach encourages *co-regulated learning* (*conceptual foundation 3*) with the aim of supporting the functional and effective development of a learning community of practices around problem solving. Students have two kinds of team tasks: reports and presentations.

In SimProgramming, students should be encouraged to get involved in pre-existing online communities of professionals (outside academia) not just to seek help, but to help other members of those communities, contributing to problem-solving and discussing the technologies under study or used in their future profession [2]. The search for help among professional communities is a common practice in real-world labour.

Tutors (in meetings, classes, and on-line) stimulate students' initiative when seeking social help (from peers, teachers, tutors, etc.), advising against using it as a mere least-effort approach to clarify doubts and difficulties. The tutors and the professor provide this support by advising on methods of gradual participation and involvement in communities, including suggestions of specific tasks for clarification of concepts, and advise on development of homogenous peer-based contributions and discussion, supporting community development, informal interactions, and debate, which can be promoted and monitored via a Facebook group for the course or other forms of groupware.

Conceptual Foundation 4: Formative Assessment

In real world labour it is well-known that companies conduct assessments of team performance. So, SimProgramming also employs Formative Assessment with the goal to improve self-reflection by providing management feedback (from tutors and from the Professor).

The Professor and tutors employ face-to-face and online contact to provide monitoring, meetings, and social media interactions, including motivational mentoring, coaching, and feedback on individual package status. The feedback is adopted as support for students' self-regulation and critical thinking. Assessment in SimProgramming takes two forms: formal self-assessment of individual students and hetero-assessment by team members at the end.

SimProgramming Phases: Learning Assignment Process

In the SimProgramming approach [18–21], the learning assignment is developed along four phases and students have specific tasks in each phase (Fig. 1), based on the SimProgramming conceptual foundations presented above. During all phases, weekly meetings take place between tutors and team leaders, providing feedback for motivation, self-regulation, possible support for technical doubts, and internal team issues.

What Are the Team Biweekly Forms?

The team biweekly form is where each team self- and co-reflects upon their work (teamwork and individual work), ponder on what to do the following weeks, and reflect upon the factors that prevented him/her or the team from achieving specific goals [18]. It acts as a reflexive diary with the goal of stimulating reflection about work performance.

Each team must answer 3 questions: (1) “What have you done these two weeks for the assignment?”; (2) “What will you do in the coming weeks for the assignment?”; and (3) “Any reason(s) for not completing the tasks?”.

	<i>Phase 1</i>	<i>Phase 2</i>	<i>Phase 3</i>	<i>Phase 4</i>
Assignment Goals	Searching for information about the technologies under study;	Integration of technologies;	Improving the assignment;	Final improvement of the assignment.
	Interacting in online communities;			
	Group work: Initiate problem-solving.	Group work hands-on examples	Final presentation with problem-solving	
Specifics Tasks	Report interaction in community of practice (team)			Final Report (team)
	Report interaction in community of practice (team)			
	Weekly forms (individual) – all phases			
	Weekly meetings between team leaders and tutors - all phases			
	Report about learning progress (team) - all phases			Grids about self-assessment of individual students and hetero-assessment by team members (of individuals)
	Presentation of the team work – all phases			Extra task for extra credit or replacement credit (individuals or team)
Duration of the Phases	3 weeks	3 weeks	2 weeks	2 weeks

Fig. 1. SimProgramming phases: goals, specifics tasks and duration.

4 Teaching Context and Learning Assignment

4.1 Teaching Context

The course goal in PM4 is for students to develop the knowledge and skills necessary to develop web applications. Students start by working with the client-server concept of web applications and study their operation, including analysis of the HTTP protocol and the processing of its messages by web clients and servers. The syllabus for PM4 includes data formats and metadata for web applications, including the meta-languages SGML and XML, and languages specified by them. It then proceeds with the internal operation of Web clients and servers, including automation, and concludes with the study of various types of Web applications and the specific case of Web services.

In the previous year, students had experienced an early version of the SimProgramming approach in the PM3 (Programming Methods 3) course, which was focused on software architectures, and code complexity management techniques, such as concern independence, transparency, etc.

4.2 Learning Assignment in PM4

We combined proposals described in the literature about techniques of technology-enhanced learning [48] and face to face teaching techniques to support students during the assignment [19]. The tutors scheduled face-to-face meetings with team leaders, either individually or as a team, when they identified problems or difficulty fulfilling the tasks.

The Moodle LMS is used as the on-line environment for the professor and the tutors to track the development of the assignment and organize the tasks into modules over several weeks. Through the LMS, students are provided with supporting materials for development of tasks, scheduling, overall objectives of the assignment and specific objectives of each task, a forum for doubts and for contacting tutors, and other course materials (e.g. slideshows used in lectures). Also, other on-line tools are used to support students: e-mail, instant messaging (GTalk now called Google Hangouts), Facebook, and SIDE, which is a locally-developed course management system [49] for students to submit their completed tasks.

The learning assignment is based on PBL [46]. In PM4, a specific problem is assigned to each team, involving protocols, web applications, and markup languages. The goal of the problem is to lead students to develop skills on the development of web applications. Students must develop a technological solution using a web system and a given web access platform (e.g. mobile devices) and explain in detail (including code examples) how it is possible to exchange information between systems using different markup languages.

The SimProgramming approach was used throughout, along all the 4 phases, during 10 weeks of the academic semester, described ahead. In the 2013/2014 academic year, students formed 9 teams (Table 1). Most of the students participated in the previous course (PM3) and thus had prior contact with this approach.

All teams successfully achieved the learning goals. Of the 49 students, 44 attained a final grade.

Table 1. Nr. of the students in assignment

Teams	Nr. students with a final grade	Comments
P	6/6	- Were regular in delivery of individual tasks and teamwork - In the final weeks, the team presented some interpersonal problems among team members
Q	5/6	- Were regular in delivery of individual tasks and teamwork - Same team as in PM3, 2012/2013 (team H, Pedrosa et al., 2016) - One student quitted the assignment
R	7/7	- Same team as in PM3, 2012/2013 (team B, Pedrosa et al., 2016) - Were regular in delivery of individual tasks and teamwork. However, they did not submit two of the biweekly team forms

(continued)

Table 1. (continued)

Teams	Nr. students with a final grade	Comments
S	4/4	- Were regular in delivery of individual tasks and teamwork. However, they did not submit two of the biweekly team forms - Intervention of the tutor through meetings providing feedback and motivation
T	6/6	- Were regular in delivery of individual tasks and teamwork
U	6/6	- Were regular in delivery of individual tasks and teamwork
V	4/4	- Were regular in delivery of individual tasks and teamwork. However, they did not submit one of the biweekly team forms
X	3/5	- Were regular in delivery of individual tasks and teamwork. However, they did not submit four of the biweekly team forms - Intervention of the tutor through meetings providing feedback and motivation - Two students quitted the assignment, one them being the initial team leader
Y	3/5	- Were regular in delivery of individual tasks and teamwork - In the first weeks, the team presented some interpersonal problems among the members of the team - Two students quitted the assignment, one them being the initial team leader - Intervention of the tutor through meetings providing feedback and motivation - Initially the team was not achieving the objectives of the activity, however after the change of the team leader, the team showed good results
Total	44/49 (89, 8%)	- 44 of 49 students (90%) who registered for the activity obtained a final classification

5 Methodology and Data Collection

During the 10 weeks in which the assignment took place, each team had to submit its biweekly form as a team, i.e. 5 forms in total. The fulfilment of this varied slightly throughout the phases of the SimProgramming approach (Table 2).

Table 2. Delivery of biweekly team forms by teams, per SimProgramming phases

SimProgramming phases	Weeks	Nr. biweekly forms delivered	Total
Phase 1	Week 1	9	9
Phase 2	Week 3	8	15
	Week 5	7	
Phase 3	Week 7	6	6
Phase 4	Week 9	7	7
		Total	37

All the 9 teams initially enrolled in the assignment completed all phases of the SimProgramming approach, performing the requested tasks.

In the first week, all teams delivered the biweekly forms. In Phase 2, Phase 3 and Phase 4, occasionally, some teams did not submit the biweekly team forms (Table 2).

The 37 biweekly team forms were subjected to thematic analysis [50] aiming to identify the strategies of self and co-regulation of the learning (SCRLS) that students mentioned during the assignment.

Content analysis matrices based on the afore mentioned research literature were constructed on the types of self and co-regulation learning strategies (SCRLS), identification of difficulties in the assignment and on the factors that the students believed that influenced their motivation (Table 3).

Table 3. The three sub-categories about SCRL strategies identified in biweekly forms

Sub-categories	Definition
1. Organizing, planning and transforming strategies	Statements about the strategies that students adopted in the planning, organization, and processing of information for carrying out the assignment
2. Identifying the difficulties in the assignment	Statements about the difficulties students felt in doing the assignment
3. Co-reflexion strategies	Statements about the work performance

We organized content into categories, subcategories, indicators, and recording units (snippet sentences), which were restated during the process of content analysis. Then, we conducted a cyclical process of improvement, synthesis, and reflection.

The steps adopted for the data analyses were as follows:

1. Construction of content analysis matrices for each team, with the SCRLS (phrases/snippet sentences that students reported on the biweekly forms, explaining what they did). The content analysis matrices are composed of grid lines (each line for a strategy –the “indicators”); and columns to record in which week it was reported by team students. In the cells we entered codes identifying the team reporting that strategy that week (e.g. E.3).
2. Afterwards, we developed general syntheses of each team for each of the indicators.
3. For each subcategory of the strategies (e.g. Organizing and planning strategies) we counted the number of teams who reported each indicator (e.g. 1.1 = 113).
4. Finally, we did a general synthesis of the indicators in each of the phases.

In a first moment, one researcher (one of the authors of this paper) created the content analysis matrixes (with the categories, the subcategories, indicators and the recording units). After this, another researcher independently validated or suggested changes to these content analysis matrixes. When changes were proposed, they were discussed later until both researchers reached an agreement. Finally, the final version of the content analysis matrixes was validated by all the researchers.

6 Results and Discussion

In the biweekly team forms, we found that the teams mentioned “Organizing (O), Planning (P) and Transforming (T) strategies” to carry out the assignment (Table 4).

In the first phases (Phase 1 and Phase 2), the most mentioned strategies by teams are team meetings to define tasks (n = 24), information research (n = 15), application of knowledge related to practice (n = 14), elaboration of notes on the information found (n = 14) and also the division of tasks among the team members (n = 13). Other strategies were also mentioned by the teams.

In the Phase 3 and Phase 4, the strategies that were most mentioned by the teams are related to the transformation of the information found through notes (n = 8), and the application of the information in practical context (n = 11). Likewise, the teams had the concern to understand the research material (n = 8).

Table 4. Organizing (O), Planning (P) and Transforming (T) strategies

Indicators	SimProgramming phases			
	Phase 1 (n = 9)	Phase 2 (n = 15)	Phase 3 (n = 6)	Phase 4 (n = 7)
Information search (O)	8	7	1	0
Collected information (O)	5	9	2	3
Work plan development (P)	1	2	1	1
Understand the project goal	1	3	0	0
Recording practices online communities (O)	1	0	0	0
Meeting schedule with team colleague (O)	4	6	2	3
Meeting with tutor (O)	1	0	0	0
Division of tasks between team members (O)	5	8	2	1
Drafting notes about collected information (T)	6	8	3	5
Application of existing knowledge about the practice (T)	5	9	5	6
Understanding (learning) through the collected information search (T)	4	4	2	6
Team meeting to define task (P)	4	4	0	0
Defining specific tasks for next week (P)	9	15	5	1

Likewise, the teams identified and expressed in the biweekly forms the difficulties they experienced carrying out the assignment, as can be seen in Table 5. The most mentioned difficulty in all phases (Phase 1, Phase 2, Phase 3, and Phase 4) was the overload of work.

In the early stages (Phase 1 and Phase 2) the teams also reported difficulty meeting (n = 11). Subsequently, in the intermediate phases (end of Phase 2, and Phase 3) the teams also expressed difficulties implementing the practical component.

Table 5. Identifying difficulties in the assignment

Indicators	SimProgramming phase			
	Phase 1 (n = 9)	Phase 2 (n = 15)	Phase 3 (n = 6)	Phase 4 (n = 7)
Team meeting	3	8	1	0
Time management (TM)	2	0	0	1
Theoretical knowledge about the technology being studied	1	0	0	0
Team member quitting and/or inter-team relationships	0	1	0	1
The practical component implementation	0	3	1	1
TM due to work in other courses or tests	1	11	2	6

Regarding the co-reflection of team members on their tasks, (Table 6), the teams mentioned whether or not they had achieved their goals, and difficulties the experienced achieving the objectives of the task. In the intermediate stages (Phase 2 and Phase 3), the teams mentioned more emphatically that they were able to reach the goals but had difficulties. Throughout all phases, most teams reflected specifically on the tasks performed.

Table 6. Co-reflection

Indicators	SimProgramming phases			
	Phase 1 (n = 9)	Phase 2 (n = 15)	Phase 3 (n = 6)	Phase 4 (n = 7)
Co-reflection (CR) - Achieved the biweek goals	3	3	2	6
CR - Achieved the goals with difficulties	3	10	4	0
CR - Failed to meet goals due to difficulty holding a team meeting	1	1	0	0
CR - Failed to meet objectives due to overload of work with other courses	0	1	0	0
CR - Reflection on specific task	3	6	4	6

Throughout the phases of the SimProgramming approach, the teams mentioned, in the biweekly forms, the adoption of several different strategies in each of the phases.

In the first phases (Phase 1 and Phase 2), the organization and planning strategies were the most mentioned ones. In all phases, the strategies that stand out are related to the application and transformation of information, writing notes on the collected research material, the application of the practical component (programming), and the respective understanding (learning) of the research material. These strategies are skills necessary for the development of teamwork and in the context of the world of work. Students have improved their skills during the SimProgramming approach phases and are better prepared for transitioning to the professional practice world.

The teams identified difficulties in time management, namely, difficulties due to the excess of tasks and tests in the various courses throughout the semester. In the initial phase, the teams mentioned difficulties holding team meetings. Subsequently, the difficulties are related to the implementation of the practical component.

Likewise, team members were involved in a process of reflection on their learning, explaining whether the team had achieved the goals of the SimProgramming approach or not. They also identified occasional difficulties in achieving the objectives, especially in Phase 2 and Phase 3. Throughout all phases, it was verified that as a team, the students reflected in more detail on their performance.

7 Conclusions

Throughout the discussion, we argued that using biweekly team forms in PM4 (2013/2014) allowed us to verify that the adoption of this type of pedagogical task contributes to the improvement of strategies of self and co-regulation of learning, since it allowed students to be aware and reflect on essential competences to reach the learning goals, an important skill for the world of professional practice, confirming what was reported in an earlier work [19].

The students mentioned several types of learning strategies that they adopted during the assignment, namely: organization and planning strategies, identification of difficulties, and co-reflection on the assignment.

Problem solving strategies were not identified in the biweekly team forms, and neither were factors affecting motivation. In these forms, the teams focused on the detailed explanation of the work in terms of content, reinforcing the perspective of our previous experience in PM3 (2012/2013) [19]: it contributed to the development of competencies [19].

However, it was observed that in comparison to the individual weekly forms of the previous course, PM3 (2012/2013) [19], in the biweekly team forms of PM4 (2013/2014) the students (as a team) mentioned in more detail the strategies of information transformation. That is, in a team context, the students explained in greater detail the process of treatment of the collected material, and how they applied the knowledge obtained in the activity, in which the team's concern in understanding (learning) that material.

The difficulties found in the assignment are identical. Namely, the overload of work and the implementation of the practical component. However, in the team reflections, it is also mentioned the difficulty of getting the team to meet.

The students as teams can reflect on whether they have met (or not) the goals and identified their difficulties. Likewise, as a team they make a more detailed reflection on the tasks, especially when accompanied by the manager. This confirms earlier studies that found that tutors play an important role in students' self-assessment [30].

Acknowledgment. Pedrosa, D. wishes to thank the Fundação para a Ciência e Tecnologia (FCT), Portugal, for Ph.D. Grant SFRH/BD/87815/2012.

This work is financially supported by National Funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., under the project UID/CED/00194/2013.

We would like to thank all the students and teachers who collaborated on this research.

References

1. Balanskat and Engelhart. Computing our future. Computer programming and coding – priorities, school curricula and initiatives across Europe. European Schoolnet (2015)
2. Morgado, L., et al.: Social networks, microblogging, virtual worlds, and Web 2.0 in the teaching of programing techniques for software engineering: a trial combining collaboration and social interaction beyond college. In: Global Engineering Education Conference (EDUCON), pp. 1–7. IEEE (2012)
3. Gomes and Mendes. À procura de um contexto para apoiar a aprendizagem inicial de programação. *Educação, Formação & Tecnologias* **8**(1), 13–27 (2015)
4. Kumar, B.: Gamification in education-learn computer programming with fun. *Int. J. Comput. Distrib. Syst.* **2**(1), 46–53 (2012)
5. Sancho, P., Moreno-Ger, P., Fuentes-Fernández, R., Fernández-Manjón, B.: Adaptive role playing games: an immersive approach for problem based learning. *Educ. Technol. Soc.* **12**(4), 110–124 (2009)
6. Curry, E., Grace, P.: Flexible self-management using the model-view-controller pattern. *IEEE Softw.* **25**(3), 84–90 (2008)
7. Cagiltay, N.E.: Teaching software engineering by means of computer-game development: challenges and opportunities. *BJET* **38**(3), 405–415 (2007)
8. Jenkins, T.: On the difficulty of learning to program. In: Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, vol. 4, pp. 53–58 (2002)
9. Fernández, E., Bernardo, A., Suárez, N., Cerezo, R., Núñez, J.C., Rosário, P.: Predicción del uso de estrategias de autorregulación en educación superior. *Anales de psicología* **29**(3), 865–875 (2013)
10. Broadbent, J., Poon, W.L.: Self-regulated learning strategies & academic achievement in online higher education learning environments: a systematic review. *Internet High. Educ.* **27**, 1–13 (2015)
11. Zimmerman, B.J.: Investigating self-regulation and motivation: historical background, methodological developments, and future prospects. *Am. Educ. Res. J.* **45**(1), 166–183 (2008)
12. Cazan, A.M.: Teaching self regulated learning strategies for psychology students. *Procedia-Soc. Behav. Sci.* **78**, 743–747 (2013)
13. Bergin, S., Ronan, R., Desmond, T.: Examining the role of self-regulated learning on introductory programming performance. In: Proceedings of the First International Workshop on Computing Education Research. ACM (2005)
14. Alharbi, A., Paul, D., Henskens, F., Hannaford, M.: An investigation into the learning styles and self-regulated learning strategies for computer science students. In: Proceedings Ascilite (2011)
15. Harley, J., Taub, M., Bouchet, F., Azevedo, R.: A framework to understand the nature of co-regulated learning in human-pedagogical agent interactions. In: 11th International Conference on Intelligent Tutoring Systems, Crete (2012)
16. Tsai, C.W.: Applying web-based co-regulated learning to develop students’ learning and involvement in a blended computing course. *Interact. Learn. Environ.* **23**(3), 344–355 (2015)
17. Hwang, G.J., Wu, P.H., Chen, C.C.: An online game approach for improving students’ learning performance in web-based problem-solving activities. *Comput. Educ.* **59**(4), 1246–1256 (2012)

18. Pedrosa, D., et al.: Simprogramming: the development of an integrated teaching approach for computer programming in higher education. In: Proceedings 10th Annual International Technology, Education and Development Conference (INTED 2016), Valencia, Spain (2016, to appear)
19. Pedrosa, D., Cravino, J., Morgado, L., Barreira, C.: Self-regulated learning in computer programming: strategies students adopted during an assignment. In: Allison, C., Morgado, L., Pirker, J., Beck, D., Richter, J., Gütl, C. (eds.) iLRN 2016. CCIS, vol. 621, pp. 87–101. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41769-1_7
20. Pedrosa, D., Cravino, J., Morgado, L., Barreira, C.: Self-regulated learning in higher education: strategies adopted by computer programming students. In: Proceedings 8th International Symposium on Project Approaches in Engineering Education (PAEE), Guimarães, Portugal (2016)
21. Pedrosa, D., Cravino, J., Morgado, L., Barreira, C.: Self-regulated learning in higher education: strategies adopted by computer programming students when supported by the SimProgramming approach. *Production* **27**(spe), e20162255 (2017)
22. Zimmerman, B.J., Schunk, D.H.: Self-regulated learning and performance. An introduction and an overview. In: Zimmerman, B.J., Schunk, D.H. (eds.) *Handbook of Self-regulation of Learning and Performance*, pp. 1–12. Routledge, New York (2011)
23. Zimmerman, B.J.: From cognitive modeling to self-regulation: a social cognitive career path. *Educ. Psychol.* **48**(3), 135–147 (2013)
24. Räisänen, M., Postareff, L., Lindblom-Ylänne, S.: University students' self- and co-regulation of learning and processes of understanding: a person-oriented approach. *Learn. Individ. Differ.* (2016). <http://dx.doi.org/10.1016/j.lindif.2016.01.006>
25. Panadero, E., Alonso-Tapia, J.: How do students self-regulate? Review of Zimmerman's cyclical model of self-regulated learning. *Anales de Psicología/Ann. Psychol.* **30**(2), 450–462 (2014)
26. Clark, I.: Formative assessment: assessment is for self-regulated learning. *Educ. Psychol. Rev.* **24**(2), 205–249 (2012)
27. Panadero, E., Järvelä, S.: Socially shared regulation of learning: a review. *Eur. Psychol.* (2015). <https://doi.org/10.1027/1016-9040/a000226>
28. Heikkilä, A., Lonka, K., Nieminen, J., Niemivirta, M.: Relations between teacher students' approaches to learning, cognitive and attributional strategies, well-being, and study success. *High. Educ.* **64**, 455–471 (2012)
29. van den Bogaard, M.: Explaining student success in engineering education at Delft University of Technology: a literature synthesis. *Eur. J. Eng. Educ.* **37**(1), 59–82 (2012)
30. Nicol, D.J., Macfarlane-Dick, D.: Formative assessment and self-regulated learning: a model and seven principles of good feedback practice. *Stud. High. Educ.* **31**(2), 199–218 (2006)
31. Pintrich, P.R.: A conceptual framework for assessing motivation and self-regulated learning in college students. *Educ. Psychol. Rev.* **16**(4), 385–407 (2004)
32. Wang, C.H., Shannon, D., Ross, M.: Students' characteristics, self-regulated learning, technology self-efficacy, and course outcomes in online learning. *Distance Educ.* **34**(3), 302–323 (2013)
33. Järvelä, S., Järvenoja, H.: Socially constructed self-regulated learning and motivation regulation in collaborative learning groups. *Teach. Coll. Rec.* **113**(2), 350–374 (2011)
34. Saab, N.: Team regulation, regulation of social activities or co-regulation: different labels for effective regulation of learning in CSCL. *Metacogn. Learn.* **7**(1), 1–6 (2012). <https://doi.org/10.1007/s11409-011-9085-5>
35. Zheng, L., Huang, R.: The effects of sentiments and co-regulation on group performance in computer supported collaborative learning. *Internet High. Educ.* **28**, 59–67 (2016). <https://doi.org/10.1016/j.iheduc.2015.10.001>

36. DiDonato, N.C.: Effective self-and co-regulation in collaborative learning groups: an analysis of how students regulate problem solving of authentic interdisciplinary tasks. *Instr. Sci.* **41**(1), 25–47 (2013). <https://doi.org/10.1007/s11251-012-9206-9>
37. Colomer, J., Pallisera, M., Fullana, J., Burriel, M.P., Fernández, R.: Reflective learning in higher education: a comparative analysis. *Procedia-Soc. Behav. Sci.* **93**, 364–370 (2013)
38. Hadwin, A.F., Järvelä, S., Miller, M.: Self-regulated, co-regulated, and socially shared regulation of learning. In: Zimmerman, B.J., Schunk, D.H. (eds.) *Handbook of Selfregulation of Learning and Performance*, pp. 65–84. Routledge, New York (2011)
39. Hattie, J., Timperley, H.: The power of feedback. *Rev. Educ. Res.* **77**(1), 81–112 (2007)
40. Johri, A., Olds, B.M.: Situated engineering learning: bridging engineering education research and the learning sciences. *J. Eng. Educ.* **100**(1), 151–185 (2011)
41. Duderstadt, J.J.: Engineering for a changing world. In: Grasso, D., Burkins, M.B. (eds.) *Holistic Engineering Education*, pp. 17–35. Springer, New York (2010). https://doi.org/10.1007/978-1-4419-1393-7_3
42. Sheppard, S.D., Macatangay, K., Colby, A., Sullivan, W.M.: *Educating engineers: designing for the future of the field*, vol. 2. Jossey-Bass, San Francisco (2008)
43. Adams, R., et al.: Multiple perspectives on engaging future engineers. *J. Eng. Educ.* **100**, 48–88 (2011)
44. Bransford, J., Brown, A., Cocking, R. (eds.): *How People Learn: Brain, Mind, Experience, and School*, Committee on Developments in the Science of Learning, Commission on Behavioral and Social Sciences and Education, NRC. National Academy Press, Washington (2000)
45. Duarte, M.O., Oliveira, I., Félix, H., Carrilho, D., Pereira, A., Direito, I.: Active classrooms: role-playing experience in telecommunications engineering education. *Int. J. Eng. Educ.* **27** (3), 604–609 (2011)
46. Savery, J.R.: Overview of problem-based learning: definitions and distinctions. In: Walker, A., Leary, H., Hmelo-Silver, C., Ertmer, P. (eds.) *Essential Readings in Problem-Based Learning*, pp. 5–16. Purdue University Press, Indiana (2015)
47. Paasivaara, M., Lassenius, C., Damian, D., Rätty, P., Schröter, A.: Teaching students global software engineering skills using distributed scrum. In: 2013 35th International Conference on Software Engineering (ICSE), pp. 1128–1137. IEEE (2013)
48. Kirkwood, A., Price, L.: Technology-enhanced learning and teaching in higher education: what is ‘enhanced’ and how do we know? A critical literature review. *Learn. Media Technol.* **39**(1), 6–36 (2014)
49. Barbosa, L., Alves, P., Barroso, J.: SIDE - teaching support information system. In: 6th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1–6. IEEE (2011)
50. Braun, V., Clarke, V.: Using thematic analysis in psychology. *Qual. Res. Psychol.* **3**(2), 77–101 (2006)