Analyzing self-reflection by Computer Science students to identify bad study habits

Self-reflection performed by students of programming courses on the study habits and skills acquired through b-learning supported by an automatic judge

Filomena Sustelo
Universidade do Algarve
Faro, Portugal
fsustelo@gmail.com

Pedro Guerreiro
Universidade do Algarve
Faro, Portugal
pguerr@gmail.com

***Abstract* — We present some preliminary results and the main conclusions of a study that we conducted at the University of Algarve, for one of the programming courses in the first year of the Computer Science degree at the University of Algarve. We analyzed the self-reflections made by the students about their study habits and about the skills they acquired in the course. This particular course uses a methodology of blended-learning supported by an automatic judge. The research data were obtained through questionnaires that were distributed and collected during the period of study between the end of classes and the exam. We took into account data from other instruments related to previous work carried out by students, in this course and in previous courses, as well as the performance of the students. We intended to ascertain to what extent the planning, motivation, previous study or knowledge about the type of examination influenced final results. The results suggest measures to be implemented in future editions of the course.**

Keywords – Blended-learning, automatic judge, self-reflections, study habits and acquired skills of programming students, learning constrains.

I. INTRODUCTION

In general, the student population enrolling in the Computer Science Programme (CSP) of the University of Algarve (UAlg) is mostly composed by young adults (with 18 or 19 years of age). These students carry to the university the knowledge, skills, attitudes, study habits, motivations, interests and expectations that they acquired in secondary education and from other cultural experiences. They also bring the weaknesses referred in [1], that we must not ignore. In any case, those motivations, interests, and expectations do not always have the expected effect in helping them achieve academic success within the prescribed time. Furthermore, we have observed that accumulated failures as a result of difficulties in developing the skills for writing computer programs that require more than superficial knowledge cause students to stay at the university as adults (say, with more than 24 years of age).

We believe that it is the teacher's responsibility to encourage students in building a deep, responsible and autonomous knowledge for the adoption of rich, diverse, motivating and demanding strategies from a cognitive point of view [2], which aim the gradual and progressive application of andragogy [1] principles (adult education), and not only of pedagogical principles, and to identify factors that can condition the development of the competences necessary for programming. Accordingly, in the academic part of the course Algorithm and Data Structures (ADS) we have been following a blended-learning approach, based on the availability of tools that support the learning process and the evaluation of results, as well as an automatic judge that automatically evaluates the programs written by students [1], thereby releasing teachers to more gratifying pedagogical tasks. Since ADS is the second course in programming in the study plan for Computer Science, students were supposed to master the basic competences that were the subject of the first course, and be able to apply them and learn new ones, such as the following: writing C programs which require the use of libraries of data structures, selecting the best data structures and the best algorithms for the task at hand, and being able to participate in programming contests, by showing that they can solve the problems that typically are presented at such contests. However, Computer Science teachers recognize the students' difficulties on acquiring these competences, and try to help them identifying their fragilities, the cause of these fragilities, and ways to overcome them. In order to fully appreciate the students' difficulties, we targeted this study to identify the cognitive competences required for reaching success in the course, that is, to obtain results that lead to a score above minimum grade.

A. Cognitive competences of programming

Studies based on characterizing cognitive competences of university students often refer to Bloom's six-levels taxonomy: knowledge, understanding, application, analysis, synthesis and evaluation [3].Those studies indicate that students, in general, initiate their learning by a superficial approach, which corresponds to the first three levels, and then evolve in a gradual and progressive way to a deeper approach, which corresponds the last three levels. A more recent version of this

[1] According to Waal and Telles there are five andragogy principles: autonomy, experience, learning interest, learning use and motivation to learning.

taxonomy [4] differs from the original one on the terms used to name competences. There are now referred to by a verb, thus reinforcing the mind activity at each level: know, understand, apply, analyze, evaluate and create. The most meaningful modification is at the two top levels. As shown on the following picture "synthesis" is replaced by "create", a new competence, placed at the top level.

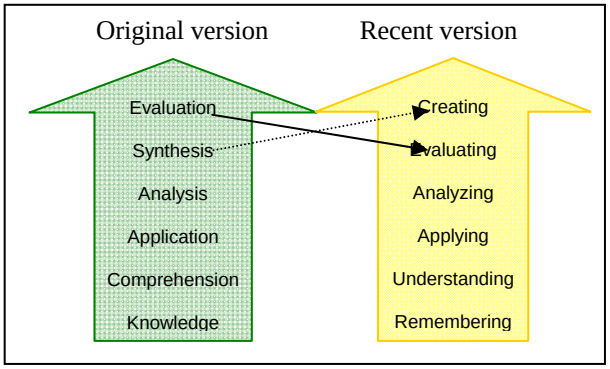

Figure 1. Changes on Bloom's taxonomy

Despite all the technical innovations that come with it, we believe that programming is still mostly a creative activity, encompassing analysis and synthesis [5], requiring a critical assessment, including self-assessment, and generating discussions with peers, centred on the resolution problems through programs and involving an important component of decision-making. All this must be supported by deep reflection and understanding of the basic principles, capable of backing up the decision that has to be made, sooner or later, in selecting the most effective programming solution for the given problem.

For the present work, we relied on Bloom's original taxonomy, which seemed more convenient to describe the natural path of an interested, motivated and hardworking college student, dedicated to learning the basic principles and good practices of programming. The student, when confronted with a hierarchy of skills he has to master, and consequent assessment he will have to go through, will certainly first familiarize himself with lower level competences, which in the case of programming means to know and understand syntactic and semantic rules of language and how to apply them to solve simple programming problems. Usually the exercises presented here should be helpful to:

- Diagnose weaknesses in students, concerning their background knowledge;
- Evaluate the degree of student difficulties in simple programming tasks;
- Elucidate and train students, preferably through effective teaching strategies, that provide them sequential action guides (say, "storylines" with step by step tasks) with the information and clues for gradual and progressive development of higher-level programming skills.

For the application of top level competences, the student should be led to solve challenging problems, which involve interpreting the problem statement, with acuity and attention, keeping in mind the principle of "divide and conquer". This means to decompose the problem into parts, as independent as possible of each other, and analyze their relationships. Assembling the parts, whose reciprocal relations must have been identified, with the aim of reaching the solution of the original problem, is the synthesis competence. Creativity in programming will reside in these two skills. Usually, it is by practicing with self-assessment on the tasks performed and with judging the value and merit of the material they created, that students improve their critical sense and their ability to take the right decisions. This seems to be the natural attitude human beings take in learning and problem solving, either everyday's problems or programming problems. But if this is the natural course of learning, why is it so difficult for the majority of students to acquire the highest level competences in programming [6, 7]? In order to find an answer to this question, we examined the difficulties and considered strategies for developing those competences in students.

1) Impeditive factors to acquire high level competences

For this issue, we considered the impeditive factors for the development of the critical and creative competences necessary to reach full success, on ADS course. We identified the following characteristics for those programming students with low capacity progression on those competences:

1. Lack of effort and persistence to apply the knowledge acquired, as a result of a lack of taste for mathematics or by a low motivation to fulfil these tasks, in this case as a consequence of the perception of difficulties or assumed individual incapacity, or even caused by competing personal interests;
2. Satisfaction on the fulfilment of the minimum requirement for the approval, for the reasons stated on the previous point;
3. Adoption of basic techniques that have been learned in the past, in other contexts, and refusal to try out new, more sophisticated methods, that, nevertheless are more effective and respect more the basic principles and good practices of programming;
4. Limits at the level of abstraction and logical thinking, of mathematical knowledge or other underlying concepts and techniques underpinning the programming problems presented;
5. Weaknesses on planning the workload, and coordinating the various tasks, a competence more important with the widespread adoption of the Bologna model;
6. Concerns caused by the amount of effort necessary to develop all the stated competences.

2) Possibilities to stimulate the development of high level competences

In an attempt to find solutions to some of these problems, we headed for a first analysis of a classic article [8], often quoted in educational circles, where Dijkstra argues that learning is a slow and gradual transformation of "new to usual", reinforcing the idea that learning programming is what he calls the "great news". Jenkins, in [6], supports the argument of Dijkstra and adds that the programming is a subject that is difficult to master, but that it should not be impossible to overcome this obstacle. Other studies promote deep learning [9, 10]. In particular, the study [9] tries to demonstrate that the

strategy "peer review" promotes deep learning in programming courses. The author argues from experience that when students evaluate the work of their colleagues they think deeply, see as others solve the problems, learn to criticize constructively, thus improving the capabilities of critical thinking. The author also refers other strategies to force deep learning, stressing the importance of the implementation of reflective learning in higher education based on the application of knowledge to new situations or in different contexts.

The present work is inserted on a research line whose goal is to evaluate the efforts on the application of b-learning supported by an automatic judge [1] that automatically evaluates the programs written by students, as strategy to promote deep learning, in Computer Science students.

In this article we report the case of the ADS course, analyzing self-reflection exercise made by students in the post-examination period of the academic year 2008-2009, their study and learning habits through b-learning supported by an automatic evaluation tool, during the period of preparation for the exam, which roughly corresponded to a week between the end of classes and the day of the exam. We attempted to identify core competencies in planning the study, effort and action in accordance with the planning and execution of the exam, as factors to develop or maintain in future editions of the subject. The research data were obtained through questionnaires collected from a sample of students who attended the examination. We took into account data from another sample of students on a preliminary study conducted during the period of classes. This study was more concerned with the gradual acquisition of key programming skills by the students, and balanced out the limitations in the study that was carried out in the period of preparation for the exam. We also factored in the results of the students both in the labs (during the period of classes) and in the exam. The questionnaires were implemented as surveys, available through the webpage of the course, at the Moodle learning management system. Both surveys were designed in Google Docs [11] [12]: the first at the end of the subject course work, on the preliminary study; and the second after the examination, on self-reflection after examination. The data were treated in terms of characterizing the profile of the group of students in the course and also the individual profiles, in order to confront the results of each student with the results of the group, and also for comparing the competences that were acquired by each with the key competences that had been identified.

The goal of this research is to instil in upcoming students an awareness of the most common weaknesses, or critical points, to the development of high level competences. Equipped with these findings, the teacher will have a concrete mean to advise or help students in acquiring key competences, which will allow them to leapfrog the learning progresses. Next, we present our case study.

II. STUDY CASE

A. Preliminary study

For the analysis to the preliminary study, were considered 23 questionnaires, gathered at the end of the five weeks of classes, for the ADS course and before the period of preparation for the examination. We expected a larger number of responses, since 136 students enrolled and 56 actually attended classes. We had 17 (74%) male students, 15 (65%) full-time students, and the rest with some kind of part-time occupation (but less than 8 hours/day), 14 (60%) with ages between 18 and 24 and the rest with ages greater than 24 years. We recorded only 4 (17%) respondents that enrolled in this course for the first time, 7 (30%) that enrolled for the second time and 12 (53%) that were enrolled for the third time or more. Apparently, many students who quit on the first week of classes (and did not make it to the exam) have shown signs of lack of preparation and courage to attend this programming course. There was a clear trend of withdrawal from this course by the younger students enrolling for the first time, balanced by a large participation of former dropouts, who continue to enrol year after year, some having already reached an adult age[2]. The most common justifications for the previous failures were: a) lack of basis on mathematics, or other basic subjects, by 12 (52%) respondents; b) the short five-week compact term adopted experimentally by our university, by 11 (48%); c) lack of motivation, little effort and persistence on learning, by 10 (43%); d) high levels of uncertainty specially near the exams, by 7 (30%). So we have identified a group of respondents that are aware of their past weaknesses in programming, but that have shown signs of: a) willingness and persistence to succeed, on the opinion of 13 (54%); b) frustration and anxiety on the opinion of 11 (46%); and c) a tendency to drag the course and postpone success, on the opinion of 9 (38%). The responses do not allow us to conclude that repeated failures are an immediate motive for academic abandonment, with the exception of 5 (21%) that have shown some signs in that direction. We observed that 16 (70%) respondents stated that they enjoy programming, but 13 (57%) did not liked mathematic and, curiously, only 7 (30%) were motivated by both programming and mathematics. We also detected 3 (13%) dramatic cases of students who like neither programming nor mathematics. These are disturbing results, but they match those from [1]. We then analyzed the degree of satisfaction in relation with activities preformed as part of the course, in relation to the usage of Moodle platform, of the automatic evaluation tool, Mooshak [3], and also in relation to the competences acquired, personal interests, programming skills, expectations for the ADS course, and their relation with the results obtained in the mid-exam (an average of 12.4 values), before the final exam.

1) B-learning supported by an automatic evaluation tool

B-Learning, supported by the automatic judge Mooshak, was the learning strategy adopted for the ADS course, to support on-class learning, on the one hand, and to induce autonomous work, on the other hand. The information on course goals, learning outcomes, assessment methods, and types of work required were presented form the beginning. The lectures were recorded on video and made freely available on the Moodle platform, in order to reinforce learning and

[2] Therefore one of the competences to improve, on future editions of the subject, will be to avoid the postponement of success.

[3] *Open-source* automatic judge applicable to programs with text entrances and exits, developed by José Paulo Leal, from Porto University.

stimulate autonomous work at the rhythm of every student. So-called "continuous evaluation" was performed by: 4 "storylines" in total, with a detailed description of the tasks to be performed, matching the decomposition of the given problem in sub-problems, with instructions or suggestions on how to write programs for them and also on how to test the sub-problems and then how to weave the programs for the sub-problems into the full program, this way paving the way towards the gradual and progressive acquisition of high level competences; 3 problems of the style used in programming competitions to apply those competences; and self-evaluation quizzes, after each lecture, to consolidate the subjects taught in the lecture. In this case, 3 (13%) respondents tried to make all the quizzes, 8 (35%) considered them useful, 1 (4%) fun and 7 (30%) have agreed that they helped remembering the issues discussed during the lecture, while 3 (13%) considered some questions took too much time to answer. All programming tasks were evaluated with automatic return, either through Mooshak[4], and either by the Moodle platform. There were also written reports on the "storylines", submitted through platform. These reports were read by the teachers, and scored by hand. Remarks were sent to each student by the teacher, again using the platform. We observed that 14 students (61%) agreed with the evaluation system used in the course, while and 7 students (30%) had some concerns and hesitation about the rules. On the issue of the interaction between teachers and students, 16 students (69%) agreed that the communication through the platform was effective and 12 students (52%) considered that the teacher's face-to-face feedback should be emphasized, again a situation similar to the one of reported in [1]. However, there were 8 students (35%) who stated the course required too much work, and 7 students (30%) revealed uneasiness on time management. Likewise, 16 students (69%) agreed that learning activities took more time than expected. Some respondents said to have acquired, through the use of Mooshak and the Moodle platform, essential study competences, such as: a) more careful on planning activities and meeting deadlines, with 15 (65%) agreements and 5 (21%) hesitations; b) more rigor on the activities fulfilled and submitted, expressed by 15 (65%) agreements and 6 (26%) hesitations; c) more motivated to fulfil the activities, with 16 (70%) agreements and 3 (13%) hesitations; d) 14 (61%) have felt some learning progresses and 6 (26%) hesitated; e) 16 (70%) developed critical spirit, creativity and problem resolution capacity and 6 (26%) hesitated, f) 12 (52%) have recognized that the evaluation system was fair and transparent and 6 (26%) hesitated; g) 10 (43%) agreed that the variety of exercises have measured their knowledge and competences, with 9 (39%) hesitations; and h) 11 (48%) said to have improved their academic performance, despite the 8 (35%) hesitations.

Based on the calculus of the average ranking (AR), through the weighted average of the four items rated using a Likert[5] scale, presented in the previous table, it was showed quite favourable behaviours (with values above 3) on the respondents' attitudes in relation to the study:

TABLE I. AVERAGE RANKING OBTAINED IN THE PREDISPOSITION IN RELATION TO THE STUDY

Aptitudes in relation to previous study	AR	Comportment
1. I actually face the study with seriousness and as an obligation to satisfy my interests	4,1	Favourable (with 17% of uncertain)
2. Usually I fulfil the terms of my study plans, even if arising calls for fun.	4,0	Favourable (with 17% of uncertain)
3. When I can't fulfil the obligations (delivering of works...) I'm frustrated.	4,1	Favourable (with 17% of uncertain)
4. Not passing on is something that bothers me a lot.	4,3	Favourable (with 13% of uncertain)

In this case, 19 students (83%) stated, some (10) with more conviction than others (9), that faced study with seriousness and as an obligation to satisfy their interests, 4 (17%) have revealed hesitations and 1 (4%) depreciated the study. Furthermore, 18 students (78%) said they pursued personal fulfilment and 19 students (82%) wanted autonomy and independence from the family. In this case, only 1 student (4%) showed signs of hedonism (4%), 3 students (13%) showed signs of indifference towards the study, and 1 student (4%) wanted with desire to exert authority, influence and enhance public image. Yet, despite the visibly favourable results, study commitment beyond favourable predisposition requires task planning, effort and actions in accordance with the plan, even if in programming, the capacity for action is, in most cases, a direct consequence of the students' abilities. Accordingly, we sought to analyze the abilities acquired in programming, through the following table:

TABLE II. AVERAGE RANKING OBTAINED ON PROGRAMMING ABILITIES

Abilities in programming	AR	Comportment
1. I'm able to solve the most trivial problems of the subject.	3,6	Favourable (with 26% of uncertain)
2. I know and understand the rules of the C language and I apply them in the most trivial exercise.	3,5	Favourable (with 17% of uncertain)
3. I acquired skills for problem solving typical programming contests.	3,4	Favourable (with30% of uncertain)
4. I'm able to analyze challenging problems, its decomposition, relationships, joint parties with the aim of reaching the solution.	3,3	Favourable (with 39% of uncertain)
5. My logical or mathematical reasoning has been effective in most evaluation work.	3,5	Favourable (with 30% of uncertain)
6. I acquired skills to decide on the best programming solution to a unique problem.	3,3	Favourable (with 35% of uncertain)
7. I rarely commit lapses in programming.	2,1	Unfavourable (with 30% of uncertain)

The table presents an analysis of data provided by respondents in relation to their ability to programming. The existence of positive behaviours can be inferred, although these are not as massive as those found in table I, as shown by AR variables and the high rates of response with hesitation. It

[4] After the submission of each task, this automatic judge returned one of the feedbacks: Accepted, Presentation Error, Runtime Error, Compile Time Error, Time Limit Exceeded, Memory Limit Exceeded, Wrong Answer or Invalid Function.

[5] The scale was scored in a way that 0 was "don't know how to answer", 1 to "strongly disagree" and successively until 5 for "strongly agree". A value lower than 3 for each item corresponds to a favorable comportment of the students; a value below 3 to an unfavorable comportment caused by some contrariety; and value 3 corresponds to an uncertainty comportment, that can be a sign of fragility or indifference.

should be noted that the uncertainty has increased and the AR decreased with the increase complexity of the exercises and of the workload in general. The results point to greater difficulties in acquiring high-level skills, visible in sections 3, 4, 5 and 6 of Table II, that are fundamental to the full success in the subject. The last variable implies unfavourable behaviours, despite the uncertainty of 30%, where it appears that most respondents admitted committing errors on programming caused by lack of attention to detail, or so they think.

We then analyzed the expectations of the students in relation to the final results in the subject, after the examination. In this respect, we found that 13 students (57%) were not very ambitious, and would be happy if they achieved the minimum score for approval, 5 students (22%) were undecided and the remaining 5 students (22%) expressed dissatisfaction in getting minimum levels. In this case, the respondents were trying to be modest when expressing they would be satisfied if they could reach the minimum requirements for approval [6], possibly because of the too many failures in past editions of the course, and also because they are aware of the weaknesses of their programming skills.

B. Self-reflection of students after the ADS examination

After the ADS exam we collected data from 10 questionnaires, from a set of 28 possible respondents who attended the exam. Thirty seven students took the exam, with relatively good performances in the evaluation carried out during the period of classes. Indeed the average for those students that came to the exam was 12.4 points, for a maximum of 20 points. The average score in the exam was 8.0 points. There were 12 that passed, 6 of them with grades above or equal to 12 points (12, 13, 13, 15, 15, 18) and the remaining 6 with a grade between 10 and 11 values (10, 11, 11, 11, 11, 11). Of the remaining, 16 students failed and 9 were absent. These results are very disturbing and more so because of their persistency over the years. It is difficult to find an explanation other than poor study habits of students.

Therefore, it is interesting for the students to do an exercise of self-reflection, after the exam, concerning their study habits, and compare the results with those of the preliminary study. We can also relate the results to the effort and actions conducted in preparation for examination, the examination itself, and if the effort was worth it and the perception of whether the preparation effort was worth it. The data indicated that on the exam 9 respondents were male, 6 were young adults and 4 were adults over 24 years of age.

1) Planning carried out during the pre-preparatory study

The planning of the study is a key skill for any student. Essentially, it consists of a reflection on the preparation, visualization and design of the tasks to be performed. At this point it was noted the following questions had at least 50% of expression of interest: Qpl_1) hot trends, that were likely to be at the exam; Qpl_2) questions frequently asked by the teacher during the lectures; Qpl_3) more profitable time of day; Qpl_4) the favourite place to study, Qpl_6) ways to avoid distracting factors, Qpl_7) definition of learning goals, materials or technology necessary; On the downside questions Qpl_5) profitable study hours and Qpl_8) the distribution of study and rest times and what strategies to use to avoid anxiety states obtained 40% of records. In the respondents reflections after having seen the exam was verified that only 20% of them, corresponding to R1 and R2 of the following table, defended the way they prepared themselves, since considering to have obtained a good result, even if R2 admitted a final result below his expectations. The key issues envisaged by all respondents, and advised to be improved by some or maintained by others, are in the following table:

TABLE III. QUESTIONS PONDERED ON STUDY PLANNING

	Qpl_1)	*Qpl_2)*	*Qpl_3)*	*Qpl_4)*	*Qpl_5)*	*Qpl_6)*	*Qpl_7)*	*Qpl_8)*
Respondents	R1	R1	R1	-	-	R1	R1	-
	R2	-	R2	R2	R2	R2	-	-
	R3	-	R3	R3	-	R3	R3	-
	-	-	R4	R4	-	-	-	R4
	R5	R5	-	-	-	-	R5	R5
	R6	R6	R6	R6	R6	R6	-	-
	-	-	-	-	-	-	R7	R7
	R8	R8	-	R8	R8	R8		R4
	R9	R9	-	-	R9	-	-	-
	R10	-	-	-	-	-	R10	-

2) Action and effort in preparing for the exam

The effectiveness of the preparation for the exam will be greater if the planning is good and if the subject studied during the classes have been consolidated.

Usually this period of preparation for the exam is used to review the course, overcome the perceived remaining difficulties, solve again tests, exercises and problems that have been done, invent of new exercises, etc. At the end of the study, and before the exam, it is important to make a self-assessment of learning, so that the students themselves feel responsible for their study habits and results they will obtain. In the case we analyzed, all respondents used the preparation time to study, as shown in the following table, and the study was always or almost always conducted with the support of materials and technology available on the platform:

TABLE IV. TIME DEDICATED TO STUDY WHILE PREPARING FOR THE EXAM

	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**
Days / Hours of study in a week	**3/40**	**4/80**	**2/14**	**5/15**	**5/45**	**3/12**	**3/12**	**3/42**	**3/15**	**2/10**
Hours devoted to the study were profitable	**fully agree**	**agree**	**agree**	**neither agree nor disagree**	**agree**	**neither agree nor disagree**	**agree**	**agree**	**neither agree nor disagree**	**agree**

The data confirmed that respondent R1 was the one who was confident that the hours dedicated to study were profitable. On the other hand, we observed that in relation to the questions presented here: Qac_1) all respondents reviewed the theoretical materials; Qac_2) 70% analyzed previous exams used in past editions of the course; Qac_3) 50% solved again the exercises presented during the course; Qac_4) only 10% invented new exercises from the ones made available on the platform, the

[6] To avoid satisfaction in obtaining minimum grade on the subject is other competence to be developed in future editions.

remaining 50% despite agreeing to the effectiveness of their study strategy did not show it clearly and there were 40% of hesitation; Qac_5) no respondent claimed to have solved the proposed exercises for which a solution was not available for reference; Qac_6) 50% revealed, but without much conviction, to have made self-assessment of their learning, 10% hesitated and the remaining 40% stated they did not; Qac_7) 30% of respondents said that throughout the study were able to formulate their own questions and that therefore there was no reason for being anxious before and during the exam, 30% hesitate and 40% disagreed; Qac_8) interestingly, 80% of respondents studied alone; Qac_9) for doubts that arose, 40% relied on colleagues (in person or in forums), 30% referred to other means and no one tried to reach the teachers during this period. It is important to notice that 40% of respondents indicated that the effort in the preparation paid off for having gotten a better mark than the one they would have got if they had not prepared themselves. The effort and actions conducted by these respondents is showed in the next table:

TABLE V. ACTION AND EFFORT WHILE STUDYING FOR THE EXAM

Qac_1	*Qac_2)*	*Qac_3)*	*Qac_4)*	*Qac_5)*	*Qac_6)*	*Qac_7)*	*Qac_8)*	*Qac_9)*
R1	R1	-	-	-	R1	R1	R1	-
R3	R3	R3	-	-	-	-	R3	-
R6	R6	-	-	-	R6	-	R6	R6
R7	R7	R7	R7	-	R7	R7	R7	-

By analyzing the Table V, the respondent R7 distinguished himself from the others for, besides all the effort and action in the preparation for the exam, having been the only one to reaffirm that he tried to imagine new exercises from the ones available at the platform and for presenting himself as the more convinced that his study strategy was effective to relax and avoid anxiety. It was found that, like R7, R1 also maintained a positive spirit during study and examination. These two students have thus shown skills of self-motivation and the capacity to formulate their own questions in situations where normally in order to find the answers one must have a thorough knowledge of the subject. Respondents R1, R6 and R7 also performed the self-assessment of their learning, thus taking self-responsibility for their actions in the study. However 40% of the respondents felt that the effort of preparation was not worth it because they obtained a mark below their expectations. The effort and actions conducted by these respondents are given below:

TABLE VI. ACTION AND EFFORT WHILE STUDYING FOR THE EXAM

Qac_1	*Qac_2)*	*Qac_3)*	*Qac_4)*	*Qac_5)*	*Qac_6)*	*Qac_7)*	*Qac_8)*	*Qac_9)*
R2	R2	-	-	-	-	-	-	-
R5	R5	R5	-	-	R5	-	R5	R5
R6	R6	-	-	-	R6	-	R6	R6
R8	-	R8	-	-	-	-	-	R8

In this case, no respondent stood up with enough self-motivation to formulate their own questions, although Table IV shows that these were the ones who have dedicated more time to study during preparation time. We observe that respondent R5 also took personal responsibility for his actions. It is worth noting that R6 respondent considered that this effort allowed him to get a better grade than he would have got if he did not make some effort. Still, he was not happy, because the grade below his expectations.

3) Making the ADS exam

During the exam, we observed that the study strategies used specially by R1 and R7 worked as a key competence to grant success at the course. Next, we present a graphical summary of the main difficulties felt by this group of respondents during the exam:

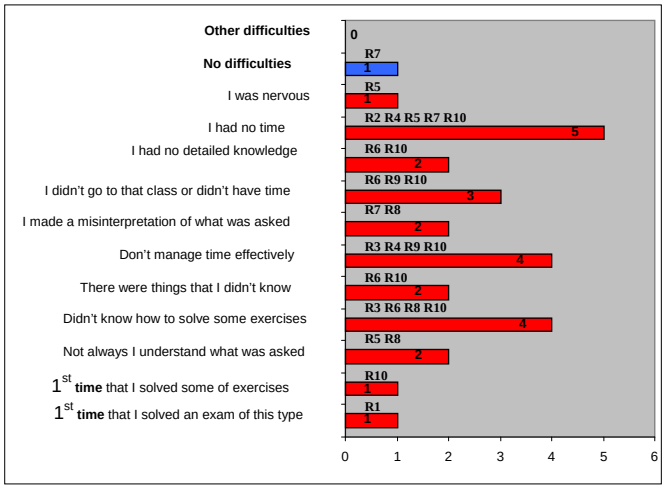

Figure 2. Difficulties on the ADS exam

R7 was the only to affirm not to have felt any difficulties, while R10 was the one who felt them the most; R5 assumed some anxiety. The most limitative factor for this group of students was time. Thus, this must be one of the key competences to be improved on future editions. Like on other areas of study, for exams in programming courses, it is always necessary a deep acquaintance with the subject, so that the right answers come to mind faster, with the self-confidence and self-control. These are fundamental aspects that help to dissolve the fragilities of time management, anxiety, etc.

The exam was structured into five groups, as follows: the first group is a quiz similar to the ones used for self-evaluation questionnaires after each lecture; the second group with a simple question about the output generated by a "mysterious" function that implements in disguise one of the algorithms studied in the course; the third and fourth groups have questions that apply the knowledge acquired in the context of a new set of functions; the fifth group is a problem requiring a complete program, like the tasks submitted to Mooshak, typical of programming contests, using some data structures and some algorithms that the students must have learned during the course. Accordingly, we analyzed the key competences to maintain or improve on the resolution of the first, second, third and fifth groups of the exam (given that the fourth is equivalent to the third):

TABLE VII. STRONG AND CRITICAL POINTS IN THE RESOLUTION OF THE THREE FIRST GROUPS OF THE EXAM

Key competencies to maintain (bold) or to enhance (normal)	**1st Group**	**2nd Group**	**3rd Group**
Many difficulties	R6	-	R8, R10
Withdrawal at first attempt	-	-	-
First time I do these exercises	R1	-	-

I kept calm and was careful	**R1, R3, R4, R6, R8, R9, R10**	**R1, R4, R6, R7, R8, R10**	**R4, R6, R10**
For not to make mistakes I preferred not to respond	R6, R10	-	-
Not studied in detail	R6, R10	R10	R4, R8, R10
I memorized this matter	-	-	-
I recognized the matter and was able to apply it in context of these exercises	**R1, R3, R4, R5, R6, R7, R8**	**R2, R3, R5, R6, R7, R9**	**R1, R2, R3, R6, R7, R9**
I had no detailed knowledge that would allow me to respond well	R2, R6	R10	R5

On the execution of the first group (the quiz), we draw attention to respondents R3, R4 and R8 who revealed more key competences. On the next level we have R1, R5 and R7. On the other hand, R6 was the one who presented more weaknesses on the quiz, followed by R2 and R10. On the two following groups, to implement functions, the emphasis goes to R6 and R7, with key competences to maintain, followed by R2, R3, R9 and R1. On these exercises, R10 was the one with more difficulties, followed by R8, R4 and R5.

TABLE VIII. STRONG AND CRITICAL POINTS, ON A RESOLUTION OF A PROBLEM TO APPLY HIGH LEVEL COMPETENCES

Key competencies to maintain (bold) or to enhance (normal)	**5th Group**
Many difficulties	R9, R10
First time I do these exercises	R10
Withdrawal at first attempt	R10
I kept calm and was careful	**R1**
For not to make mistakes I preferred not to respond	R10
Not studied in detail	R7, R10
I recognized the matter and was able to apply it in context of this problem	**R1, R2, R7**
I had no detailed knowledge that would allow me to respond well	R3, R6, R8, R10
I should have studied the matter thoroughly to respond quickly	R3, R10
I correctly interpreted the statement of the problem	**R1, R4, R6**
I identified the data to solve the problem	**R1, R2, R5, R6, R7, R8, R10**
I was able to build a mental algorithm to solve the problem	**R1, R6, R8**
I decomposed the problem into parts independent of each other	**R1, R2, R3, R6, R7**
I analyzed the relationships of the parties, which could resolve	**R1**
I joined the parties well linked, with the aim of reaching the solution of the problem	**R1, R7**
I'm aware that I presented the most efficient solution that it is possible	-
I had difficulty in applying the knowledge in the context of the problem	R5
I had difficulties in my reasoning	R3
I couldn't reach an initial settlement proposal	R3, R4, R5, R10
I wasn't able to understand what was meant	R10
There were steps which I couldn't perform	R6

In the last group of the examination, on a typical problem of programming contests, respondent R1 stood up positively, showing a deeper understanding of the subject, and its application in the context of the problem presented, despite not being sure that the solution he found was the best. This is a core competence to be developed by all students of this case study, in a course on Algorithms and Data Structures. R7 stood in a significant distance from R1, followed by R2 and R6. In this group, as in others, the respondent who had more difficulties was R10, followed by R3, R4 and R5 who also showed reduced ability to apply knowledge to new situations. Thus, it's not a surprise that the more superficial approaches to the study lead students, whenever asked differently, to difficulties in understanding the meaning of the problems.

In the exam, in general, the respondent who stood with the largest number of core competencies to maintain, and with self-motivation to develop the subject, was clearly R1, followed by R7 and then R2 and R6, although the latter two have presented some weaknesses in the first group. The respondent with more difficulties in terms of the various groups was clearly R10, followed by R3, R4 and R5, although the latter three, with surface trends, presented skills to keep on the quiz, and R3 also in the second and third groups.

4) Post-exam reflection over key aspects to maintain or improve, or if the effort was worth it

After the reflections over how study habits have influenced exam performance, the respondents reflected over key aspects to maintain or improve in the way they prepared themselves. We observed the following:

- 20% (R1 and R2) defended the way they prepare themselves, since they considered to have obtained a good result, although R2 admitted that the effort did not pay, since the grade was below his expectations. On the opposite, R1, R4 and R10 considered their effort rewarding, since they learned things they did not know. However, both R4 and R10, which presented more weaknesses at the level of acquired key competences, considered to change their preparation way by making more exercises;
- 60% (R1, R2, R3, R4, R7 and R10) considered that the effort was rewarding, since they liked programming, with the exception of R2. On the reflection on the aspects they would change on the way they prepared themselves, R3 referred he would study more deeply certain parts of the course, and R7 that he would have made more exercises;
- 30% (R1, R3 and R7) defended that their effort paid off, since they got a better grade than the one they would have if they were not prepared, despite the fact that R3 did not present signs of having a good grade;
- 40% (R2, R5, R6 and R8) considered the effort and action was in vain, since they had a grade lower than they expected. On the reflections of R5 and R8 on what they would change in the way they prepare themselves, they mentioned making more exercises, while R6 would have paid more attention to the quizzes;
- At last, also R9 did not felt any advantages on the effort, considering that he did not have time to prepare for the exam. Still, he referred that he would have made more exercises, if he had had time.

In this case we can conclude that a significant percentage of the respondents assumed the responsibility for bad study habits, caused not by the lack of study hours, but for the lack of a deep study on matters that require more than superficial knowledge. In this case they all agreed that the exam was

within reach for most the respondents, and that it covered the subject discussed in the course, in the lectures and in the labs.

III. CONCLUSION

The set of results presented in this paper may not represent the reality of what is happening in programming courses that are part of Computer Science programs in Portugal. Nonetheless, we believe significant and interesting. The study reveals that, despite concerns expressed by respondents for personal achievement, through the attainment of academic and professional success, and autonomy and independence of families, and a favourable dispositions for the study, only a comparatively small part (21%) of students of the first year programming course we analyzed achieved competencies and skills necessary for writing programs with the required in-depth understanding, which requires more than a superficial knowledge, for full success in this subject. The exercise of self-reflection carried out by these students on their study and learning habits through b-learning supported by an automatic judge reflects the skills of self-knowledge that are part of the experiments and experiences in university. Indeed, they are key skills for the current knowledge society.

It is important that each student remain aware of both the critical points for the development of his high-level skills and of his abilities to manage his one academic path, so that he can prepare himself to successfully face the challenges of university life and reach the goals he has set for himself. It is interesting to note that if a student considers lack of effort as one of his critical points, this understanding will favour him since he knows he has to work harder. Nevertheless, if he has a lack of appetence for programming, a condition that is difficult to overcome by reason alone, he will feel that he is not able to control the situation and his chances to succeed are less.

We conclude by presenting the competences of the more successful respondents, that have emerged of this case study: a) self-motivated to carry out the various course activities, to formulate their own questions and to work independently, b) self-motivated for wanting more than obtaining minimum passing score, and for not procrastinating success in the course, c) self-disciplined and rigorous in planning and implementing activities, as well as in meeting the deadlines, d) with good work habits and effective study strategies, e) with ability to program, f) with handling capabilities of the technology available on the platform, g) with a desire to learn and participate in the forums and in the classroom, h) with the ability to communicate through the technology available on the platform or in person.

REFERENCES

[1] F. Sustelo, P. Guerreiro, “Estratégias de combate das fragilidades e contrariedades comuns nos estudantes que ingressam num curso de informática do Ensino Susperior” 2009. Actas da 4ª Conferência Ibérica de Sistemas e Tecnologias de Informação , 17-20 Junho 2009, Póvoa de Varzim, Portugal. (Em Livro, pp. 601-608. ISBN: 978-989-96247-0-2).

[2] P. Ramsden, Learning to teach in Higher Education, 1992, London, Routledge Press. http://ultibase.rmit.edu.au/Articles/june97/learn1.htm

[3] Bloom, Benjamin S. & David R. Krathwohl. (1956). Taxonomy of educational objectives: The classification of educational goals, by a committee of college and university examiners. Handbook 1: Cognitive domain. New York, Longmans.

[4] Pohl, Michael. (2000). Learning to Think, Thinking to Learn: Models and Strategies to Develop a Classroom Culture of Thinking. Cheltenham, Vic.: Hawker Brownlow

[5] http://ctp.di.fct.unl.pt/~pg/docs/telegramas_s.pdf.

[6] T. Jenkins, “On the difficulty of learning to program”, in *Proc. of the 3rd Annu. LTSN_ICS Conf.*, Loughborough University, United Kingdom, August 2002, pp. 53-58.

[7] Gomes, A. and Mendes, A. , "Learning to program – difficulties and solutions", *International Conference on Engineering Education*, September 2007.

[8] Edsger W. Dijkstra. On the Cruelty of Really Teaching Computing Science. Comm. ACM, Vol.32, pp 1398-1404, 1989. http://www.cs.utexas.edu/users/EWD/ewd10xx/EWD1036.PDF.

[9] Jirarat Sitthiworachart and Mike Joy, "Using Web-based Peer Assessment in Fostering Deep Learning in Computer Programming", *Proceedings of the International Conference on Education and Information Systems, Technologies and Applications (EISTA 2004)*, Orlando, USA, pp. 231-236, 2004. http://www2.warwick.ac.uk/fac/sci/dcs/research/edtech/publications/sitthiworachart04b.pdf.

[10] Campbell, Teaching Strategies to Foster “Deep” Versus “Surface Learning”, http://www.uottawa.ca/academic/cut/options/Nov_98/TeachingStrategies_en.htm, 2004.

[11] http://spreadsheets.google.com/viewform?formkey=dDhFUDRWYzFRSzE1UzFqdXZURGdEQVE6MA..

[12] http://spreadsheets.google.com/viewform?formkey=dHRUUEJVYVI0U1FDM0x5UVlwUE1qQUE6MA..