Analysis of the Teaching-Learning Methodology Adopted in the Introduction to Computer Science Classes

Edna Dias Canedo, Giovanni Almeida Santos and Sérgio Antonio Andrade de Freitas
Faculty UnB Gama (FGA) – Software Engineering - University of Brasília (UnB)
Brasília – DF, Brazil
ednacanedo@unb.br, giovannix@unb.br, sergiofreitas@unb.br

Abstract*—To meet socio-cultural demands imposed by the new millennium and its implications for education, it was necessary for teachers and students to participate actively in the teaching-learning process. In this context, the classes' evaluation by students has been a tool widely used in various universities in different countries. For institutions and teachers, these assessments allow knowing and measuring the results, further analysis of the institutional status, reviewing projects, adjusting goals, diagnosing weaknesses and correcting possible deviations. For the monitoring and improvement of this process, it was necessary to create instruments that aimed to evaluate the performance of the proposed changes. The teaching-learning methodology adopted in the Introduction to Computer Science classes may be a process that makes it difficult to understand the principles of language programming for undergraduate students in Computer Science and related areas, generating high failure and course drop out rates. This paper presents an analysis of the results obtained in the Introduction to Computer Science classes taught in five Engineering courses, of the Faculty of Gama (FGA) at University of Brasília (UnB). It was analyzed the evaluation questionnaire answered by the undergraduate students in 2017, perform validation of the questionnaire, check the level of students satisfaction in relation to the evaluated subject and the association among the level of satisfaction, the percentage of practical activities of the discipline, student performance and the level of absenteeism. 519 responses were studied, 63% of students reported satisfaction with the classes taken, however a significant number of students 37% reported dissatisfaction with assessment practices and the dynamics of the classes.

Keywords*—Teaching-learning methodology; Language programming; Evaluation ; Failure and Drop out.

I. INTRODUCTION

The teaching-learning methodology adopted in the Introduction to Computer Science classes may be a process that makes difficult to understand the principles of programming language for undergraduate students in Computer Science and related areas, generating high failure rates and drop out of the course.

There are myriads of proposals and research related to the fields of Engineering, Technology, and Education, with the objective of making available resources and contexts that help professors and students make learning more effective. As for the leaning of programming languages, the advances in this research context are easily identifiable, with countless tools and environments presented as proposals for better learning [1].

Regardless of the countless resources and tools in existence, at the UnB Gama Campus (FGA) of the University of Brasília, we concluded that a high number of failing students in the basic programming course is a reality for our five undergraduate courses, namely: Aerospace Engineering, Automotive Engineering, Energy Engineering, Electronic Engineering, and Software Engineering.

The option to start the different Engineering courses of the FGA is common to all candidates. Currently FGA has 3 forms of entrance for the courses of Engineering:

1) Traditional Entrance Exam: it occurs in the middle of the year;

2) Serial Evaluation Program (PAS): PAS occurs in 03 stages. The first test is done during the first year of high school, the second test during the second year of high school, and the third test during the third year of high school;

3) Unified Selection System (SISU): Through the note of the National High School Exam (ENEM).

Each semester, FGA receives an average of 280 new students and they will be studying common subjects that belong to the basic cycle and, in the third semester, they will decide which course they will take among the five engineering branches. The courses have duration of 10 semesters (5 years).

In the period from 2008.2 to 2013.1 the Introduction to Computer Science class was taught in the first semester of the course. Since 2013.2 this class has been offered in the second semester of the course, with its workload increased from 4 to 6 credits, as well as its name changed to Basic Computing (CB). The content taught in the discipline (subject matter) has not changed; only the semester in which the discipline is offered to the student.

This semester offer change was based on the attempt to improve the performance obtained by the students, thus seeking to reduce the number of disapproval occurred in this class and / or dropping out of the course. Offering this class in the second semester of the course, it was expected that the student would have a greater maturity to understand the basic concepts of programming language discussed in class.

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

Biannually 6 classes of CB are offered, with different teachers and methodologies. Some of them teach Programming Language in C and others in Python Programming Language. The collected and analyzed results show that the semester change of the course did not decrease the number of failure, quite the opposite, there was an increase in the failure, compared to the period in which the class was being offered in the first semester of the courses.

The results show that it is necessary to seek new teaching-learning proposals to introduce the concepts of programming language for the students of an Engineering course, seeking to reduce the number of failures and dropping out of the course.

This paper presents an analysis of the results obtained in the Introduction to Computer Science subject, taught for the five Engineering courses of the FGA, between 2008.2 and 2013.1, and the Basic Computing subject, between 2013.2 and 2016.2. Besides that, a survey in the form of a questionnaire is also given to the students of the current semester, 2017.1, regarding the teaching techniques adopted by the professors of the subject, which is now called Algorithms and Computer Programming, with the objective of identifying improvements in the teaching-learning process.

The remainder of this paper is organized as follows. Section 2 presents the background. Section 3 presents the analysis of the scenario of the ICC and CB subjects. Section 4 presents the evaluation and analysis of results obtained with the survey given to FGA students. Section 5 presents the conclusions and future studies.

II. BACKGROUND

It is known that in the academia there is an intense effort by researchers and professors to try to comprehend the reason behind the difficulty in learning programming concepts. This difficulty of comprehension is an obstacle for an ever higher number of students [2,3]. Not only that, there is an effort from the leaders of Teaching Institutions and the Government to provide better learning conditions throughout the undergraduate course. Moreover, there has been much debate among computer science professors as to which programming language should be taught to students, for example, adopting an object-first or imperative first approach [16].

[18] is performing a systematic review of introductory programming literature, in an effort to statistically consolidate further quantitative evidence on the often cited worldwide high failure rates of programming courses. The contributions of this work were:

1) Verify the findings of Bennedsen and Caspersen [17].

2) Demonstrate that failure rates in introductory programming courses have not significantly improved over time.

3) Explore possible moderators of failure rates, including:country, grade level, language, and cohort size.

The study of Bennedsen and Caspersen [17] was based upon surveying the authors of selected conference papers and performing a statistical analysis of the responses. The study [18] was based upon performing a systematic review of the literature on introductory programming courses, and performing a statistical analysis of the data extracted from relevant articles.

The study [20] describes the course that was developed to introduce all first-year engineering students to the fundamentals of computer programming within the context of solving engineering problems. The course was designed to utilize active learning techniques by having the students complete a series of laboratory exercises and projects that introduce computer programming and engineering applications. This study describes the origins of the course, the laboratory exercises and projects, how the course was administered, and an assessment of how successful the course was based on student grades, student feedback, and a student survey. The results indicate that the course increased students' knowledge of programming in the context of solving engineering problems.

Clearly teaching effectiveness is a highly complex and very personal process involving a multitude of variables [19]. The study only attempts to empirically examine the role of one possible factor, the level of faculty research productivity. However, unlike the vast majority of previous empirical studies that simply used student perceptions of teaching, they employed the results of a standardized and quantified student learning outcome assessment process. Few, if any, empirical studies exist that utilize this type of school-wide standardized student outcome measures for teaching effectiveness [19].

In the governmental level, programs with social and digital inclusion policies are offered, both for college and high school students. Other investments include renovating the infrastructure of Universities, Federal Institutes, and Federal and Local Public Schools, improving the technology and equipment of the class rooms and laboratories, along with several programs to incentivize teacher qualification. Even with the amount of programs and research, there are still problems in learning programming.

A. Learning Programming Languages

At the University of Brasília (UnB), there's an agreement between professors and researchers involved with the Intro to Programming subject that learning to program is not a trivial activity, since it introduces a series of cognitive requirements to the daily life of the student, going further than technical requirements.

In their majority, these cognitive requirements incorporate a need for the student to change their way of thinking and acting in their academic life to a different reality than the one they got used to during high school.

If we consider that they're met with this different way of thinking in the first semester of their course, changes need to be incorporated/absorbed in a short period of time, since the subject lasts for a semester. The subject load is of 60 hours of class that usually begin in March and finish in the end of June, or begin in August and finish in the beginning of December of each year.

The cognitive requirements pointed out by the academia are:

- The resolution of problems is a competency that involves cognitive processes such as creativity and rationality, through a set of mental meta-skills that sometimes go unnoticed, and which are supported by other skills, such as reading and interpreting the description of a problem [4,5];
- The full understanding of the requirements of a programming paradigm is not a trivial activity, and entails an inherent degree of difficulty [4,5]. The understanding involves abstraction and problem-solving skills.

The focus of teaching evaluation should be the educator's ability to create the conditions necessary to optimize student learning. It is on this premise that some researchers seek to determine what is "good" teaching. For example, the good practice in teaching and learning emphasize the following seven principles [15]:

1) *Encourages contact between students and faculty;*
2) *Develops reciprocity and cooperation among students;*
3) *Encourages active learning;*
4) *Gives prompt feedback;*
5) *Emphasizes time on task;*
6) *Communicates high expectations;*
7) *Respects diverse talents and ways of learning.*

Based on these seven principles previous work has categorized "good teaching" into three distinct facts:

1) *Creation of a student-centered learning environment;*
2) *Appropriate assessment of student learning and;*
3) *Sufficient preparation for teaching and learning.*

Generally, the issue with learning isn't on the language or the programming paradigm chosen by the professor to be taught in class, or even in the relationship between professor and student, but in the difficulty to develop skills that make the student able to contextualize their knowledge in order to abstract and solve problems.

A majority of professors of programming subjects try to make the student understand that to program is, first and foremost, an exercise of basic reasoning skills (reading, writing, and calculating) and mental skills (comparing, describing, interpreting, classifying, and analyzing) that develop over constant practice and exercise. It is necessary to put in a considerable amount of time to pass a programming subject, especially in the first semesters of an undergraduate course.

B. Programming Subjects at the FGA

Several programming subjects are offered in the FGA courses. The first contact for the student comes with the Basic Computing subject, which will be called Algorithms and Computer Programming (APC) starting 2017.1, even though it will maintain the syllabus and workload.

In the second semester of the course, the Software Development subject is offered and, in the third, Object-oriented Programming, focusing on developing programs using Java. Afterwards, the students take Data Structure I and II. Other related subjects include Algorithm Analysis and Project, Programming Techniques for Emerging Platforms, Programming Paradigms, and Programming for Parallel and Distributed Systems. The student's education in programming is reinforced over the course, with all these subjects and in every semester.

C. Projects that aid the learning of Programming at the FGA

The ICC, CB, and APC subjects have the support of many assistants (students who have already taken the subject and passed with excellent marks), aiding the students with the objective of providing a fast and efficient learning process. This guarantees students are well cared for, be it when clearing doubts about the course contents, or when receiving help to carry out exercises and other tasks. The subjects of ICC, CB and APC do not have different content; they have the same syllabus and are taught using the same programming language in a given semester.

The current syllabus of the subject is: Fundamental principles of program creation; Algorithm design and its representation in pseudo code and high-level languages; Notions in abstraction; Specification of variables and functions; Tests and debugging; Solution patterns in programming; Notions of structural programming; Identifiers and types; Operators and expressions; Control structures: conditional and repetition; Data entry and exit; Statistical structures of data: homogeneous and heterogeneous aggregates;. Iteration and recursion; Notions of cost and complexity analysis; Systematic development and program implementation; Structure, debugging, tests, and documentation of programs; Problem Resolution; Applications in real scenarios and environmental issues.

At the FGA, there is a project called Educational Support System (SAE), which arose from studies involving professors and students from three Brazilian institutes of higher learning (UCB, UFRGS and PUC-RS). The project aims at employing technological resources adapted to a coherent educational methodology in a multidisciplinary project (Education, Psychology, and Computing).

The main technological resource of the SAE is an educational software that allows for an adequate interaction through the internet. As such, it can be used as a support system for teaching-learning either in class or from home, as well as in a combination of both [6]. This tool is used by some teachers in the ICC, CB, and APC subjects at the FGA.

Another project in place was the Educational Robotics Laboratory, whose main ideal was to provide a space for the integration between Engineering students at the FGA and high schoolers of the Distrito Federal. This laboratory has educational robotics kits and didactic material available to the

students, in such a manner that experiments can be done and solutions to proposed problems, elaborated. This project was sponsored by The National Council for Scientific and Technological Development (CNPq).

III. Scenario Analysis of the ICC and CB Subjects

Teaching evaluation is described as the process whereby the quality of teaching is assessed. This measure of assessment can be conducted using formative and/or summative approaches. In higher education, formative assessments of teaching are focused on providing instructors/professors with information that can help them to improve their teaching [14].

In the first academic term of the FGA (2008.2), students of the Intro to Computer Science (ICC) subject had a failure rate of 25%. In Figure 1, it can be seen that this rate went up with each passing term. In 2010.2, the failure rate was at 56%. In the term in which the ICC subject was offered, 2013.1, this rate was at 54%.

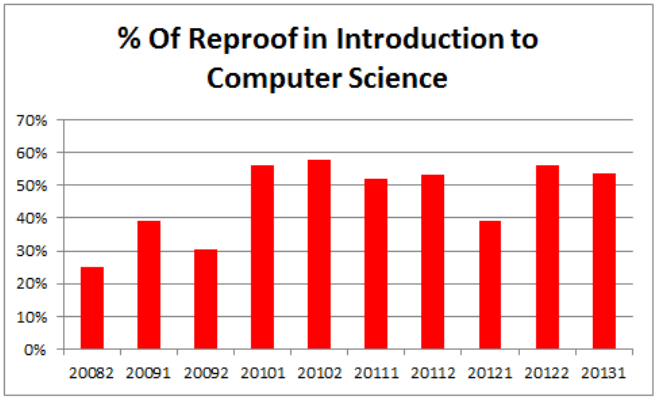

Fig. 1. Percentage of students who failed the Introduction to Computer Science (ICC) subject

Starting with the second term of 2013 (20132), the subject was renamed Basic Computing (CB). The failure rate in that semester was of 65%, was shown in Figure 2. In the second semester of 2015 (20152), this rate decreased to 36%, rising back up and finishing 2016 (20162) with a rate of 43%.

Although there are many initiatives at the FGA looking to reduce the percentage in this scenario, the results at hand are not satisfactory.

Starting at the first term of 2017, the CB subject was rethought and renamed Algorithms and Computer Programming (APC). The monitoring of the failure rate will remain throughout the new subject. There were changes to the syllabus and workload of the subject.

With the objective of understanding the rising number of failing students, a survey – in the form of a questionnaire – was made available to all FGA students. The survey should shed some light on the current scenario and possible shortcomings of existing results-oriented programs.

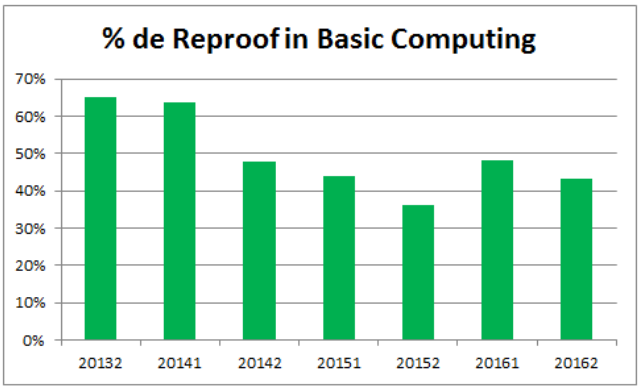

Fig. 2. Percentage of students who failed the Basic Computing (CB) subject

A. Evaluation of Education by the Students

There are national and international validated questionnaires that are used to evaluate, in a broad and formal way, higher learning institutions as well as any given subject in its Pedagogical Course Project (PPC).

The majority of these questionnaires evaluates not only the teaching-learning process, but also the physical and administrative structures, being used to verify the conditions offered to the students by the Educational Institutions. The international form of educational evaluation by students is known as SETS (student evaluation of teaching) [7], and is composed mainly by four core tools, each of them with unique characteristics and institutional objectives, as shown in Table 1.

TABLE 1. SETS (STUDENT EVALUATION OF TEACHING) [7].

Authors	Year of Publication	Name	Country of Origin	Characteristics of Tools
Lucas et al	1997	The Module Experience Questionnaire (MEQ)	England	Effects of the size of the modules with student involvement and learning experiences within the modules.
Ramsden	1991	Course Evaluation questionnaire (CEQ).	Australia	Explores quality teaching experiences in learning programs.
Hounsellet al	2005	The Experiences of Teaching and Learning Questionnaire (ETLQ)	Edinburgh	Focuses in exploring the notions of students in teaching-learning environments, and in what was learnt by the student in a specific moment of the course, unit, or module.
Herbert and Marsh	1992	The Students' Evaluation of Educational Quality (SEEQ)	Australia	Evaluates the quality of teaching in individual modular courses, based new approaches in effective teaching defined by Marsh.

The Brazilian framework established by the federal government is the National Exam for Student Performance (ENADE), a component of the SINAES which, besides cognitive contents, also evaluates the infrastructure and opens up space with some questions about the teaching-learning process [8]. Since is a tool that deals with the big picture, in order to deal with the courses of various nature, it brings information with little specificity for the evaluated Institutions.

Even though there is a series of questionnaires available for the evaluation of teaching, these tools prioritize the acquisition of global data on teaching in these institutions, not focusing in the satisfaction of the student in relation to the general and specific aspects of the subjects offered throughout their education.

The students' satisfaction with a given subject, when reviewed in literature, regards the subject as a whole, exploring aspects such as the infrastructure, the professor, teaching strategies and methodologies, professor evaluation, different ways of grading in tests and assignments, breaching psychological and skill aspects [9], [10], [11].

B. *Evaluation Questionnaire of the ICC/CB/APC Subject*

The evaluation tool came to be thanks to the interest by some FGA professors, in 2016, in knowing the level of satisfaction of FGA's students in relation to the introductory programming subject and in the potential of information that could be raised in a student-focused subject evaluation.

A group of professors of FGA's Software Engineering course developed an internal tool to evaluate these subjects. The main advantage is of creating one's own questionnaires is that they can correspond to the college's traits and necessities. The group was formed by professors Edna Dias Canedo, Giovanni Almeida Santos and Sérgio Antônio Andrade de Freitas. After many literature reviews and group discussions, the first version of the tool was tested between the faculty and then made available for all FGA students.

The tool is a questionnaire (https://canedo.typeform.com/to/ti15IB) that seeks to assess the satisfaction of the student in relation to the different domains of the subject. This includes an overview of the subject and various specific aspects of it, in a quantitative way with open space for a qualitative expression. Thereby, as they evaluate the subject, the student actively participates in identifying the problem related to the high level of failure and high dropout rates and, especially, in the discussion of possible improvements. This allows the student to acquire ways of analyzing their action in a more critical manner, making them take a role of responsibility in the teaching-learning process [12], [13]. To the faculty, it allows for the diagnosis of the status of the discipline in the Engineering courses at the FGA, detecting its frailties, potential, problems, and successes.

At the launch of the questionnaire, there was considerable resistance by the faculty. Many felt threatened, as they believed it would focus on investigating and punishing them for their teaching roles; they believed the students wouldn't be competent enough to evaluate their teaching methodology and so on.

The main objective of this study was to identify the reasons for such a high level of failure of our Engineering students in their first contact with a programming language subject. The objective was to understand if the issue is related to the syllabus, the programming language itself, the professors' teaching method and the resources used by them, or the infrastructure made available at the university.

IV. Evaluation and Analysis of Results

It is known that the student pass rate for a certain subject does not necessarily mean that the teaching methodology or computational resources employed by the professor are successful. Therefore, how to evaluate the success and benefits achieved in student learning thanks to a certain teaching methodology?

As a way to evaluate the results achieved with the methodology adopted by the professors of ICC, CB, and APC, some research questions were defined to compose an investigative and evaluative questionnaire. This questionnaire was made available to all students of the current APC classes, and to FGA Engineering students who had already taken CB or ICC.

In total, **519 students** answered the questionnaire. The obtained results were as follows:

1) In relation to the subject taken – ICC/CB/APC

Of the total number of students who answered the questionnaire, 21% took ICC, 43% took CB, and 36% are taking APC in the current semester. This is the first semester in which it is offered. 99% of the students took the C Language subject. Figure 3 represents the result of this question.

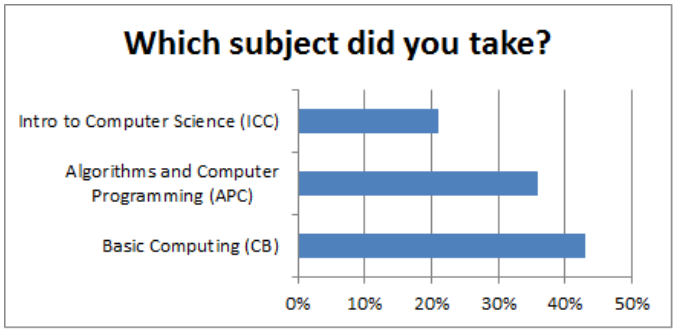

Fig.3. Introduction to Programming Subject taken by students

2) In relation to finding the subject challenging, considering learning and passing rates.

13% of the students considered the subject a little challenging. 66% considered it to be one of the most challenging, and 21% considered it very challenging. Without a doubt, the subject is challenging for most students, since some of them don't grasp its real necessity for the course, especially non-Software Engineering courses. Figure 4 presents the results of the survey.

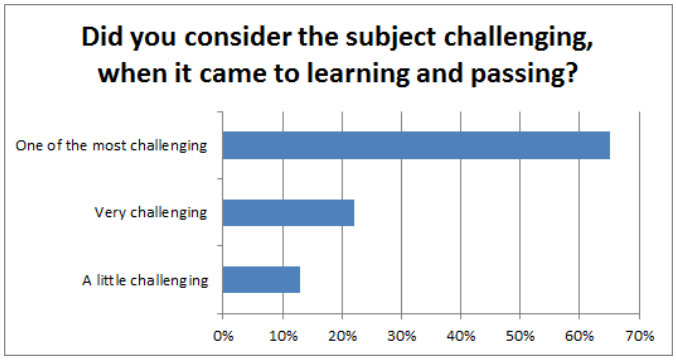

Fig.4. Students' Answer to the question: Did you consider the subject challenging, in regards to learning and passing rates?

3) In regards to the teaching methodology of the professors assigned to the ICC, IB, or APC subjects.

32% agree that the professor assigned to either subject has good teaching methodologies. 20% completely agree to the statement, 17% were neutral, and 14% completely disagree with it. 17% disagree that the professor had good teaching methodologies. Figure 5 presents the results of student satisfaction in relation to the methodology adopted by the professor in the subject.

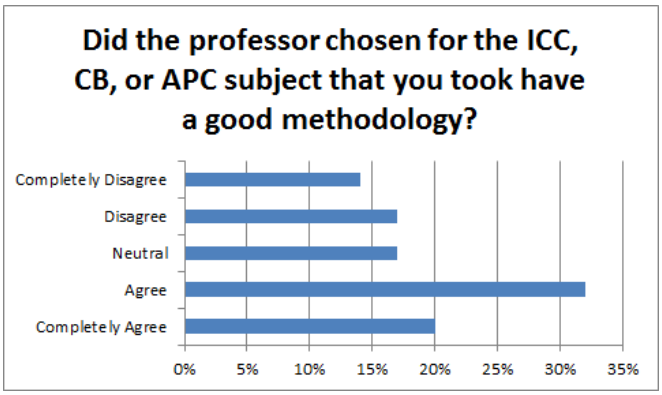

Fig.5. Opinion of the students in regards to the methodology of the professor assigned to the ICC, CB, or APC subjects.

4) Problems listed as being negatively impactful in students' learning.

Within the listed problems, the ones with the highest negative impact mentioned the students are: the professor's methodological aspects at 75% and the amount of students in the classes at 65%. Currently, classes have between 65 and 80 students. 67% claim that internet access quality is a problem. 65% mentioned the obsolete machines found in the laboratories, while 67% blame the course material made available by the professors, and 55% mentioned the lack of practical exercises. Lack of air-conditioning at 58%. Figure 6 presents the result related to this question.

5) Is the infrastructure (laboratories) provided by the FGA a good environment to take the ICC/CB or APC subjects?

11% completely agree, 49% agree, and 17% are neutral. Meanwhile, 19% disagree and 4% completely disagree that the infrastructure offered by the FGA is an adequate environment for the learning of the subject. Figure 7 presents the obtained results.

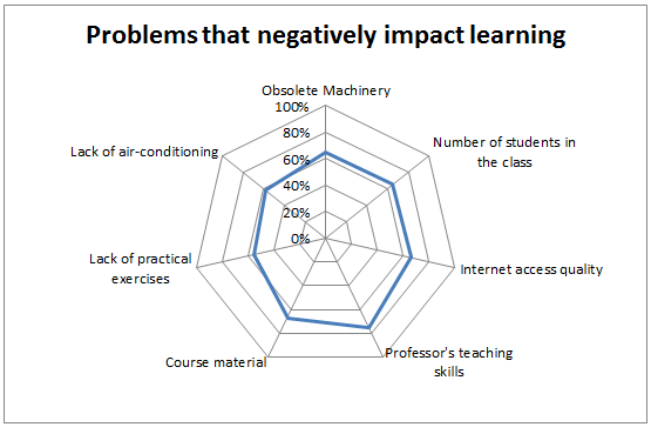

Fig.6. Factors that negatively impact student learning.

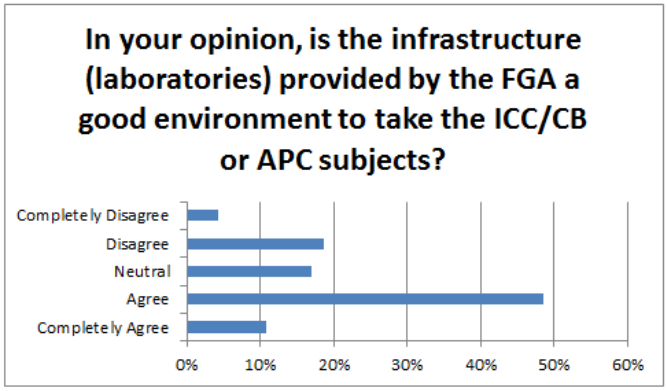

Fig.7. Infrastructure of laboratories offered by the FGA

6) Is the support by the assistants enough to learn the subject?

8% of the students completely agree and 32% agree that there is enough support. 30% are neutral about it. 8% completely disagree and 21% disagree that the support by assistants is enough to learn the subject. Figure 8 shows this scenario.

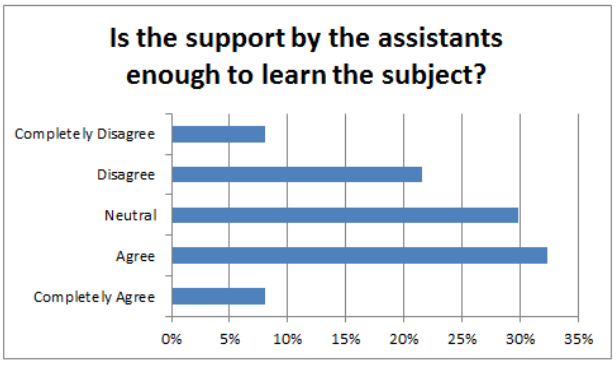

Fig.8. Is the support by the assistants enough to learn the subject?

7) Is the teaching practice adopted by the professors adequate to the level of difficulty of the subject?

In regards to the teaching practice adopted by the professors of ICC, CB, and APC being adequate to the level of difficulty of the subject, 13% of the students completely agree and 33% agree. 19% of the students are neutral. 12% of the students completely disagree, and 23% disagree that the teaching practice is adequate. Figure 9 presents the results related to the level of difficulty of the subject.

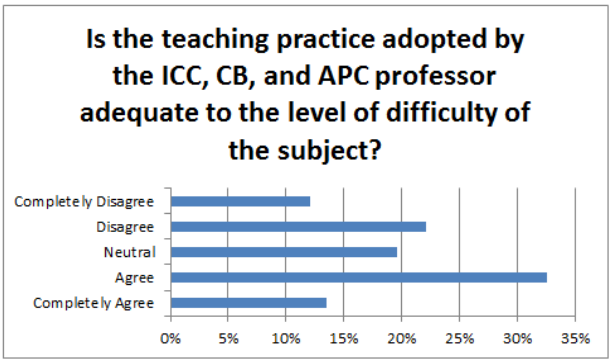

Fig.9. Adequacy of the teaching practice adopted by the professor to the level of difficulty of the subject.

V. CONCLUSION

Analyzing the information from the database of UnB's academic system, it's possible to see a high rate of failure in the ICC and CB subjects, even with a change in the workload and subject syllabus. In changing from 4 to 6 credits in 2013.2, the failure rate jumped from 54% to 65%.

It's necessary to identify what causes such scenario. The questionnaire allowed us to understand a bit more of this situation thanks to the students' answers.

The students considered the campus' infrastructure a negative factor that needs to be remedied. There are many broken machines in the laboratories, which calls for an investment for better lab equipment. It's also necessary to rethink the number of students in each class. A reduced number of students would allow for a closer and more personal teaching practice by the professor and better, more efficient understanding by the student. Besides, the material made available by the professors needs to be improved, adding more practice exercises and their respective solutions.

Within the currently adopted practices, we can identify the ones that are well received by students, such as:

1) The inclusion of assistants: 71% of the students claim they look for the assistants to look for help with the subject.

2) Perception of the importance of the subject: 88% say that the Algorithms and Computer Programming subject is very important for their course.

3) Related to the use of a methodology: 37% of the students assure they noticed that the teacher used some teaching methodology or practice.

4) Leadership: 55% of the students believe the professor of the subject possessed leadership in relation to the class.

It's worth remembering that this is an initial evaluation/monitoring, which we intend to continue over the next five years and establish if there was a change in the current scenario. We will also check if the change in culture and teaching methodology brings results, providing positive change in the subject's evaluation.

ACKNOWLEDGMENT

The authors of this paper would like to thank all the students of the FGA who answered our survey and sent us reports and improvement proposals for the research.

REFERENCES

[1] M. Scheila W., M. António J., F. D. Antonio. Comunidades de Investigação em Programação: Uma Estratégia de Apoio ao Aprendizado Inicial de Programação. IEEE-RITA Vol. 5, Núm. 1, Feb. 2010.

[2] T. Jenkins, "On the difficulty of learning to program," in 3rd Annual Conference of Learning and Teaching Support Network of Centre for Information and Computer Science LTSN-ICS, (United Kingdom), pp. 27-29, Loughborough University, The Higher Education Academy, Agosto 2002.

[3] E. Lahtinen, K. Ala-Mutka, and H.-M. Jaarvinen, "A study of difficulties of novice programmers," in ITiCSE '05: Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education, (New York, NY, USA), pp. 14-18, ACM Press, 2005.

[4] B. C. Wilson and S. Shrock, "Contributing to success in an introductory computer science course: a study of twelve factors," SIGCSE Bull., vol. 33, no. 1, pp. 184-188, 2001.

[5] S. Wiedenbeck, "Factors affecting the success of non-majors in learning to program," in ICER '05: Proceedings of the First international workshop on Computing education research, (New York, NY, USA), pp. 13-24, ACM, 2005.

[6] S., Giovanni Almeida, and V. Roberto Vilardi Rissoli. "Benefícios no Uso de um Assistente Inteligente no Ensino-Aprendizagem de Programação Computacional." Anais do XXII Simpósio Brasileiro de Informática na Educação. Aracaju (2011): 2244-2253.

[7] E. Keane, Labhrainn I. Obtaining student feedback on teaching & course quality. Brie ing paper. 2005;2:1-19.

[8] INEP. Disponível em< http://www. inep. gov. br>. Acesso em Março de 2017.

[9] P. Cohen. Student ratings of instruction and student achievement: A meta-analysis of multisection validity studies. Review of Educational Research. 1981;51(3):281-309.

[10] Al-Jishi E, Khalek N, Hamdy H. Students' perceptions of the effectiveness of a professional skills program in preparation for clerkship training. Educ Health. 2009;22(2):1-7.

[11] D. Kahneman, Fredrickson BL, Schreiber CA, Redelmeier DA. When more pain is preferred to less: Adding a better end. Psychological Science. 1993;4(6):401-405.

[12] MC. Grillo, de Medeiros MF. A construção conhecimento e sua mediação metodológica: Edipucrs; 1998.

[13] D. Enricone, Grillo M. Avaliação, uma discussão em aberto. 2ª ed. Porto Alegre: Edipucrs; 2003.

[14] P. Pitterson, Nicole, et al. "Investigating current approaches to assessing teaching evaluation in engineering departments." Frontiers in Education Conference (FIE), 2016 IEEE. IEEE, 2016.

[15] A. W. Chickering and Z. F. Gamson, "Seven principles for good practice in undergraduate education," AAHE Bull., pp. 3–7, 1987.

[16] A. Ehlert and C. Schulte, "Comparison of oop first and oop later: first results regarding the role of comfort level". In Proc. ITiCSE, pages 108-112. ACM, 2010.

[17] J. Bennedsen and M. E. Caspersen, “Failure rates in introductory programming”. ACM SIGCSE Bulletin, 39(2), page 32-36, 2007.

[18] W. Christopher and F. W.B. Li, “Failure rates in introductory programming revisited.”, in Proceedings of the 2014 conference on Innovation technology in computer science education (ITiCSE '14). New York: Association for Computing Machinery (ACM), pp. 39-44, 2014.

[19] G. Craig S., and G. B. Merrill. "Faculty research productivity and standardized student learning outcomes in a university teaching environment: A Bayesian analysis of relationships." Studies in Higher Education 37.4, page: 469-480, 2012.

[20] J. P. Hoffbeck, H. E. Dillon, R. J. Albright, W. Lu, T. A. Doughty, "Teaching programming in the context of solving engineering problems." Frontiers in Education Conference (FIE), 2016 IEEE. IEEE, 2016.