

Developing Problem-Solving Skills in Engineering Through a Programming Methodology: A Team-Work Visual Approach

Elizabeth Vidal¹[0000-0002-8367-9439], Milagros Zegarra¹[0000-0001-8415-3247], Ricardo Gacitúa²[0000-0002-3900-2770], Mauricio Dieguez²[0000-0001-8728-7101]

¹ Universidad Nacional de San Agustín de Arequipa

{evidald, mzegarra}@unsa.edu.pe

² Universidad de la Frontera

{ricardo.gacitua, mauricio.dieguez}@ufrontera.cl

Abstract. Engineers are problem solvers whether they are involved in software, electronic, analytical, civil, experimental, or design work. International accreditation institutions were strong in the student outcome programs that engineering graduates must have an ability to identify, formulate, and solve engineering problems. Our work describes the experience of developing problem-solving skills in the first year of Engineering using a Programming Methodology that makes a strong emphasis on visual components and team-work: R&G Methodology. The R&G Methodology is a methodological proposal that guides the analysis and solution of problems in a systematic way. Canvas templates allow students to write ideas while solving a problem. Students define, explore, analyze the problem, and graphically design solutions. R&G seeks the development of cognitive skills in the student, instead of mastering the syntax of a specific programming language, promoting logical-computational thinking, the decomposition of processes, the modeling of solutions, the identification and use of patterns, among others. We have to drive quasi-experimental with a sample of 70 students from Ingeniería de Sistemas and Ingeniería Electrónica. The p-value obtained in the experimental group shows that there is an improvement regarding to problem-solving skills when students work in team for solving-problems under a structured methodology. Since programming courses are part of most of the Engineering Curriculum we consider our experience could be valuable for other Engineering.

Keywords: Problem-solving., Team-work., Programming Methodology., Visual-Methodology.

1 Introduction

In the Engineering field, teaching programming is a way of teaching problem-solving. Future graduates must realize that there are multiple solutions to a given problem and that selecting among them is not a purely technical activity. Many students have

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

difficulty analyzing problems, formulating a solution, and converting it into a solution written in a programming language. Researchers have long been interested in problem-solving [1-3] and the relationship between problem-solving skills and programming skills [4-5]. Lishinski et al. [6] use a theory-based approach to measure problem-solving as a means of predicting performance in an introductory undergraduate programming course. Ambrosio et al. [4] found that differences in problem-solving skills are due to cognitive processes and mental organizations and that some characteristics shared by good problem solvers are: reasoning, critical thinking, perception, memory, creative thinking, and reading comprehension. On the other hand, Kirschner et al. [6], ensures that learning requires mental representations and schemes that are subject to the amount of information provided. To address this problem, it seems reasonable to think that the use of appropriate systematic methodologies for teaching programming, which explicitly considers the construction of mental representations and schemes that allow focusing first on analyzing the problems, and then, through a guided process, students can get the solution.

Based on Kirschner et al. [7] foundations we have used the programming methodology “Rapid and Global” (R&G) that proposed problem-solving defined in four main phases: define, solve, convert, and coding. This methodology allows early inclusion of teamwork and is based on the use of canvas for strength visual representation of the problem and the proposed solution. The proposed research question is: *Does the use of a Programming Methodology based on team-work allows the development of problem-solving skills?*

2 Background

2.1 Problem Solving

Problem-solving skills for introductory programming courses refer to understanding how to apply knowledge to solve real problems, not just writing code and moving bits. Students must realize that there are multiple solutions to a given problem. Students should also be able to communicate their solution to others, including why and how a solution solves the problem and what assumptions were made [8]. Woods et al [9] state that students who solve problems exhibit specific attributes such as: (a) they are willing to spend time reading, gathering information, and defining the problem (b) they emphasize precision over speed (c) they write down ideas and create boxes/figures while solving the problem (d) they are organized and systematic (e) they are flexible (keep options open, you can see a situation from different perspectives/point of view) (f) they are willing to risk and deal with ambiguity, welcoming change and managing stress.

2.2 Rapid and Global (R&G) Methodology

Rapid and Global (R&G) methodology [10] was developed by the third author and has been tested in Chile and Peru. This methodology is problem-solving oriented and also allows promoting the early inclusion of teamwork. The methodology, as illustrated in Fig. 1, defines four main phases: define, solve, convert, and encode. Each stage uses canvas templates that guide students in identifying, describing, and specifying items that make up an algorithm. R&G has specific templates for each phase. The configuration of the templates allows solving problems as part of a team where students have to set goals (identify the problem), plan tasks (propose a high-level algorithm) and meet objectives (solve the problem) The first to phases are related to the solving problem and the third and four are related to translating the solution to code. We describe the first and second phase since they are the focus of our study.

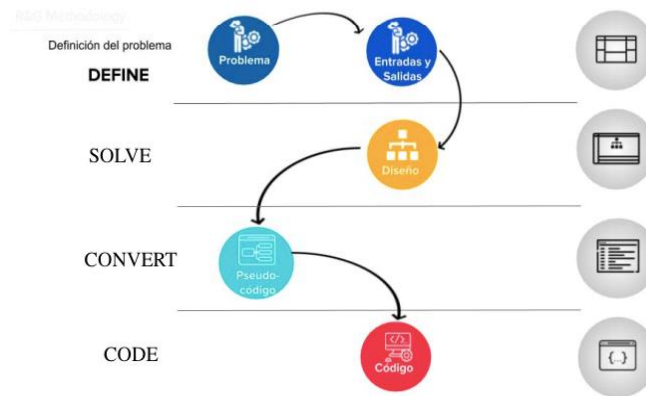


Fig. 1 R&G Methodology phases: Define, Solve, Convert and Coding. Define and Solve is related to problem-solving.

Define: The first step of the methodology is the identification and description of the problem. To make this identification, we must see what the problem is, what data we need to solve it (inputs) and what data will be obtained as a response after solving it (outputs). We must also have identified the considerations that the problem requires. This stage is important as misidentification of the problem can lead us to "solve" non-existent or incorrect problems. This is why this stage is extremely important. Once the problem has been identified, the main conditions or requirements that must be met for its solution must also be identified and in this way structure a solution, which will have a series of activities (reads, loops, storage and reuse of the information generated) In the case of programming, it is quite useful to use functions, procedures, subprograms or methods that have been created and tested before, as it facilitates a certain part of the solution process. Finally, a preliminary outline is made of how this solution process will be carried out, both in the conceptual part and in the interface. To carry out this scheme, we use the R&G Canvas template (Fig 2) in which we can identify and indicate the inputs, outputs and processes at the level of sentences, activities and modules.

R&G – Canvas: Analysis		Author:	Date:
Problem:			
Input:	Process:	Output:	
Input Screen:	Functions:	Output Screen:	

Fig. 2 R&G Canvas Analysis. It helps to define the problem; the first step is to identify and describe the problem. Answer the questions (i) What is the problem?, (ii) What are the inputs?, (iii) What are the outputs? (iv) How is the process? (v) Are there predefined functions that we can reuse? This step helps students understand and describe what problem they need to solve.

Solve: It corresponds to conceptual refinement where students use principles of abstraction, successive refinement, and modularity. Its purpose is to identify the sequence of individual actions, called activities. The basis for this step is based on the premise that "nothing is particularly difficult when divided into small tasks." The problem (main activity) is divided into 3 sub-activities (Input, Development, and Output). Fig 3 illustrates the Design Canvas, called R&G-Canvas-Design, used to represent the solution graphically.

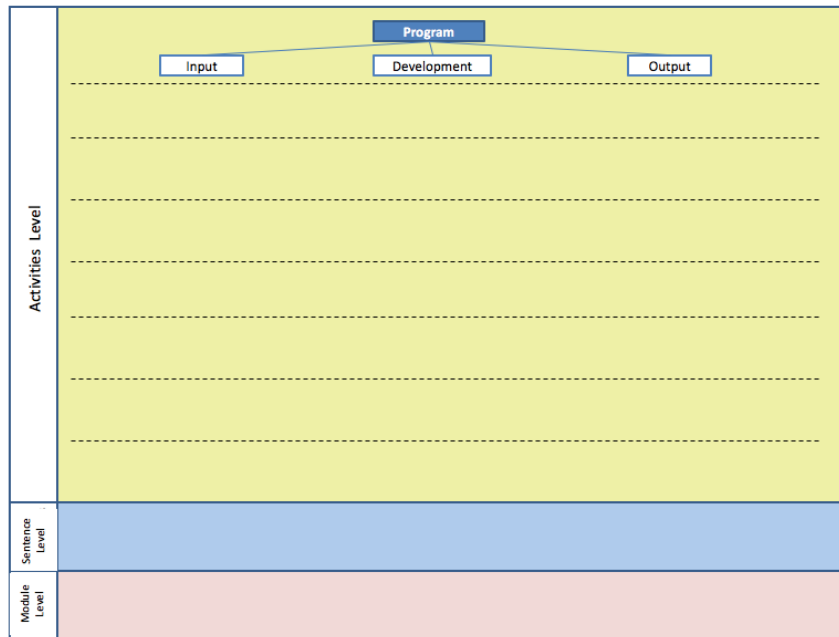


Fig. 3 R&G Canvas Design. It helps to describe the design of the solution proposed, through a graphic language, in which the activities that solve the problem are detailed. At this stage, it is possible to obtain the set of instructions that solve the problem, from successive refinements of activities that have a high level of abstraction.

3 Methodology

3.1 Description

Problem-solving skills with respect to introductory programming courses refer to understanding how to apply knowledge to solve real problems. The use of the R&G methodology could allow students to develop problem-solving skills through teamwork. The first step is to identify and describe the problem. The second step identifies what data we need to solve it (input) and what data we will get as a response after solving it (output). The third step describes the main process for solving the problem, and finally, students must identify if there are pre-defined solutions that can be reused. While working in teams, students realize that there are multiple points of view for a given problem. Working as a team would help students to have a broader understanding of the problem. Using Canvas Analysis (Fig 2) allows students to focus on specific parts of the problem: inputs, outputs, and processes. Also, students can write down their initial ideas. Canvas templates strengthen students' ability to work as a team, as they can establish an area where they can identify all input results and parts of the process and then carry out discussion or deliberation on the best solution alternatives (Fig 4)



Fig. 4 Students using Canvas Design. It is used to discuss group discussion for a proposed solutions in a graphic approach.

3.2 Materials and Methods

This experience was implemented in Ingeniería de Sistemas and Ingeniería Electrónica undergraduate at the Universidad Nacional de San Agustín de Arequipa (UNSA) in 2019. The study corresponds to the use of R&G methodology in the first programming course that is based on the ACM CS2013 curriculum international recommendations [8]. From the learning outcomes related to problem-solving described by the ACM CS2013 [8] we focused on: (1) Discuss the importance of algorithms in the problem-solving process, (2) Discuss how a problem can be solved by multiple algorithms, (3) Create algorithms to solve simple problems, (4) Use a programming language to implement, test and debug algorithms to solve simple problems and (5) Apply decomposition techniques to divide a program into smaller parts.

The first programming course is taught in the first year of study. It lasts 17 weeks with 4 academic hours and 2 laboratory hours. Students have no prior programming experience. Students attended laboratory sessions in groups of 20. In the laboratory sessions, students were asked students to solve weekly assignments. Student grades are based on two midterm exams, a final exam, and laboratory assignments. Student grades are evaluated from zero to twenty in Peru.

To answer our research question **¿Does the use of R&G Methodology based on teamwork allows the development of problem-solving skills?** we conducted a quasi-experimental study. The study population is established in the set of students of the first year of studies of the mentioned careers. We have considered a non-probabilistic convenience sampling with 120 students. The students are between 17 and 20 years old without previous experience in programming. The total number of students in the sample was: 50 students for the control group and 70 students for the experimental group. The control group had traditional programming laboratory assignments (individual assignments). Students in the experimental group were required to complete

their laboratory assignments using the R&G methodology (team-work). Students in the experimental group were randomly assigned in groups of four to six.

4 Results and Discussion

The application of the pre-test and post-test was divided in two dimensions: a) Problem identification (define the problem, explore the problem, and plan the solution) and b) Problem solution (implement the plan and verify the solution). The objective was to verify if there were differences in the results regarding the scores between both groups and if these were significant. We presume that:

HI: There is a significant difference between the scores obtained in the pre-test and post-test of students who were exposed to the R&G Methodology.

To test the hypothesis, we use a T-Student test for two independent samples. We verified the criterion of the normal distribution of the data (through the Kolmogorov-Smirnov and Shapiro-Wilk tests). The criterion for deciding was: if the probability obtained from the P-value \leq alpha (5%), HI is accepted. If the probability obtained P-value $>$ alpha (5%) HI is rejected. Table 1 summarizes the pre-test and post-test averages for each of the two dimensions, both for the control group and for the experimental group.

Table 1 Results

Group	Dimension	Pre-test	Post-test	P-Value
Control Group	Problem Identification	11.75	12.78	0.380
	Problem Solution	11.58	11.58	1
Experimental Group	Problem Identification	8.19	16.48	0.000
	Problem Solution	11.14	14.88	0.01

It is observed that the P-value obtained in the two tests for the control group does not show a significant difference between the scores obtained. On the other hand, the experimental group presents a P-value in the two tests of less than 5% that shows a significant difference in the scores obtained after having used the R&G Methodology in both dimensions. The results show us that the application of the methodology based on team-work generates a positive impact on the development of problem-solving skills. A review of the literature has found other approaches that make use of programming to develop problem-solving, such as media computation or game orientation [12-15] or even the use of applications such as Google Earth [16]. The related works had shown important results in the development of problem-solving through programming courses, but we have observed that the focus is oriented on

motivational issues. In the related works [12-16], the problem-solving practice is carried out individually and focused on the product (code) and not on the process of how to solve the problem. The main difference from our experience is the work-team base that allows students to discuss different points of view and to observe that a problem can have many solutions. Likewise, we consider that the graphic component through the use of canvas facilitates the discussion. Having a process that guides the steps for problem-solving and facilitated discussion is a good tool for students.

5 Conclusion

The results obtained showed us that the use of a Programming Methodology based on team-work could help in the development of problem-solving skills, in our case we made use of R&G Methodology that guides the analysis and systematic problem-solving. The canvas templates provide by R&G used in the problem definition and problem design phases allowed students to write down ideas while solving a problem. The students define, explore, analyze the problem, and design solutions graphically. We can affirm that the R&G Methodology could allow students to enhance the development of cognitive abilities. The limitations of the use of the R&G Methodology are given in the availability of the canvas template which must be provided by the professor to achieve a greater impact on the discussion process. While they can be printed on individual sheets, we have found that it has a greater impact when students discuss in groups with a large canvas. The methodology can be incorporated into any programming course independent of the programming language to be used (Java, c ++, Python, etc.) since it focuses on problem-solving and not coding. As future work, it seeks to extend the application to other Engineering courses that present strong elements of problem-solving and to see its application beyond the programming courses.

References

1. Frederiksen N.: Implications of Cognitive Theory for Instruction in Problem Solving. *Review of Educational Research*, 54(3):363–407, (1984)
2. Murphy P., McCormick R.: Problem solving in science and technology education. *Research in Science Education*, 27(3), 461–481, 1997.
3. Palumbo, D. Programming Language / Problem-Solving Research: A Review of Relevant Issues. *Review of Educational Research*, 60(1):65, (1990)
4. Ambrosio, A. P.; Costa, F. M.; Almeida, L., Franco, A.; Macedo, J.: Identifying cognitive abilities to improve cs1 outcome, in: 2011 Frontiers in Education Conference (FIE), IEEE, (2011), pp. F3G–1
5. Morazan, M. T.: Functional video games in the cs1 classroom, in: International Symposium on Trends in Functional Programming, Springer, (2010), pp. 166– 183
6. Lishinski, A.; Yadav, A.; Enbody, R.; Good, J.: The influence of problem solving abilities on students' performance on different assessment tasks in cs1, in: Proceedings of the 47th ACM technical symposium on computing science education, ACM, (2016), pp. 329–334

7. Kirschner, P.; Sweller, J.; Clark, R.: Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching, *Educational Psychologist* 41. (2006) doi:10.1207/s15326985ep4102_1
8. ACM Curricula Recommendations: Computer Science Curricula 2013, <https://www.acm.org/binaries/content/assets/education/cs2013> , last accessed 2020/11/28
9. Woods R., Hrymak; A. N., Marshall; R. R., Wood, P. E.; Crowe, C. M. ; Hoffman, T.W., Wright, J. D.; Taylor, P.A; Woodhouse, K.A.; Bouchard, C.G.K: Developing problem solving skills: The McMaster problem solving program. *ASEE J of Engng Educ.*, 86, 2, 75-91 1997.
10. Gacitúa, R.; Diéguez, M.; Vidal, E.: Forming software architects in early stages: From craft to engineering, in: 2017 36th International Conference of the Chilean Computer Science Society, pp. 1–8. Chile, (2017) [http:// www.doi.org/10.1109/SCCC.2017.8405130](http://www.doi.org/10.1109/SCCC.2017.8405130).
11. Simon, B.; Kinnunen, P. ; Porter, L., Zazkis.; D.: Experience report: CS1 for majors with media computation. In *ITiCSE*, pp. 214–218, (2010).
12. Corral, J., Balcells, A. ; Estévez, A. ; Moreno G. ; Ramos M.: Game-based approach to the teaching of object-oriented programming languages. *Computers & Education*, 73, pp. 83-92, (2014).
13. Sung, K.; Pace, J.: Building casual game SDKS for teaching CS1/2: a case study. *Journal of Computing Sciences in Colleges*, 32(1), pp. 129- 143, (2016)
14. Vidal, E.: Teaching the First Programming Course with Python's Turtle Graphic Library. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education ACM*, pp. 244- 245. Arequipa-Perú (2016). <https://doi.org/10.1145/2899415.2925499>
15. Vidal, E., Castro Gutierrez, E., Aedo, M.: When the Robot Meets the Turtle: A Gentle Introduction to Algorithms and Functions. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, pp. 78-79. Bologna-Italy (2017). <https://doi.org/10.1145/3059009.3072974>
16. Vidal, E.; Gutierrez, E. C., Erkan, A.: Using Google Earth and KML to teach Fundamental Object Oriented Programming. In 2019 38th International Conference of the Chilean Computer Science Society, pp. 1-4. Chile (2019). <http://doig.org/10.1109/SCCC49216.2019.8966389>