Problem-based learning. An experience on the inclusion of quality problems in Educational Software Engineering

Yasnalla Rivero Peña
Faculty of Mathematics and Informatics
University of Holguin
Holguin, Cuba
yasnalla@uho.edu.cu

Jesús Rafael Hechavarría Hernández
Faculty of Architecture and Design
Catholic University of Santiago de Guayaquil
Guayaquil, Ecuador
jesus.hechavarria@cu.ucsg.edu.ec

Maikel Leyva Vázquez
Faculty of Mathematical and Physical Sciences
University of Guayaquil
Guayaquil, Ecuador
mleyvaz@gmail.com

***Abstract*— This article presents an experience developed with students of fourth year of the Computer Engineering major with the support of educative software in the teaching-learning process of quality problems in Software Engineering. It also incorporates a procedure as a guide during their laboratory practices to perform the process of quality measurement to software selected as case studies with the support of the tool developed. It shows that the development of educative software based on the pedagogical model of Problem-Based Learning, PBL, promotes in university students the motivation towards the search and production of knowledge in virtual spaces and environments with innovative characteristics. The results of the analysis of opinions of specialists in software development, as well as of professors who teach the subject, indicate that the computer tool contributes to self-improvement through the requirement of the use of specialized tools in quality issues in order to perform the or the roles in the execution of a project, mainly in the workflows dedicated to quality management.**

Keywords*— quality; quality metrics; problem-based learning; software engineering ; educational software

I. INTRODUCTION

The application of Information and Communication Technologies (ICTs) in teaching and learning are a great potential to increase access, quality and success of the educational process. Higher education has experienced during the last decades a constant process of transformation, stimulated by the changes that are generated in the different spheres of political and social life.

An innovative university is the antithesis of static organizations, subject to models that belong to the past. It is the university that is permanently reformed. Innovating on the part of the university also means systematically updating, at the undergraduate and postgraduate levels, curricula, teaching methods, evaluation methods, relations between students and professors; incorporate the most advanced and relevant technologies for educational purposes, among other aspects [1]. In this principle, their work is based on Cuban universities committed to achieving graduates at the height of his time.

In the field of Information Technology, training for the workplace requires students to achieve certain competencies associated with the roles they play in the exercise of the profession [2]. These modes of action show that computer engineer requires the ability to interact with others to achieve common goals. Generally, it is known as a "role" within a software development team. The most common roles are: leader, analyst, developer and tester. To achieve this, the university must be responsible for contributing, modernizing, diversifying and promoting the knowledge and development of skills that are associated with each of the roles and means involved in the activities of teachers and students.

This paper presents a procedure and educative software for the solution of quality problems during the evaluation of software developed at the University of Holguin, Cuba. They defined a set of actions, roles and artifacts that guide the work with quality metrics. To do this, a review of the bodies of knowledge related to software measurements was carried out, as well as the curricular guides and curricula of the Software Engineering at the undergraduate level. The guide to the practices tested was supported by an ABP approach, applied to Software Engineering.

II. SKILLS QUALITY MANAGEMENT SOFTWARE TO DEVELOP IN THE FORMATION OF COMPUTER ENGINEER

A. Curricular guidelines of the curriculum related to elements of software quality

The inclusion of topics and basic concepts related to the process of quality measurement are essential in a Software Engineering curriculum. SWEBOK (Software Engineering Body of Knowledge) is a guide that frames the contents of the discipline of Software Engineering. One of the objectives of SWEBOK is to provide the basis for the development of study curricula [3]. These guides provide indications to educators about the content of the syllabi, how to teach content in different contexts, and the skills that undergraduate students need to cope with software development processes.

The study plan D of the Computer Engineering Major at the University of Holguín guarantees this preparation based on the elements proposed in the main integrating discipline "Engineering and Software Management". The training of these professionals is conditioned by the need to satisfy the essential use of the best techniques of Engineering and Software Management; therefore, among its main instructional objectives are:

- Develop habits and skills of quality management software.
- Learn the techniques of verification and validation of software and its role in assurance and quality control of software projects.
- Identify and evaluate software quality metrics for the different workflows of the process [4].

The fulfillment of these objectives favors the training of software engineers specialized in issues related to quality management.

B. Problem-based teaching as a strategy for learning software engineering

Problem-based learning is widely regarded as a successful and innovative method for engineering education. As for the PBL, Barrows [5] methodology defines it as a learning method based on the principle of using problems as a starting point for the acquisition and integration of new knowledge. Its fundamental characteristics were set by this author, among them we can mention: learning is centered on the student, occurs in small groups, teachers are facilitators or guides, problems are the focus of organization and stimulus for learning and the new information is acquired through self-directed learning [6].

This methodology favors the possibility of integrating different subjects or academic disciplines to try to solve a real world problem. When engineering students work in a problem-based learning environment they learn to solve real-life problems by developing technological solutions. Problem solving is the backbone of engineering practice, and therefore, problem-based learning models have been singled out as a powerful tool for engineering education communities to promote employability [7].

The dominance of computer engineers, to develop habits and skills of quality management software; as well as, plan and control quality. It is essential in the modes of action of graduates when transiting working life and quality managers in software projects. The use of educative software to train, learn and perform real evaluation and measurement activities in software development environments can improve student performance, since it allows the experience that is not offered theoretically in learning.

In the elaboration of quality problems for the laboratory practices of the students of the fourth year of the Computer Engineering career of the University of Holguin, the characteristics of the work under the PBL model were taken into account. Where the teacher and the student have clearly defined roles and responsibilities. The execution of these exercises is the development of a set of competences related to quality measurements to different software projects taken as case studies. The student must observe the behavior of each evaluated characteristic and its incidence in the proper functioning of the software, time in which the quantitative and qualitative volume of data necessary to make the written report of the evaluation of the assigned project is generated.

III. TOOLS TO ENCOURAGE THE TEACHING OF QUALITY THEMES

A. General conception of the procedure to evaluate the comformity of computer products

The procedure proposed in Figure 1 goes through four phases that govern the software conformity assessment process. In each of them, steps, objectives, contents, tasks and techniques to be used are declared. The stages of the procedure are the following:

Phase I. Diagnosis

Objective: to check if the software conformity evaluation that is carried out in the organization corresponds with the use of the established quality metrics. In addition, define the main organizational characteristics of the projects in the organization based on: the existing documentation, the opinion of users, developers and other personnel involved in this activity.

Phase II. Planning

Objective: to prepare the evaluation guide through a set of activities that are articulated with each other and planned in advance, with the purpose of guiding how they are going to proceed [8]. This guide defines a set of actions (tasks), human resources (roles) and materials (artifacts) to guide the work of the evaluators.

Phase III. Measuring

Objective: to start the measurement process as one of the fundamental sub-processes in software evaluation. We proceed to calculate the metrics with the support of the Metric_calc.exe tool and subsequently to interpret its results.

Phase IV. Report

Objective: in this phase the work team reaches a conclusion on the degree of conformity that the product that was being evaluated had to give a conclusive verdict. The conclusions and recommendations in the evaluation process are shown according to the results obtained in the measurement process.

The procedure is applicable to any software. In its application, the operational stage in which it finds itself is not decisive.

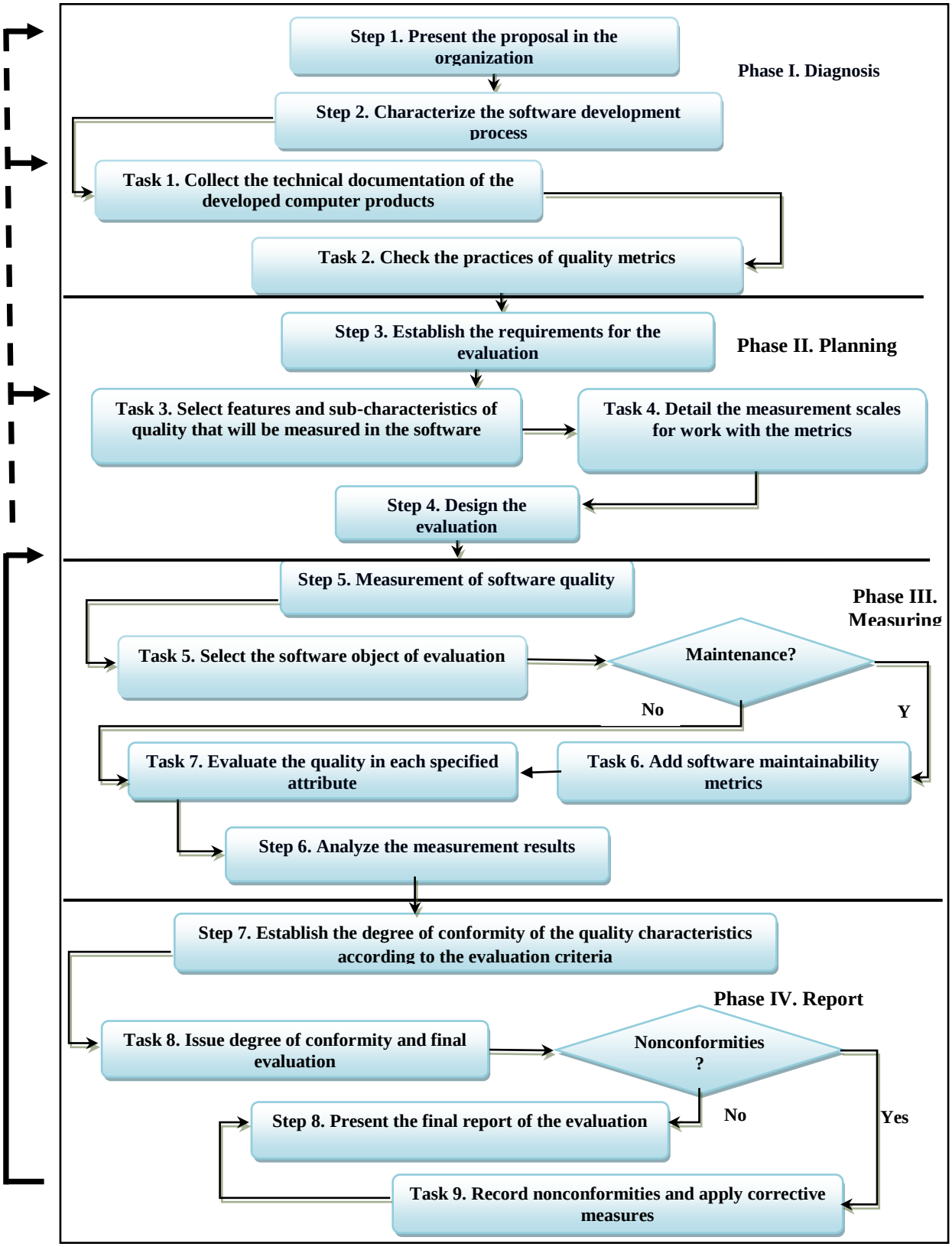

Fig. 1. Procedure for conformity assessment [9].

B. Characteristics of the tool for the calculation of quality metrics

The Metricalc.exe tool is a desktop application developed to support the calculation of quality metrics in the measurement phase of the procedure proposed by [9]. The characteristics that are evaluated with the tool are:

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability

Among the normative references used for its conception, it is worth highlighting the NC ISO / IEC 9126-1: 2005 Software Engineering. Product quality. Part 1: Quality model and Part 2: External metrics and the ISO / IEC 14598-evaluation model. Part 3: Procedure for developers and Part 5: Procedure for evaluators.

The design of the application facilitates the work of the evaluators and has the scales, formulas and evaluation criteria to support this activity. The conception of the tool is based on a friendly application, easily accessible, since it has interfaces and a user manual.

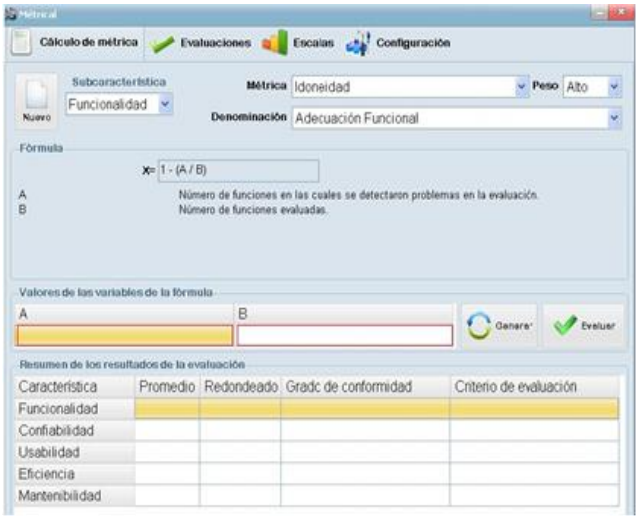

Fig. 2. Main interface of the Metric_cal.exe tool

C. Lab with the use of the tool as educative software

The good practices for the engineers are useful and the concrete examples from the study of cases, for that reason the students always need from the exercise of their profession to specify knowledge. For the development of the activity, the professor will make the presentation of the tool, the configurations to be taken into account in the software; as well as, the roles that students and the teacher will play in the process of measuring the quality of different projects taken as case studies.

The measurement process is carried out through the user interface shown in Figure 2. It consists of selecting a characteristic and granting each specified attribute its respective weight. The assigned weights depend on the incidence that this characteristic has on the operation of the product being evaluated. The following scale is suggested: 0 (low), 1 (medium) and 2 (high). The scales used were the result of works published by [10] and those published later by the same authors together with Cruz Torres in [11]. These data are indispensable for the subsequent calculation with the metrics and are averaged with the objective of obtaining a qualitative as well as a quantitative view of the quality behavior of that characteristic.

Fig. 3. Interface of the scales to be used in the calculation of the metrics

The use of Metric_calc.exe facilitates the computer engineer the implementation of roles (Table 1), as part of a multidisciplinary team during workflows dedicated to quality management.

Table I. Roles and responsibilities declared in the procedure [8].

Roles	Responsibilities
Coordinator of the quality assurance team	The person who exercises this role should have knowledge and experience in software development, since this role is responsible for monitoring and controlling the activities carried out by the quality assurance team, in relation to the evaluation of the practices of development and the software product [12].
Evaluator	The person appointed to this role in the work team must have solid knowledge of software evaluation and quality metrics, as it will be responsible for

	executing the activities of the measurement and evaluation processes.
Tester	This role, designated by the coordinator of the quality assurance team, will test the software. In addition, it will execute the test cases and analyze the results of the tests to record future non-conformities that the customer may present. [13].

During the development of the activity, the professor acts as coordinator of the quality assurance team and the student as evaluator and tester. In the practice of measurement, a history of evaluations is used (Figure 4). The student must observe the behavior of each evaluated characteristic and its incidence in the proper functioning of the software, time in which the quantitative and qualitative volume of data necessary to make the written report of the evaluation of the assigned project is generated.

Fig. 4. Evaluation history interface

The objective of this exercise, with software support, is the development of a set of competences related to quality measurements to different software projects taken as case studies. In practice students must make a series of adjustments, surveys and measurements on a particular configuration of the instrument, with this is intended to ensure that students can have a simulation tool referred to the presentation and description of the quality metrics used for the development of the practices In addition, through metric_cal.exe and its interfaces learn to enter and interpret mathematical data to then issue evaluation criteria.

D. Results and discussion

In this exercise, two practices were developed that integrate the PBL methodology in an experimental way, the first of them incorporates a process of monitoring the behavior of specific characteristics of a software during the test stage and the second deals with a process of analysis of the characteristics evaluated previously. The proposed practices have a guide and, in addition, some tutorials about the experiment and the configuration of the software associated with the practice are included in the computer.

The teachers considered that the use of educational software for the teaching of quality issues constitutes a good teaching strategy to follow in the subject of software engineering because it allowed them to have another vision about the development of their laboratory practices in the subject. It was demonstrated that this teaching methodology will favor learning and allow students to prepare for their insertion in professional life. It would also enable developing transversal competences such as teamwork, individual and group responsibility and self-learning.

As a support to the positive evaluation of the practical results of the research, the tool was used in the professional practices of the students of the fourth year of the Computer Engineering course of the 2016-2017 course for the evaluation of the quality and compliance of software in the following companies: the National Company of Projects and Engineering (ENPA), Electric Company, Trading Company and Distributor of Medicines (ENCOMED), among others.

From the statistical processing of the aspects treated in the survey with the experts and the exploitation of the tool by the students in the development of their curricular activity, people agree that all aspects are "Very Relevant", which positively influences the quality of the tool and the objectives set with the research.

In this way, it can be concluded that the degree of satisfaction of users is high, as well as the evaluation that they issue about the product made, related to: the organization, uniformity, consistency and availability of the information provided. In order to continue perfecting the teaching-learning process and give continuity to the work done, the authors are open to migrating the Metric_calc.exe tool to open source, which will allow a more timely exploitation of the application.

IV. CONCLUSIONS

The tool proposed as support to the teaching and learning process of software quality issues, based on the pedagogical model of Problem Based Learning, guarantees the systematization and consolidation of quality issues in the computer engineer training related to the modes of acting in a software project. The figure of the teacher in the ABP model changes to take the responsibility of a tutor, accompanying the learning process as a driving force and acting as an educational facilitator who must transmit a set of values and attitudes to conduct the professional's training.

REFERENCES

[1] R. Alarcón , "Universidad innovadora por un desarrollo humano sostenible: mirando al 2030," La Habana,Cuba , Editorial Félix Varela, 2016.

[2] V. Muñoz, A. Hernández, and M. André, "LA GAMIFICACIÓN COMO PARTE DE LA ESTRATEGIA DE FORMACIÓN DE ROLES EN EL INGENIERO INFORMÁTICO," Paper presented at the XVII Congreso Internacional de Informática en la Educación, "Inforedu 2018", XVII Convención y Feria Internacional Informática 2018, La Habana, Cuba.

[3] M.Villavicencio and A.Abran, "Sugerencias para la Inclusión de Temas de Medición de software en un Currículo de Ingeniería de Software para Estudiantes de Pregrado," Revista Latinoamericana de Ingeniería de Software, vol. 1, 2015, pp. 117-126.

[4] Y. Rivero, J. Hechavarría and M. Leyva, "Incorporación de las TICs a la enseñanza de la calidad de software sustentado en el método del aprendizaje basado en problemas," Revista Opuntia Brava, vol.9 No.4.,2017.

[5] H. Barrows , "A taxonomy of problem-based learning methods. Medical Education". Springer, New York, 1986, pp. 481-486. DOI:10.1111/j.1365-2923.1986.tb01386.x

[6] N.A. Dalfaro , P.B Demuth , N.F. Aguilar, and C.G. Del Valle, "Hacia la integración efectiva de un modelo por competencias en la carrera de ingeniería de la Facultad Regional de la Universidad Tecnológica Nacional: Entre la formación inicial y la formación docente". Educación en Ingeniería, 13(25), pp. 58-63, Febrero, 2018.

[7] F. Rodriguez, , A. Kolmos , and A. Guerra, (Eds.) . "Aprendizaje basado en problemas en ingeniería:Teoría y práctica ". Aalborg: Aalborg Universitetsforlag,2017.

[8] Y. Rivero and J. Hechavarría, "Aplicación del aprendizaje basado en problemas como estrategia docente en la asignatura de Ingeniería del Software de una universidad cubana", Revista Educación Superior , Año XVI Núm. 24. Julio-Diciembre, Pág. 065-077,2017.

[9] Y. Rivero,"Evaluación de la conformidad de los productos informáticos. Caso de estudio Facultad de Informática y Matemática de la Universidad de Holguín".Director(es): M. Moreno y L.Lamoth . Tesis presentada para obtener el grado de Máster en Matemática Aplicada e Informática para la Administración,Universidad de Holguín, Holguín, Cuba,2015.

[10] M. d. l. A. Rubalcaba, and Y. Zambrana, "Medición de la calidad de Software durante el Proceso de Pruebas en el Proyecto Modernización del CICPC ". Tesis presentada en opción al título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, La Habana, Cuba,2008.

[11] M. d. l. A. Rubalcaba, and Y. Zambrana and H. M. Cruz , "Medición de la calidad de un producto software durante el proceso de pruebas en un proyecto productivo ". Serie Científica de la Universidad de las Ciencias Informáticas 3, La Habana , Cuba, 2010.

[12] F. CENDITEL, Aseguramiento de calidad en el desarrollo de software libre. Venezuela: Fundation CENDITEL,2013.

[13] S. Beovides, and D. Castellanos , "Procedimiento de pruebas no funcionales para software educativo," Paper presented at the XIV Edición de la Semana Tecnológica de FORDES, La Habana, Cuba,2013.