

Methodologies development and software quality metrics in educational applications

JOAN F. CHIPIA LOBO
Maestría en Educación mención
Informática y Diseño Instruccional
Universidad de Los Andes
La Liria, Mérida 5101
VENEZUELA

joanfernando130885@gmail.com

GLORIA MOUSALLI-KAYAT
Departamento de Medición y
Evaluación
Universidad de Los Andes
La Liria, Mérida 5101
VENEZUELA

mousalli@ula.ve

FRANCKLIN RIVAS
Universidad de Los Andes
Laboratorio de Sistemas Inteligentes
La Hechicera, Mérida 5101
VENEZUELA

rivas@ula.ve

Abstract - This paper aims to show some of the aspects related to the need of using software development methodologies and software quality metrics for the generation of educational software applications that meet technical quality standards, plus provide an educational proposal for the solution of problems related to the teaching and learning. Software development methodologies present a structured way for carrying out computational development projects, for making more effective production and achieve high quality in an affordable way; the measurements of this quality is given by assigning a value to a software attribute, either a product or process and are collected as they perform the technical tasks of analysis, coding and testing. Educational programs must follow strict assessment processes for determining the appropriateness of such programs to the educational context.

Keywords: - Educational Software, software quality, software development methodologies

1 Introduction

The software engineering methodology is a structured approach for making more efficient the software applications production, incorporating elements aimed to improve the quality and profitability of such developments. Methodologies such as Structured Analysis and Jackson System Development (JSD) [11], were developed in the 70's, and tried to identify the basic functional components of a system and that function-oriented methods are still widely used. In the 80's and 90's, methodologies were oriented to functions complemented by object-oriented methodologies as proposed by Booch and Rumbaugh. The object-oriented programming paradigm aims to integrate the data, methods and their interactions, implementing features of data abstraction, polymorphism, inheritance; all articulated under the reusability principle [12].

Despite of changes in the way of developing software which has been designed for improving the developments quality and organization, it must be noted that there is no ideal method, and their fields of application may vary according to the areas.

In the educational applications field, it has been used methodologies based on the structured and object-oriented paradigms with educational adaptations.

The study and search for better ways of creating software is linked to the need for some measures on

the obtained products quality. The software measurement refers to a numerical value derived from some software or processing attribute. Comparing these values together with the standards applied in the organization, it is possible to draw conclusions on the software quality of concerning the process for developing it. The software quality metrics do help to better understand the specific attributes of software engineering work products and are collected as they perform the technical tasks of analysis, coding and testing [9].

This work describes the need for using and contexting software development methodologies and software quality metrics in educational environments, which is essential for reliable results when trying to solve an educational problem in an area and / or level of study.

2 Problemized Situation

Nowadays, it is necessary to prepare an individual to be able to get any information and process it consciously. Under this approach, the technological tools introduce innovations in the media and teaching methods that allow the teachers preparation, early vocations feelings and contribute to population cultural development without distinction of race, creed and socio-economic conditions. Therefore the teaching act must improve the quality of the educational process related to science and technology and thus change the typical traditional view, which is very limited,

achieving a major breakthrough to obtain a user able to address the various phenomena that occur at present.

The classroom work presents problems to be solved appropriately, then the teacher should handle innovative methods such as the use of educational software for the knowledge dissemination; however it arises the following question: how to determine the software to be used according to the needs of the moment?.

Additionally, there is the concern of the educational software reliability that has been selected according to usability, accessibility, efficiency, maintainability criteria, among others. How a classroom teacher or a team of classroom teachers can differentiate quality educational software, beyond the relevance for the development or a particular theme support. This is common today, due to the proliferation of information technology developments for teaching [14], due to the rise and support that have given by governments and organizations to the use of Information and Communication Technology (ICT) in education.

The creation of educational applications without proper measurements quality verification or certification are coupled with the lack of progress on specific methodologies for the development of such applications [13], because the implied stages in the software creating process should be considered in context and activities relating to educational problems; they can not be treated the same way as the computer solutions for the administrative or industry area, since the principles governing teaching situations are linked to people learning theories and are not the product of a structured sequence of steps that can be modeled in a unique way.

3 Prospects for the solution

Due to the characteristics of educational developments and taking into account the pedagogical and communication with the user, in each particular case, the answer to the problem must be based on an adaptation of existing development software paradigms to the educational theories that satisfy a particular demand, thus it will be presented a general outline of the software development methodologies.

3.1 Software development methodologies

The methodology is a set of philosophies, phases, procedures, rules, techniques, tools, documentation

and training issues for information systems developers [9].

The development methodology is a set of procedures, techniques, tools, and supporting documentation to help developers in making new software [8]. Summarizing the above, the author says that a methodology "represents a way to develop software in a systematic way." The methodologies pursue three main needs: best applications, leading to better quality, a controlled development process, a standard process in an organization, not dependent on the people, through the generation of adequate documentation of each process or stage, which constitute the mechanism of interaction between members of the team. The processes are divided to the level of elementary tasks or activities where each task is identified by a procedure that defines how to implement it. For applying a procedure it can be used one or more techniques, which may include graphic with text [9].

3.2.1 Characteristics of software development methodologies

It can be listed a number of features that should be present in the methodology and that could influence the development environment:

- Predefined rules
- Identification of life cycle steps
- Checks at each stage
- Planning and control
- Effective communication between developers and users.
- Flexibility: application in a wide range of cases
- Easy to understand
- Automated tools support.
- It can be defined measurements that indicate improvements
- Allow changes
- Support reusability of the software [8].

Early computer developments applied to the field of education were based on elementary steps: analysis, design, development and testing; the changes required to improve the quality of the products has led to the incorporation of concepts and models created by software engineering. That is the case of the Unified Modeling Language (UML) that have been used as a fundamental tool from the earliest stages of educational application design, in order to link users and developers from

what can be expected of a particular application to test cases for validating the development.

It is necessary to incorporate new elements in the analysis and design stages when making educational computer developments, since the user to which the product is directed has unique characteristics from the point of view of learning, this must be identified and considered at subsequent design stages. Therefore it is necessary to include an analysis section of instructional requirements for end users, similarly, these requirements should be considered in the educational design at the following stages of software development methodologies specified by software engineering.

3.2. Software Quality

The ISO 9126 Norm, defines quality as: "Total product or service characteristics that are involved on its ability for satisfying stated or implied needs". These specified requirements, may be in a contract or have to be explicitly defined [3].

Achieving quality may have three origins: made quality is the one that is able to get the person doing the work, scheduled quality which is the desired to obtain and necessary quality is the one required by the client and the one he would like to receive [8].

From the perspective of software engineering, quality is referred to the product with special features, considering it as a product being developed and is focuses on its design with instructions existing logic for support. This product is not spent for its use as others and repairing it, is not restoring to the original state, but to correct a manufacturing defect which means that the product delivered has flaws that can be resolved in the maintenance stage [8].

Pressman defines the software quality as "software agreement with the explicitly stated requirements with development standards specifically set and implicit requirements, no formally set wanted by the users" [9].

When we speak of educational software quality, it is logical to think that we mean the ability of the software to promote the user (student) cognitive and metacognitive processes that lead to enhance learning; from this perspective there have been developed quality checks with qualitative trends [14], however, educational software quality can not be separated from the technical issues described in ISO Norm, is therefore that the processes of creation and evaluation of educational software

must consider both aspects in a comprehensive manner .

3.2.1 Software technical quality Metrics

For the quality evaluation is more commonly referred as "product measures" that in "process measures". A metric is "assigning a value to an attribute of a software entity, whether a product or a process." In all cases, the metrics represent indirect measures of quality, since they only measure the manifestations of it. It can be used metrics based on the text of the code and metrics based on the structure of code control [9].

3.2.1 Metrics based on the text of the code

In general, it can be taken the code lines number, as an indicator of size, comments lines number as an indicator of internal documentation, the number of instructions, code lines percentage or documentation density, among other [9].

3.2.2 Metrics based on the structure of code control

It can be taken two types of measures: one related to the intramodular control, based on the control graph and other related to the modular architecture, based on the call graph or structures chart [2]. Piattini holds that better results suggest that metric values mean less maintenance later due to fewer defects [8].

3.3 Software verification and validation

Software verification and validation include a set of procedures, activities, techniques and tools used in parallel to software development, for ensuring that the product solves the problem that was designed for. It acts on the intermediate products trying to detect and correct early defects and deviations from the target [9].

These procedures let taking corrective actions at intermediate stages, in order to continue adapting, by making the necessary changes, for a better quality product. This verification and validation must be provided in development methodology. They are generally carried out by experts in the knowledge areas that are covered by the software, expert teachers and experts in human-machine communication, are used for reviewing the usability criteria, which can become a risk factor because the design can be adapted from the instructional point of view, but its usability can be low, so learning is limited. Although, some authors

argue that they are independent variables [15], it can be considered that the technical aspects can become risk factors for achieving the instructional objectives listed in educational software.

3.3 Educational software evaluation

Educational programs evaluation is a process for determining the adequacy of these programs to the educational context. When the program reaches the teacher, it is supposed that it has been tested and evaluated both its pedagogical and didactic skills and the technical area that correspond to the developed product quality according to certain quality assurance guidelines [7].

Basically, there are performed internal and external evaluations of the software, for detecting, as soon as possible, the problems that generate changes in the product, in order to reduce costs and effort later. These evaluations consider any changes suggested by the development team and final users, taking into account both teachers and students in the learning context [7].

When an education commercial product is reached by teachers, it has passed the internal and external evaluation stages. Also, for obtaining the effectiveness and efficiency of the product it must be evaluated in the context of use [7].

It's important to define certain criteria or indicators for selecting a program according to the teacher's needs, and should be considered the assessment and evaluation for determining whether a given program meets both, technical and pedagogical and didactic objectives [7].

3.3.1 Software evaluation tools

In general, the most used instruments are the assessment questionnaires, where the answers to these questionnaires are valued between 0 and 5, for example: the user's compliance level with the claims proposals [7].

The forms assessment instruments should be made including closed, open and check boxes questions, allowing the final user having a description of areas and individuals problems in the program that have not been taken into account during the instrument preparation. It must be taken into account when drafting the questionnaire the use of appropriate vocabulary, unambiguous and clear for the intended recipients in each particular case [7].

In most of the questionnaires there are considered key or outstanding aspects, such as the achievement of the objectives, technical issues, content development, activities and documentation. These

issues are categorized in sections in each proposal [7].

As each software evaluation proposal is unique, it must be carefully examined the various proposals for teaching aids and educational software evaluation in particular, having them only as a "guide" which then must be "restructured" to each particular educational context [7].

3.3.2 Educational Software selection and evaluation Proposals

In recent decades, many proposals have been drawn up with criteria lists for selecting and evaluating educational software, some individually and others at institutional level. They vary in content and style, but all have a common goal which is to help the teacher to choose and evaluate a program suited to them needs.

Regarding the evaluation plan proposals, it can be found the evaluation form for computer contents MicroSIFT (Microcomputer Software Information For Teachers) [5] and the Northwest Regional Educational Laboratory, Oregon in the U.S. (OCDE) [6]. OCDE was the first institution devoted to information about educational software. It was designed an instrument for serving as the basis for the evaluation process.

Interestingly, the approach according the doubts about the validity of the checklists submitted by Johnston, says that it can not be predicted how a particular software will be used in the classroom, because it depends on the teaching approach, type of curriculum, strategy used and final users. Often teachers and students can explore technology applications that could be unnoticed to the designer. This is perhaps one of the most significant findings, because it takes into account the multidimensionality of the involved processes and the different styles of actors in the processes, as well as institutional aspects when considering the type of curriculum [4].

Cabero argues that the proposals for the software evaluation are very varied [1]. Marques, from the Autonomous University of Barcelona offer a tab for documentation and evaluation of educational programs; warning that in evaluating a program, it's important to consider its characteristics and its relevance to the context in which it's desired to use. Consider contextual assessment of the programs and the way it has been used in a particular application class regardless of their technical and educational quality. This assessment takes into account the achievement degree of educational

objectives from the planned. He insists that the methodology used by the teacher, is the main factor for the success of educational intervention. It must therefore take into account the previous reasons made by the teacher before the session, the distribution of learners in class, autonomy for interacting with the program. It plays an important role the student characteristics, the motivation degree, cognitive styles, interests, previous knowledge and skills [7].

Special mention should be made concerning educational software distributed evaluation proposed by Alonso Sanchez of the University of Chile [10]. He presents a proposal for improving and enriching the educational software evaluation using a questionnaire that considers three key aspects of the production of educational software: the user interface design, educational computing and visual communication. They present an innovative way to access using Internet in order to make simpler the process of educational software evaluation. They note that some authors consider that it should respond to a curricular model (for example, it doesn't have the same purpose a learning reinforce program that a collaborative learning support program) focused on the teacher or the student and that is constructivist or behavioral. Several authors feel that all software should be considered in its construction a formative assessment and summative process [10].

4 Concluding remarks

The incorporation of educational software in the classroom allows the knowledge integration and training-type activities, hence it is essential the student-teacher interaction that should stimulate the creative potential of students for using their initiative within a classroom environment that facilitates decision-making autonomy and control.

This is not an easy task, however, the use of various strategies will help facilitating the teaching and learning of students and the greatest reward is the construction of meaningful knowledge for themselves which are the focus of the important teaching work.

It is essential, when developing software to take into account the needs in each case, educational theory and / or curriculum for adapting some of the methodological paradigms, discriminating at each stage the activities to do with the documentation, techniques and tools to be used. The selected methodology will direct the activities taking place

in each of the stages; likewise will be a determining factor in operational decisions making as the development languages selection, people involved in the project, estimated time and cost.

Analyzing needs for the software methodology used is similar to the complex problems faced by teachers in the classroom every day. In them, the activities success depends on its ability to handle uncertainty and solve practical problems. The skill required is knowledge and technology intelligent and creative integration. This skill or practical knowledge is an action reflection process or a reflective conversation with the specific problem situation. This reflection involves the man's submersion in the world of experience, connotations, values, symbolic exchanges, correspondence emotional, social interests and political scenarios. In the reflecting part, the professional remake his practical world and reacts restructuring some of its action strategies through innovative methods different from the traditional ones.

For this reason, it can be taught using independent practice based on educational software when there are some drawbacks, such as the teacher had no opportunity to explain some content for the appropriate understanding, because there was not enough time during the school year or semester. For these reasons, software is extremely important for teaching and is presented as a meaningful tool for supporting teachers using technology tools.

Finally, it should be noted that the long criteria lists developed for evaluating educational programs, are relative data when using educational resources. The teaching role drives the use of the programs, being the proposals creativity and originality the key aspects that increase the media value. Similarly, it must be pointed the need to aware the incorporation of educational software as a tool within the classroom without proper review of its quality and relevance, because it may hinder the process of student learning. This issue should not be considered lightly, but must be the result of a careful process where it's justified the selection of an application program based on a methodology for ensuring the generated product quality.

References

- [1] Cabero. Para una Tecnología Educativa. Horsori, Barcelona, 1993.

- [2] Ian Sommerville .Ingeniería de Software. PEARSON EDUCACIÓN, S.A., Madrid, 2005.
- [3] ISO. Information Technology Software Quality Evaluation Characteristics. ISO 9126. Ginebra, Suiza, 1991.
- [4] Johnston. The evaluation of Microcomputer Programas: An area of debate, Journal of Computer Assisted Learning, Vol. 3, No. 1, 1987, pp. 40-50.
- [5]MicroSIFT. Evaluation guide for Microcomputer–Based Instructional Packages. Microcomputer Software Information for Teachers (MicroSIFT). 1982
- [6] OCDE. Information Technologies in Education: The Quest for Quality Software. París, Organisation for the Economic Cooperation and Development, 1989.
- [7] Peré Marqués. Metodología para la elaboración de software educativo en Software Educativo. Guía de uso y metodología de diseño. Estel, Barcelona 1995.
- [8] Piattini. Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. Rama. Madrid, 1996.
- [9] Roger Pressman. Ingeniería del Software. Mc Graw Gill, Madrid, 2005.
- [10] Sánchez y Alonso. Evaluación distribuida de software educativo. 1997. Documento Web, disponible en: <http://www.dcc.uchile.cl/~oalonso/educacion/>
- [11] Cameron John R. The Jackson Approach to Software Development , IEEE Computer Society Press, Silver Spring. 1989
- [12] Schach, Stephen. Object-Oriented and Classical Software Engineering, Seventh Edition. Mc Graw Gill. 2006
- [13] Cristancho José A. Evaluación de la calidad del software Educativo bajo el estándar ISO 9126. Evaluación e investigación Vol 1 N° 2. 2006
- [14] Pérez M., Pérez N., Méndez S., García A., Gálvez M., Quincoces V., Liberatori H., Fiorito B. y Lasserre C. Construcción y Validación del cuestionario: Métrica de Calidad de Credibilidad e Interacción de Cursos de Teleformación. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología. 2007
- [15] Squires D. and Preece J., Usability and learning: evaluating the potential of educational software Computers and Education. (1996).