International Journal of Computer Science and Software Engineering (IJCSSE), Volume 7, Issue 12, December 2018
ISSN (Online): 2409-4285 www.IJCSSE.org Page: 282-288

Analysis of Knowledge Evolution on Software Quality Concepts

Rogério Rossi

Department of Computer Engineering, University of São Paulo, São Paulo, São Paulo, Brazil

rogeriorossi8@gmail.com

ABSTRACT

The analysis of knowledge evolution of a student's group based on Software Quality concepts is the main objective of this research. The course was prepared using appropriate instructional design model and cognitive strategies. The research respondents consisted of 28 students of a Graduate Software Engineering Program coming from different undergraduate courses. The method used for the research development is based on data collection using a structured questionnaire with questions about software quality concepts as well as software quality frameworks and management practices. The questionnaire was given to the respondents at three different times: the first one was in the first class, the second on the fourth class and the third one on the seventh class. With these data collection, a diagnostic evaluation could be done at the first time and formative evaluation was done in the next ones. These evaluations favored the understanding of knowledge acquisition and its evolution, enabling instructional adjustments towards better learning.

Keywords: *Cognitive Strategies, Instructional Design, Knowledge Acquisition, Software Quality*.

1. INTRODUCTION

Quality is an extremely required characteristic by contemporary society, for products and services of many sectors. The same occurs with the software product that has to meet requirements regarding quality criteria, especially due to its large-scale of use. Thus, the concepts about software quality are treated in disciplines with different names in many Software Engineering Programs at any academic level.

Knowledge on these concepts can be acquired in a diversified way, be it from specific courses using specific frameworks, from specific training sessions, from courses offered in undergraduate or graduate programs, or in a practical way in the software development industry.

However, formal learning process in an undergraduate program can support Software Quality concepts acquisition and evolution. According to [1], one of the key objectives of formal education is the transmission of knowledge. Thus, the learning process should be performed to transform existing knowledge into new knowledge, according to the information considered in this process.

For [2], the learning process addresses the processing of information that occurs from the learner environment, being information received, recorded and stored in short and long-term memories. The model of learning and memory process presented by the authors correspond to modern cognitive theories and favors the transmission of information towards knowledge formation.

As mentioned by [3, p. 4] "cognition means 'coming to know'"; in this sense, cognitive strategies should be considered to treat instruction and to generate knowledge. For the authors, cognitive strategies should be used in the stages of courses (instruction) definition and construction, so that, together, they can favor knowledge acquisition.

The learning approach called 'cognitive constructivism' focused on how knowledge is constructed and accordingly to [4] contributes to the studies about how instruction under formal learning mechanisms can transform knowledge.

Hence, this research is justified by the need to transmit and to modify knowledge about software quality concepts, since for [5] software quality is considered one of the five most important concepts by students of software engineering undergraduate courses.

On this sense, this research was conducted with students of a Software Engineering Program, specifically during its Software Quality Management course and has the objective to check the knowledge acquisition and evolution associated to software quality concepts. The study of knowledge engineering associated to the investigation on how the learning process can interfere and modify knowledge derives from this main objective. It's also considered how instructional design and cognitive theories can improve the learning process when it is applied in a course development.

To achieve the aforementioned objectives, the paper is organized as follows: section two presents a theoretical literature review on the study of education and knowledge, presenting some cognitive strategies related to the learning process through instructional design theories; section three presents the scenario in which the research was conducted, as well as the methodology adopted for its completion; section four presents the analysis and discussion; and the fifth section presents the conclusions and future works related to this research area.

2. LITERATURE REVIEW

2.1 Education, Learning, and Knowledge

One of the functions of education in formal learning environments is to transmit information. When this occurs, the learner goes from a state of less knowledge to a state of more knowledge. In this sense, accordingly to [1], the cognitive approaches have been very influential in the educational field as they have been transforming the learning process.
For [6], the learning processes that take place according to different cognitive strategies generate or transform knowledge that culminates in the 'society of knowledge and learning'. For [7], the current society is considered the 'knowledge society', in which education becomes its center, having the school as the key institution. So, it is possible to verify the relevance of the learning process considering formal mechanisms of education for knowledge construction. Based on cognitive sciences, cognitive constructivism is an approach that aims at building knowledge, as stated in [8].
According to [4], before 1990, cognitive processing considered the idea that knowledge was external to the learner. Modern cognitive theories from the 1990s refocused this vision considering that knowledge occurs internally, as part of the mental processes.
Reference [2] allows verifying a model of information processing related to the learning process. This model considers the (short and long-term) memory as it is guided in modern cognitive theories, which consider eight steps for the internal learning process: 1) reception of stimuli by recipients, 2) information registration, 3) selective perception for short-memory, 4) semantic encoding for long-term memory, 5) retrieval working memory, 6) response generators to effectors, 7) performance in the learner's environment and 8) control of process.
This model and its processes favor understanding the cognitive mechanisms capable of supporting the learning process that should transform the received information through stimuli in knowledge, when associated to prior information.

2.2 Instructional Design and Cognitive Strategies

Theories focused on the instructional design associated to cognitive processes related to learning are able to support instruction transmission. [2], [9] proposed specific models that address the instructional systems to support the learning process, as is also possible to see an instructional design model in [10]. For [2], any instructional system model can be categorized into one of three functions: 1) identifying the outcomes of instruction; 2) developing instruction; and 3) evaluating the effectiveness of instruction.
Reference [9] proposed an instructional system model based on ten steps that are able to support the design of instructional systems from the definition of instructional purposes to the final stages of evaluation, either formative or summative, considering steps that describe performance goals and addressing the selection of instructional materials.
The approaches about instructional design theories identified in [2], [9], and [11] favor the pedagogical engineering related to the construction of courses for information transmission and knowledge construction.
The knowledge and the involvement of cognitive science with instructional design theories facilitate the dissemination of education through a formal course, as verified by different cognitive strategies, such as [3]. The authors present a set of cognitive strategies that can be associated to instructional design, being able to facilitate the design and transmission of instruction considering the group of receivers (learners).
In [3], four families facing the cognitive strategies are verified: 1) Chunking (that considers only the Chunking or Organizing Strategy) , 2) Spatial (considering Frames Type One, Frames Type Two and Concept Maps, 3) Bridging (that considers Advance Organizer, Metaphor, Analogy, Simile and Frames Type Two; and finally), and 4) Multipurpose (considering Rehearsal, Imagery and Mnemonic Strategies).

3. SCENARIO AND METHODOLOGY

3.1 Course Design

The 'Software Quality Management' course was defined using the instructional design theories, as well the model presented by [9]. It also considers cognitive strategies to favor the achievement of instructional objectives. The course was prepared regarding specific software quality concepts and major frameworks, such as models and standards, related to software quality.
The ten steps defined by the Instructional Design Model presented by [9] are: 1) Identify instructional goals; 2) Conduct instructional analysis, 3) Analyze learners and contents, 4) Write performance objectives, 5) Develop assessment instruments, 6) Develop instructional strategy, 7) Develop and select instructional materials, 8) Design and conduct formative evaluation, 9) Design and conduct summative evaluation, 10) Revise instruction. As the model steps are not linear, its representation is presented on Fig. 1. These steps were used for course construction, as well it was considered in the course preparation, as briefly mentioned below.

The instructional objectives were defined according to some specific variables, such as: students' academic background; relevance of the course to the Graduate Program and other correlated courses; updated scenario concerning the content provided.

Performance objectives were derived from the instructional objectives and were comprehensively described, considering the students' group of instruction, to determine the variations in performance through evaluation mechanisms, and to control whether performance objectives were met.

The instructional strategy was defined according to the course content. Some specific software quality frameworks presentation was planned to be done during the course. It was necessary to present some preliminary concepts that favored the understanding of related software quality frameworks. Therefore, the instructional strategy was based on the presentation of preliminary concepts before the presentation of frameworks and specific case studies.

Cognitive strategies, according to [3] should be used according to the students' needs, therefore, for this specific course, some cognitive strategies were considered, such as analogy, metaphor and conceptual maps. Cognitive strategies were also used for visualizing the consistent use of processes defined in the frameworks that were presented.

The development and selection of instructional material were performed based on the goals and instructional strategies as previously defined. The material was developed specifically considering the concepts necessary for the course; selected and current references were defined, too. The instructional material concerned the quality frameworks defined in the instructional strategy was presented to all the participants.

The definition and implementation of formative evaluation favored the instruction control and performance analysis that had previously been defined. In this sense, the formative evaluation was conducted for instrumenting tests, texts and oral presentations, which favored adjustments and instructional reviews.

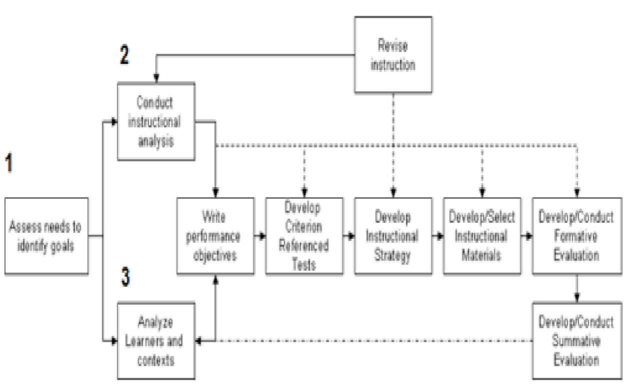

Fig. 1 Instructional Design Model [9].

The course design was completed with the definition of the summative evaluation phase considering all the instructional objectives and how to verify the transformation of knowledge according to the concepts, specific frameworks and case studies presented along the course.

3.2 Data Collection

The Software Quality Management course was presented to students from different undergraduate programs, such as Computer Science, Information Systems, Production Engineering, Business Administration, among others; most students had experience in software development, either in technical or managerial activities. They sometimes had different backgrounds, and some had not participated in formal courses in academic environment for over five years.

Thus, the data collection was given just with by the group of participating students enrolled in the course. During the survey, some aspects of formative and summative evaluations could also be treated. A total of 28 respondents answered the survey.

Data collection was based on a questionnaire developed during the course preparation. The questions were organized based on the lessons to be taught. The set of questions adhered to different references related to software quality, such as [12], [13].

Data collection was performed at three different times:

The first time data collection was occurred in the first of nine meetings scheduled for the completion of the course. A questionnaire (see Table 1) with ten questions was distributed to be answered without personal identification;

The second time data collection was in the fourth meeting, i.e., the same questionnaire (Table 1) distributed in the first time was again presented. When the respondents answered the questionnaire for the second time, they did not have access to the answers given in the first time, since the first questionnaire was not available to them;

Table 1: Questionnaire on Software Quality Concepts

Question #	Question Description
Q1	Do all clients (users) perceive the software quality in the same way?
Q2	Is Software quality likely to be built?
Q3	Do CMMI, ISO specific standards such as ISO 12207 and ISO 15504, and the MPSBR Model refer to models and standards that support the software quality management?
Q4	Is the same quality model capable to assess both the quality of product and the quality of software process used to develop it?
Q5	Should models, standards or guidelines be used at the end of the software development process for quality control?
Q6	Can the software process development, in general, influence the qualitative aspects of the software product?

Q7	Can the quality of the software product can be measured throughout its development process?
Q8	In a software development organization, should the Quality Management System be restricted to the managers and specialists in software product development?
Q9	Can the quality of software components (interface, code, etc.) be measured according to specific standards for each of these components?
Q10	Does software testing reflect the main point of software quality control?

The third time data collection was in the same way as the second, the same questionnaire with the same questions was answered, but the students had no access to what they had answered the first two times. This occurred in the seventh meeting.

These three times of data collection were also relevant to the evaluation process. First time data collection helps the group diagnosis and the next two times allowed verifying the answers which at that time corresponded to formative evaluations focusing on the instructional content presented. The formative evaluation was possible given that the questionnaire was previously prepared to contain issues related to the concepts and case studies presented along the course.

In the first time data collection, the respondents answered the questionnaire with positive impact, since it was not a practice in the Program do these kind of diagnosis evaluation. This first data collection favored the previous analysis of students' knowledge and some course adjustments.

Although some of the questions could pose a high degree of difficulty for some students, most respondents had some notion about all questions and could easily answer the questionnaire. Other respondents, with greater difficulty in answering the questions, presented their responses according to their experience or prior knowledge obtained in their undergraduate programs or in their professional life.

The second time data collection also had a positive impact, but some respondents commented on the similarity with the content of the questionnaire presented at first time data collection since they did not have access to the questionnaire used during the first time data collection.

The questions were not detailed or explained while the data collection was held, requiring the respondents to answer the questions according to their own knowledge. The questions were fully related with the instructional content of the course, increasing the possibility of better results in the third time of data collection.

The third time came more naturally because the respondents perceived the purpose of the research, although the questions had not been detailed or broadly discussed again. At that time, most students linked the questions with the instructional contents previously presented over the course.

Thus, the data collection was performed for future analysis of results related to the evolution of knowledge about the instructional content of software quality presented throughout the course.

3.3 Research Methodology

The research methodology was conducted using a questionnaire with ten questions as presented in Table 1. For [14], one of the advantages of using structured questionnaires for data collection in scientific research is to obtain fast and accurate answers. The questions were designed according to the content of the course concerned with Software Quality Management. Each question had multiple choice alternatives as presented in Table 2.

All the questions should be answered by selecting one of the possible alternatives according to the Likert scale [15]. The possible answers to be selected for the ten questions are presented in Table 2.

Thus, the research data collection was guided by a questionnaire that was provided in printed form to be answered by considering only one of the five alternatives presented as the correct alternative.

4. RESULTS AND DISCUSSION

The knowledge evolution analysis was verified in accordance with the transformation of knowledge presented by the group through the questionnaires. As the survey took place with a group of graduate students, the interviews allowed observing the group background related to software engineering and, more specifically, to software quality concepts.

The questionnaire (see Table 1) was presented at three different times with the same set of questions and the same possibility of answers. The results for each question at each time of data collection can be verified in Table 3 as well its considerations are presented below.

For the first question, which considers that each user perceives software quality differently, there is a significant variation from the first time, in which only nine of the respondents (32.14%) considered the SD (Strongly Disagree) response, reaching fifteen respondents (53.57%) in the second time and keeping that way in the third time;

For the second question, which considers the construction of quality, there was no variation between the first and the third times considering the sum of SA and PA responses, in which 96.4% of respondents considered responses between SA and PA for the first and third times. This reflects an important result because, according to [16], quality should be built along with the product construction;

The third question addresses the relationship of quality frameworks (models, standards or guides) and their ability to interfere with quality management. There was a

continuous increase considering the three collections, doing the sum of SA and PA responses with the variation from the first time to the third time, which was 14%;
The fourth question concerned the ability of quality models to measure the quality of the process and product. Indeed, part of the frameworks presented corresponds to quality models guided by software process improvement, but these support the implementation of institutional processes focusing on improving the quality of the product. There was little variation among the three times if considering the sum of SA and PA or PD and SD;

Table 2: Questionnaire Answers based on Likert Scale

Answer Code	Answer Description
SA	Strongly agree
PA	Partially agree
NN	Neither agree nor disagree
PD	Partially disagree
SD	Strongly disagree

The fifth question regarded the use of models and quality standards only for quality control and, more precisely, at the end of the software development. This question presented a positive variation of 21% when considering the sum of PD and SD from the first to the second time and an 11% negative variation from the second to the third time;
The sixth question deals with the relationship between the development processes and the software product itself; in this sense, low variation is found by considering the sum of SA and PA for the three times, but when only SA is observed, from the first to the third time, there was a significant variance of 15%. This is due to the interference of the process engineering approach with the software quality product discussed during the classes, and also to the fact that quality models regard the continuous process improvement approach. During the course, some SPI (Software Process Improvement) models were presented such as CMMI (Capability Maturity Model Integration) [12], ISO 15504 [13];
For the seventh question, high rates geared to the sum of SA and PA responses were observed, with low variation from the first time to the second and without variations from the second to the third time. The seventh question considers the possibility to measure the quality of software product throughout its development process;
The eighth question mentions that the quality management system is restricted to managers and specialists in software development. This relationship should not be true given that the entire organization, including users, should participate in the quality management system, even specifically for software development. In this sense, there is a decrease of 14% from the first to the third time. If verified the sum of PD and SD, no variation is observed between the first and the third times, maintaining the high level of disagreement related to this issue;

Table 3: DataCollection Results

	Data Collection – First time					Data Collection – Second time					Data Collection – Third time				
	SA	PA	NN	PD	SD	SA	PA	NN	PD	SD	SA	PA	NN	PD	SD
Q1	2 7.14%	9 32.14%	1 3.57%	7 25%	9 32.14%	1 3.57%	6 21.42%	0	6 21.42%	15 53.57%	2 7.14%	6 21.42%	2 7.14%	3 10.72%	15 53.57%
Q2	14 50%	13 46.42%	1 3.57%	0	0	13 46.42%	12 42.86%	1 3.57%	2 7.14%	0	13 46.42%	14 50%	1 3.57%	0	0
Q3	12 42.86%	9 32.14%	6 21.42%	1 3.57%	0	16 57.14%	7 25%	4 14.29%	0	1 3.57%	21 75%	4 14.29%	0	3 10.72%	0
Q4	5 17.86%	10 35.71%	2 7.14%	8 28.57%	3 10.72%	6 21.42%	13 46.42%	1 3.57%	3 10.72%	5 17.86%	7 25%	7 25%	4 14.29%	6 21.42%	4 14.29%
Q5	4 14.29%	5 17.86%	4 14.29%	2 7.14%	13 46.42%	2 7.14%	2 7.14%	3 10.72%	2 7.14%	19 67.86%	3 10.72%	4 14.29%	3 10.72%	6 21.42%	12 42.86%
Q6	20 71.43%	7 25%	0	1 3.57%	0	17 60.71%	11 39.29%	0	0	0	24 85.71%	4 14.29%	0	0	0
Q7	21 75%	3 10.72%	4 14.29%	0	0	19 67.86%	7 25%	1 3.57%	1 3.57%	0	22 78.57%	4 14.29%	0	2 7.14%	0
Q8	0	4 14.29%	0	5 17.86%	19 67.86%	1 3.57%	3 10.72%	1 3.57%	7 25%	16 57.14%	0	3 10.72%	1 3.57%	9 32.14%	15 53.57%
Q9	15 53.57%	8 28.57%	3 10.72%	2 7.14%	0	13 46.42%	8 28.57%	2 7.14%	4 14.29%	1 3.57%	11 39.29%	12 42.86%	2 7.14%	0	3 10.72%
Q10	5 17.86%	17 60.71%	0	5 17.86%	1 3.57%	4 14.29%	11 39.29%	0	7 25%	6 21.42%	2 7.14%	12 42.86%	0	6 21.42%	8 28.57%

The ninth question considers the specific and generic quality standards for software components (interfaces, codes, etc.). No variation was verified considering the first time and the third time data collection according to the sum of SA and PA, thus maintaining 82.1% at both times;

The tenth question is related to the testing process, and considers that the test phase is the main point of software quality. In this case, there is a significant negative variation if the first and third times and the sum of SA and PA are considered, with a variation of 29% in the third time, when respondents considerably lowered their agreement with this question. Similarly, the sum of PD and SD for the first and third times was observed to reveal a significant increase of 29% indicating that respondents disagree that the test phase reflects the main point of software quality.

The results allow comprehensively assessing the group knowledge transformation regarding the concepts and frameworks presented, whereas the individual variation of knowledge was impossible to obtain given that, during data collection, the respondent identification was not required.

It should be emphasized that learning can occur in specific formal environments, in social environments; as well it is possible to consider that there is self-learning. In this sense, the deviations of the results of this survey were not

calculated because there was no restriction for the learning environment, since the students could learn in their work environment or by some self study. However, this variance should be taken into account as it is feasible, especially due to the fact that the survey took place in a six-week interval, the first data collection occurring in the first week (first class) and the last in the seventh week (seventh class), which implies that knowledge may have been modified not only by the formal learning process but also according to external learning or self-learning. So, it is possible to consider that knowledge undergoes continual transformation by using the learning process.

The results variance is observed because of the instructional content prepared for each class, since the students did not receive all instructions in just one class for answering the questionnaire. As well, the integration between concepts and some software quality frameworks were presented class by class during the course, using specific case studies to favor the learning process. This course plan can also provoke some results variation since the instruction reception did not coincide with data collections (questionnaires).

5. CONCLUSIONS

The course was prepared according to a specific instructional design model and appropriate cognitive strategies to some learning objectives. Over the meetings, a detailed overview of the key software quality concepts and practices was presented to the group. Associated to these concepts, a set of frameworks (models, standards, guides, etc.) were used to consolidate some software quality concepts. Several case studies were presented and discussed in depth, promoting knowledge improvement.

From the data collection and tabulation, the analysis evidenced that the knowledge evolution by the group was positive, given that 90% of the questions had answers that corresponded to the relevant assimilation of the content presented. Satisfactory modification of knowledge about the subject matter presented could be observed.

Each of the questions was thoroughly checked in order to analyze the knowledge evolution regarding the issues addressed during the course that correspond to one or a set of questions of the questionnaire given to the respondents.

A significant change was verified considering the first question. At first time data collection, only 9 respondents (32.14%) considered a SD (Strongly Disagree) response, but fifteen respondents (53.57%) at the second and third times. The changes took place due to the instructional content presented in the initial class, in which specific instructions and readings to discuss specific concepts modified the vision of quality and quality construction.

Future researches may be carried out considering other courses in the Software Engineering Program; as also, it can be made in other programs favoring the formative and summative evaluations, and especially allowing knowledge assessment of specific concepts in a particular knowledge area.

REFERENCES

[1] M. Carretero. "Cognición y Educación" [Cognition and Education], in: Desarrollo cognitivo y educacíon [I]: Los inícios del conocimiento [Cognitive development and education [I]: starting knowledge], J. A. Castorina and M. Carretero, Ed. Buenos Aires: Paidós Press, 2012, ch. 3, pp. 87-112.

[2] R. M. Gagné, L. J. Briggs, and W. W. Wager, Priciples of instructional design. Orlando, FL: Harcourt Brace Jovanovich College Publichers, 1992.

[3] C. K. West, J. A. Farmer and P. M. Wolff, Instructional design: Implications from Cognitive Science. Boston, MA: Pearson Custom Publishing, 1991.

[4] G. R. Saadé, F. Nebebe and W. Tan. (2011). Knowledge management systems development: theory and practice. Interdisciplinary Journal of Information, Knowledge and Management. [Online]. 6, pp. 36-72. Available: http://www.ijikm.org/Volume6/IJIKMv6p035-072Saade472.pdf.

[5] K. McMaster, S. Sambasivam, S. Hadfield and S. Wolthuis. (2012). Software engineering frameworks: perceptions of second-semester students. Issues in Informing Science and Information Technology. [Online], 9, pp. 317-329. Available: http://iisit.org/Vol9/IISITv9p317-329McMaster022.pdf.

[6] J. Delors, Learning: the treasure within. Paris, UNESCO, 1996.

[7] P. F, Drucker, The age of social transformation. The Atlantic Monthly, November, 1994.

[8] P. Cronin. (1997). Learning and assessment of instruction. Unpublished report, Edinburgh: University of Edinburgh Centre for Cognitive Science. [Online]. Available: http://www.cogsci.ed.ac.uk/~paulus/work/vranded/litconsa.txt.

[9] W. Dick, L. Carey and J. O. Carey, The systematic design of instruction. Boston, MA: Allyn & Bacon, 2005.

[10] N. Mehddeb. “A new course on management education introduced by a new instructional design model”. in International Journal of Information and Education Technology, 2015, 5(4), pp. 278-286.

[11] P. L. Smith and T. J. Ragan, Instructional Design. New Jersey: Prentice Hall Inc., 1999.

[12] M. B. Chrissis, M. Konrad, and S. Shrum, CMMI: Guidelines for process integration and product improvement. Boston, MA: Pearson Education, 2004.

[13] E. K. Emam, J. Drouin, and W. Melo, SPICE the theory and practice of Software Process Improvement and Capability Determination. Piscataway, NJ, USA: IEEE Computer Society, 1998.

[14] M. A. Marconi and E. M. Lakatos, Fundamentos da Metodologia Científica [Fundamentals of Scientific Methodology]. São Paulo: Atlas, 2010.

[15] R. Likert, “A technique for the measurement of attitudes” in Archives of Psychology, 1932, pp. 1-55.

[16] D. A. Garvin, Managing quality: the strategic and competitive vision. Rio de Janeiro, Brazil: Qualitymark Ed., 1992.