Applying a Contest to Improve Learning in the Information Systems Development – An Interdisciplinary and Extracurricular Approach

Marcelo FANTINATO, Marcos Lordello CHAIM,
Marcelo MORANDINI, Sarajane Marques PERES,
Esteban Fernandez TUESTA

School of Arts, Sciences and Humanities, University of São Paulo
Rua Arlindo Béttio, 1000-Ermelino Matarazzo, CEP: 03828-000, São Paulo, Brazil
e-mail: {m.fantinato, chaim, m.morandini, sarajane, tuesta}@usp.br

Abstract. Contests are usually applied in the academic environment to simulate real professional situations that require from the participants a more pro-active attitude than the one shown in conventional coursework. Although they are commonly applied in the scope of a unique course, the contest described here was an extracurricular experience applied in an Information System undergraduate program. The evaluation of the contest is also presented; the objective was to assess the role of the contest as a tool to bring together interdisciplinary subjects, complementary to the traditional disciplinary structure of the program curriculum. The results indicate that a significant portion of the participants noticed increase in their knowledge after the contest, which is verified by statistical tests. However, students from the first stages received more benefits, probably because such students were more motivated and had more available time to be involved in the contest activities.

Keywords: education in information systems, education in software engineering, contest, problem-based learning.

1. Introduction

The undergraduate program in Information Systems (IS), offered at the School of Arts, Sciences and Humanities (EACH) of the University of São Paulo (USP), Brazil, has as one of its objectives to prepare students to work as professionals in the diverse environments of Information Technology (IT) business. This program presents a modern pedagogical project aimed at preparing professionals with extensive knowledge about the IT management and development process. This pedagogical project is included in an educational environment in which there is a strong promotion of interdisciplinary coursework and the students are placed as the main agent of their own learning, in accordance with the practice of Problem-based Learning (PBL; Bound and Feletti, 1998), which is strongly encouraged at the EACH-USP.

The IS program has a curriculum of four years. There are three specialization areas in this IS program, in which the students can focus their elective courses: (i) IS development; (ii) IT process management; and, (iii) internet technologies (EACH-IS, 2010). During their planning for the program, students can optionally enroll in all the offered courses associated to one or more of these areas. In the first area, there are Software Engineering (SE) courses whose goal is to work both: the SE process in all its phases; and, the related development technologies, such as programming languages, software development methodologies and environments, application servers and database (DB) management systems.

Teaching IS development presents a series of challenges as already discussed in prior studies, such as the one presented by Gibbs (1989). A major challenge presented was to ensure that the degree program presented a variety of principles, tools, and skills from multidisciplines such as computer science, computer engineering, industrial engineering, management science, mathematics, psychology, and economics. Currently, the teaching-learning process in this area is still influenced by some problems such as: predominantly theoretical courses; isolation of phases of the SE process in disjointed courses; little attention to certain phases of the development process; deficiency in the integration between the academic and business environments; and difficulty in motivating the students to learn the techniques and methodologies used in the area.

In order to propose alternatives to improve the preparation process of IS development professionals, the SE professors at EACH-USP held a stimulating contest on IS development, based on PBL principles. This contest had as main goals: (i) providing a real motivation for the use of SE formal techniques and methodologies, using an application domain which could be deployed in the real context of the IS program management; (ii) allowing the systematic application of all the phases of the IS development process in a single project; and, (iii) assessing the teaching-learning process currently applied in the IS program and identifying possible deficiencies.

To make possible achieve all these goals, the contest was carried out as an extracurricular activity inside the IS program, but not related to any specific course in the program. Historically, contests inside an undergraduate program are commonly performed inside a specific course, which can present a series of drawbacks, although they also presented a series of proven strength. The most important drawbacks to be avoided in a extracurricular approach are: difficulty in working interdisciplinary subjects – which means, in the IS development case, gathering several computing and software engineering related subjects; difficulty in motivating students because some of them participate in the contest only by requirement of the course; difficulty in using alternative learning methodologies, as PBL, since, for some programs, traditional methodologies must be used; and, difficulty in involve different professors and students (from different program stages) interested in the contest.

This paper presents the contest and the achieved results, through the following sections: Section 2 contains the related works that inspired this work, including a comparison among the initiatives; the description of the contest, including the information system that should be developed and the contest rules, is presented in the Section 3; Section 4

describes the evaluation of the contest, including a description of the participants, the questionnaires for assessing, the achieved results and the analysis of results; Section 5 presents a summary of Lessons Learned; and, in Section 6, the conclusions and present the future directions are presented.

2. Related Works

A major concern regarding teaching SE applied to IS is to reach better results by using active learning methodologies, including the application of PBL. This can include: simulation of problem situations (case study; Törngren *et al.*, 2007; Claypool and Claypool, 2005); simulation of corporate and cooperative environments (Sherrel and Mills, 2008; Hogan and Thomas, 2005; Schlimmer *et al.*, 1994); monitored insertion of the students in real problems in the industry; promotion of academic environments for distributed software development (Zagar *et al.*, 2008); and, several proposition of contests.

The most common initiatives for the contest realm are the programming marathons or Olympics, often promoted in technical-scientific events. In such examples, students must solve programming problems, which contributes to develop their abilities to work in teams, propose new solutions and work under pressure. These modalities usually have duration of hours and several stages – local, regional, national and global (ACM-SA, 2007; ACM-Int, 2010). Contests are designed so as to provide a rewarding experience and opportunity for achievement for all competitors; not just the winners (Cormack *et al.*, 2006).

A well-structured problem has a clear initial state, a known goal state, a set of rules to achieve the solution, and an optimal solution. On the other hand, an ill-structured problem is not clearly described and the information to solve the problem is not entirely contained in the problem statement; thus, it is not obvious what to do to solve it and a reasonable solution probably takes into account multiple views. Jonassen (2001) identified two kinds of ill-structured problems: case analysis and design. In case analysis, students deals with solution identification, alternative actions, and argumentation, while in design students are supposed to act on goals to produce artifact, and structure and articulate the problem.

Paulik and Krishna (2001) discuss the use of a contest in an undergraduate course to build a product (autonomous land vehicle), in which both – project and its documentation – are artifacts that must be evaluated. The authors argue that contests lead to greater interaction between students and professor which enriches the environment of the course.

The contest described and analyzed by Massey *et al.* (2006) was created based on PBL principles as the one presented here. Their objective was to encourage an environment in which students should leave the role of system end users and assume the responsibility for the development, decision-making processes and sales strategies modeling. They argue that this strategy allows the learning of technical concepts as well as the development of skills and knowledge independent of the technical domain, which are all highly relevant to the preparation of the future IT professional. This contest had as application domain the development of mobile applications. Unfortunately, this work does not present related

quantitative data that demonstrate the real benefits in terms of improving learning for students, as shown by this paper.

In a similar line, Firebaugh and Piepmeier (2008) showed the use of a contest to apply PBL in the design of micro robots. These authors highlight that the main difficulty in using PBL is to find a suitable problem that is sufficiently complex but also manageable within the time window of an undergraduate program. The authors noted that the approach promoted in-depth learning of specific topics but did not provide conditions to cover the entire scope planned for the course. Comparatively, the contest discussed here has the advantage of taking place in parallel to the regular courses, acting mainly as strengthening for the learning process.

The contest presented here, despite being held as part of a social-technical-scientific event, had educational objectives, as the Bowring (2008)'s ideas which refer to a programming contest. According to him, attention should be paid with respect to the process quality and to the addition of technical and artistic merit in the judgment criteria.

The following works are more closely related to the one presented here in terms of motivations and the object of development in the contest, although they have differences in terms of scope, focus or method.

Gotel *et al.* (2009) point out contests as a motivation for students to participate in a complete SE process. They emphasize that such approach covers a gap in the correlated traditional courses, in which is very difficult to get students delivering a quality software product at the end of the course. Differently from the contest reported in this paper, this experience occurred inside the scope of a specific course.

SCORE (Jazayeri and Mandrioli, 2009) is also a contest held as part of a scientific event, in the software engineering area – the International Conference on Software Engineering. Its implementation follows some features similar to the contest described here, such as long-term and qualifying phases. Whereas, on one hand, the SCORE is not scoped in a unique course from a undergraduate program, on another hand, its scope is so broad that is unfeasible to put into practice the goals of the contest presented here, as assessing the teaching-learning process currently applied in the program and identifying possible deficiencies, for which a controlled environment is essential.

Yang and Liu (2009) encourage the use of contests as part of a revision of the SE teaching practice. Among their motivations, two are more aligned with this work: minimizing the inherent abstraction in the traditional teaching process for the SE; and, promoting the integration of the concepts that are taught in a fragmented way. Yang and Liu's work is highly related to teaching SE in a non traditional way but it is focused on the scope of a unique course, without exploring interdisciplinary aspects, which differs their approach of our contest.

Other contests are carried out in the computing field to exploit other subjects than SE and IS. Ribeiro *et al.* (2009), for example, conducted a contest in the scope of undergraduate teaching of Artificial Intelligence. A competition framework, which involved Prolog programmed contenders and game servers, including an appealing GUI, was developed to motivate students on the deepening of the topics covered in class of a specific course.

3. Contest Description

The contest was held as part of the activities of the "First Information Systems Week", an event promoted by the IS program at EACH-USP. It took place some months before the event and the winners were announced in the end of the event. Sixteen teams have registered, from which seven of them were classified for the final stage.

In this section, the contest is presented, including the description of the system that should be developed by the teams and the rules established to manage the contest.

3.1. *Target Information System*

The information system, developed during the contest, aims to manage the website of the IS program at EACH-USP. More than a collection of static pages, the system allows the management – by authorized users – of the information related to the undergraduate program. Such information is stored in a DB and dynamically loaded in the website pages, allowing its visualization through an internet browser by different classes of users. As a result, professors, staff and students register and/or change information through the system, which then automatically publishes the information on the website pages. The system must provide a flexible management of users and user groups and of access permissions, including the implementation of the classic functions of a system administrator.

A summary of the most important items of information that should be handled by the system is shown in Table 1. Using a DB management system, the data registered separately should be related to others in order to create an information network accessible via the website hyperlinks. For example, to register a new research project, professors registered in the system can be selected as participants of this project; as a consequence, this project will be part of the set of descriptive information of the professor personal data, and vice versa.

Table 1

Items of information handled by the system

Item	Details
General information	Undergraduate program overview, basic contacts, news, events, received awards and honors, FAQs.
People	Professors, staff (positions/functions), students.
Education	Courses table, training (training vacancies, ongoing trainings, training reports), rules/regulations, forms/document templates, education laboratories.
Research	Research groups, research/orientation areas, research projects, research laboratories.
Extension	Extension projects, extension courses.
Meetings	Calendars, meeting minutes.
Publications	Technical reports, scientific initiation reports, under graduating monographs, training reports, papers, books.

3.2. *Contest Rules*

A set of rules was published (EACH-Contest, 2009) to regulate the contest. The most important of them are presented as follows:

1. *System requirements*: a preliminary version of the system requirements was published for all potential interested teams during the contest dissemination. Requirements were deliberately described in a general and imprecise way to be refined as part of the development of the system by the teams.

2. *Customers and stakeholders*: the organizing committee formed by professors of the IS program performed the role of system customers. Two interview meetings were scheduled with them for each registered team in the beginning of the contest, so that the work of requirements refinement could be carried out.

3. *Development methodology*: the teams were free to use the development techniques and methodologies and the programming technologies that they might prefer. Moreover, they were allowed to consult other professors of the IS program or system development professionals.

4. *Teams structures*: each team should be formed by two to four students regularly enrolled in any semester in the IS program. Teams from different semesters were allowed in the contest for the following reasons: (i) to achieve a considerably large amount of registered teams, encouraging competition between them; and, (ii) to allow a comparison of different possible effects of the contest on different stages of the IS program.

5. *Conditions for participation*: all participants should agree to develop the system as free software, using only development tools licensed for the EACH-USP.

6. *Artifacts delivery*: mandatory intermediate deliveries – with qualifying character – should be carried out in accordance with a previously established schedule; and a final delivery – with classification character – should be carried out in the end of schedule. The artifacts to be delivered were: requirements specification, described according to the IEEE Std 830-1998 standard (IEEE, 1998); system architecture; entity-relationship model (ER-Model); source and executable codes; system build scripts; automated scripts for unit and integration testing.

7. *Weights in the evaluation*: intermediate deliveries comprised 40% of the final score, divided into: requirements specification (15%); system architecture (12.5%); and, ER-Model (12.5%). Final delivery comprised 60% of the final score, divided into: acceptance testing (10%); stress testing (10%); performance testing (10%); graphical user interface – usability (10%); graphical user interface – design (10%); system build scripts (5%); and, testing scripts (5%).

8. *Scoring*: some professors evaluated the deliveries cited in item 7. Different professors collaborated, depending on the artifact type and their specialization fields. For each team was defined a score for each artifact; these partial scores were added to form the final score for each team. The winning team was the one with the highest final score. For the sake of simplicity, no distinction was made with respect to the stage of the undergraduate program of the participants, i.e., its semester in the IS

program, to choose the winning team, although this could be considered a potential source of unfairness.

9. *Awards*: the best team was awarded and two other teams received honorable mentions. The organizing committee received prizes, from private companies, which were offered to the winning teams, including vacancies in official training for certification and textbooks. All finalist participants received a certificate of participation.

10. *Schedule*: the following dates were applied: Nov, 10th 2008 – contest dissemination for students; Dec, 5th 2008 – teams registration deadline; Feb, 15th – delivery of requirements specification; Mar, 15th 2009 – delivery of system architecture and ER-Model, and presentation of first version of system prototype; Apr, 15th 2009 – presentation of second version of system prototype; Apr, 30th 2009 – delivery of final version of the system; May, 7th 2009 – indication of winners.

The purpose of Rules 1, 2 and 3 was to provide, according to the PBL methodology, an environment of free research in which the teams could develop self-learning skills and competencies related to management of cooperative work.

4. Contest Evaluation

The subject of this evaluation was the contest, described in Section 3, as a tool for learning, and learning strengthening, of the students involved. The focus was to identify the participants' SE knowledge acquisition in comparison to their prior knowledge. Both qualitative and quantitative data were collected. The qualitative data was obtained by questionnaires to assess the participants' perception of their improvement of SE knowledge. Quantitative data was collected by grading the artifacts produced by the teams and by small examinations administered to the participants of the contest. For the analysis of results, statistical tests were applied to assess the main hypotheses raised during data analysis. To allow quantitative analyses, statistical tests specific to small samples were applied, minimizing a possible limitation of a small number of students have participated in the contest and evaluation.

4.1. *Characterization of Participants*

Participants of the contest were students enrolled in the third, fifth and seventh semesters of the IS program at EACH-USP. The program requires eight semesters of coursework to obtain the bachelor's degree in IS. The participants' profile was quite diverse: from students who had attended various courses related to SE (seventh-semester students) to students who had attended only two semesters of Java programming (third-semester students). In addition to this diversity, some students had previous systems development experience in research projects or in development companies. Table 2 presents the number of semesters of coursework completed by the participants and their experience in systems development.

Table 2

Coursework completed and system development experience

Coursework completed			System development experience			
3rd sem	5th sem	7th sem	Undergrad research	Internship	Employment	None
4	12	12	8	8	3	9

The creation of the teams was free, though most of them were composed of students belonging to the same semester, with the exception of two teams which contained members from different semesters. At the beginning of the contest, 16 teams enrolled in the contest; the final stage was composed of seven teams, totaling 28 students.

4.2. *Assessment Questionnaires*

The questionnaires were submitted to the students after partial deliveries. Students completed the questionnaires individually in the presence of one of the professors from the organizing committee, in meetings also used for other purposes, such as elucidation about upcoming deliveries.

To minimize the validity risk of the data collection, it was made clear to participants that the data informed would not influence the contest results. The first one was applied after the delivery of the requirements specification, having 28 students answered; and, the second after the delivery the ER-Model and the system architecture, having 24 students answered.

The questionnaires had two sections: *Knowledge Identification*; and *Knowledge Assessment*. The former aimed at evaluating the previous experience of the student (whose data were used to build Table 2) and the qualitative self-perception of its knowledge before and after the delivery made; the goal was to identify the technical knowledge improvement perceived by the student by participating in the contest according to its own point of view, and not comparing with the other students or with some specific knowledge target. The objective was simple: to measure how much the student believes he/she improved in terms of technical knowledge. The latter section was designed to obtain an objective assessment of current knowledge of students and to identify differences in expertise between the participating teams.

Concerning to the *Knowledge Identification* section, in what follows it is presented a question in the first questionnaire applied to the participants to assess their learning perception.

*Regarding **Requirements Elicitation**, assign a value (from 0 to 10) representing your experience on this subject: ( ) **Before** this contest stage; ( ) **After** this contest stage.*

No reference value was given for students to answer this type of question, since they should answer according with their own perception. Thus, each student should interpret in its own way which means a zero, a five or ten value, for example. Moreover, the evaluation was more interested with the increment in knowledge; consequently, for example,

a student answering 2 and 4 would present the same result than another answering 5 and 7. Students were oriented to answer according to their own understanding of the range.

Additionally, with respect to the learning perception of the participants, after the second delivery, 15 similar questions were submitted to the participants regarding the developing of the ER-Model and the system architecture. The questions in the second questionnaire were about the following subjects:

A. **DB-ER_Model**: (1) ER-Models; (2) Simple Attributes; (3) Multivalued Attributes; (4) Simple Relationship; (5) Ternary Relationship; (6) Cardinality; Specialization; (7) Weak Entity;

B. **DB-R_Model (Relational Model)**: (1) Primary Key; (2) Secondary Key; (3) Foreign Key; (4) Composite Key;

C. **Architecture**: (1) Software Architecture; (2) Layered Architecture; (3) Architectural Models.

Concerning to the questionnaires' *Knowledge Assessment* section, the first questionnaire contained six questions on requirements specification drawn from the IEEE Std 830-1998 document and the second one contained three questions concerning the development of ER-Models and system architectures. The technical concepts covered in the questions were supposedly used in the preparation of delivered artifacts. In what follows two examples of questions, one for each of the two questionnaires, are presented.

1st questionnaire. *How a software requirements specification document can be checked with respect to its correctness? (a) Through the review of the requirements document by the client; (b) Through the review of the requirements document by the user; (c) By comparing the software requirements specification document with the system specification; (d) All of the above.*

2nd questionnaire. *Consider that you will develop an ER-Model to represent the data of a pizzeria. In such pizzeria, every pizza is identified by a number and has name, price and size. Price depends on size (small, medium and large). What is the* **wrong** *alternative? (a) The pizzas will be represented by entities; (b) The name will be represented by an attribute of the* Pizza *entity; (c) The size of the pizza will be represented by an entity* Size*; (d) The price will be represented by an entity attribute* Pizza.

4.3. *Achieved Results*

As some teams were formed by students from different semesters, the evaluation of the effects of the contest on the students were carried out individually, and not by team, so that a categorization by semester was possible.

Table 3 presents the most relevant data related to the students' perception of knowledge improvement. The concepts necessary for the elaboration of the delivered artifacts are listed in the first column; the other columns present, firstly, the number of students that pointed no knowledge increase and, in sequence, the number of students that pointed any knowledge increase, comparing before and after their participation in the contest, pointing it as bigger or equal to one unit, to two units, and so forth. The increase was

Table 3

Knowledge increase before and after the contest (by concept)

Concept	Perceived improvement						
	= 0	⩾ 1	⩾ 2	⩾ 3	⩾ 4	⩾ 5	⩾ 6
1) Requirements elicitation	4	24	17	14	11	7	7
2) Entity-relationship models	6	18	8	3	3	2	2
3) Software architecture	1	23	18	9	5	2	2
4) Layered architecture	3	21	14	10	7	4	3
5) Architectural models	6	18	13	6	6	3	1
Average percentage	16%	84%	56%	34%	26%	15%	12%

defined by the variation between what was considered as previous knowledge compared with the posterior, signed by a scale of one (minimal) to ten (maximal). The last row presents the average percentage of participants who perceived a minimum knowledge increase of units treated in the respective column, calculated as follows: the sum of the values in the column divided by the total possible responses to these five concepts, which would be 124 (i.e., 28 for the first concept and 24 for each of the next four concepts).

There were 16 questions related to knowledge improvement in both questionnaires. Table 3 presents the data of five questions for which the largest number of students indicated an increase in their perception of knowledge. These five concepts do not uniformly increase since some of them had a more robust improvement. The Mann-Whitney non-parametric statistical test (Kvam and Vodakovic, 2007) was run to validate this hypothesis, i.e., that "there were no significant differences between the increments given the concept groups – Requirements Elicitation, DB-ER_Model, DB-R_Model, and Architecture ". Considering paired differences with a normal distribution, applying the t-paired test, with an interval of 95%, it was found that the groups Requirements Elicitation and Architecture have larger increments that the groups ER_Model and DB-R_Model, rejecting the null hypothesis, regardless of the participant.

Table 4 presents data regarding the difference of knowledge improvement perceived by students of different semesters. While third-semester students perceived an average in-

Table 4

Knowledge increase before and after the contest (by semester)

Semester	Average prior knowledge	Average posterior knowledge	Average perceived improvement
3rd semester	2	7	5
5th semester	7	8	1
7th semester	7	8	1
All semesters	6	8	2

crease of five units, fifth- and seventh-semester students saw an average increase of only one unit. The last row presents data when all the 24 students are considered together, from the three semesters. To validate this difference, the Kruskal–Wallis nonparametric statistical test was conducted. The null hypothesis considered was that "there were no significant differences between the increments given the semesters in which students belonged". Having obtained a p-value of 0.007 (i.e., less than 0.05), the null hypothesis was rejected in favor of the alternative hypothesis of having significant differences.

Fig. 1 presents the five concepts that most students showed no knowledge improvement. Moreover, the graph presents the knowledge unit pointed by these students for each of these five concepts, as average values, which are equal for both before and after their participation.

Table 5 presents the students' percentage of correct answers for the questions applied in each questionnaire. The first column indicates the semester that the student is enrolled; the second and third columns present information about the average number of students' correct answers for the first and second questionnaires, for each semester. Finally, the fourth column presents the average number of correct answers between both questionnaires. The Kruskal–Wallis test was run to verify differences between the grades taken by the participants of the different semesters. The null hypothesis considered was that "there were no significant differences between the grades given the semesters in which students belonged". The null hypothesis could not be rejected for any case, since the p-
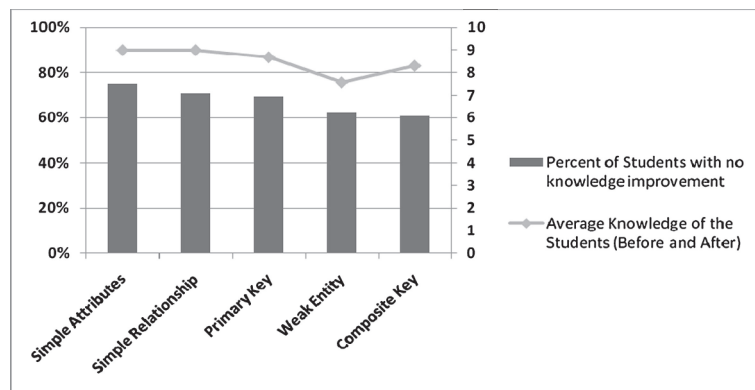

Fig. 1. Five main concepts for which most students did not show any knowledge improvement. The secondary data of the graph show the average knowledge of the students showing no improvement for these concepts.

Table 5

Objective assessment of knowledge after contest participation

Semester	1st questionnaire	2nd questionnaire	Average
3rd semester	70.50%	66.00%	68.25%
5th semester	59.50%	75.27%	67.39%
7th semester	56.58%	58.78%	57.68%

Table 6

Partial deliveries' grades

Team	Sem	Requirements specification	ER-model	System architecture	Intermediary grade
1	3	9.0	6.3	8.5	3.2
2	5	5.0	6.8	3.0	2.0
3	5	7.0	6.3	2.0	2.1
4	5	6.5	7.8	7.0	2.8
5	7	9.5	6.0	3.0	2.6
6	7	6.5	7.0	6.0	2.6
7	7	8.5	8.5	4.0	2.8
8	7	7.5	6.8	7.5	2.9
Average		7.4	6.9	5.1	2.6

values obtained were 0.333 and 0.096 (both greater than 0.05) for the first and second one respectively.

Finally, Table 6 presents the grades assigned to teams regarding the partial deliveries of the contest, according to the criteria presented in Section 3.B. In the end of the contest, the teams did not deliver completely operational systems to be tested, since it was considered too complex to be developed in this context and in the time frame available. Consequently, the evaluation criteria planned for the last stage was not followed. The Kruskal–Wallis test was run to verify differences between the grades assigned to the teams, and consequently to the participants, of the different semesters. The null hypothesis considered was that "there were no significant differences between the grades given the semesters in which students belonged". The null hypothesis could not be rejected for the ER_Model artifact, with a p-value of 0.097 (greater than 0.05); but it was rejected for the Requirements Elicitation and System Architecture artifacts with p-values of, respectively, 0.001 and 0.002 (less than 0.05), and in both cases the third-semester students had better grades.

4.4. *Results Analysis*

Data reported in Table 3 indicate the beneficial effect for students. In a general way, it is observable that more than a third part of the participants perceived an improvement in their knowledge for 13 of the 16 questions related to the *Knowledge Identification*. For five of these questions, the increase can be considered significant as at least three quarters of the students pointed some knowledge increase; and, for four of these questions, the indicated increase was of two or more units for about half of the students.

On the other hand, for some questions, many students perceived no improvement at all in their knowledge with the contest, as presented in the Fig. 1. For all these five concepts, students pointed a previous knowledge very close of what they consider the maximum possible, which explains the difficulty to have some increase in these cases. Moreover,

they are all specific Entity-Relationship Diagram concepts, which can be understood as well learned by them in the DB basic course.

Concerning the *Knowledge Evaluation*, students were able to correctly answer more than 50% of the objective questions that were applied to them, as indicated in Table 5.

Nevertheless, the knowledge improvement was not uniform considering the different student categories. The third-semester students presented higher increases in terms of knowledge increase perception (see Table 4). Also, according to data in Table 6, the third-semester students received two of three above-average grades regarding the delivered artifacts. Some hypothesis can justify this better performance. First, the third-semester students were more motivated to face the contest as a learning opportunity since they had still not learned the content necessary to develop the target system in the undergraduate courses they have taken so far. In addition, all four students of the third semester who were approved to the contest final stage composed a selective group since they have good grades in all of the previous courses they completed in the IS program. Some other teams of the third semester were not classified to the final stage. A simple possible explanation for the better performance of the third-semester students is that since students in first stages have more to learn, for them the experience is much more interesting and enriching.

Another important observation was that the third-semester students had very good grades for the objective questions of the first questionnaire, but not so good ones for the second. A possible explanation for this fact is that there was an extremely clear model (IEEE Std 830-1998 standard) that was used as the basis for the questions prepared for the first questionnaire, but no model was used for the second one. Anyway, even with this constraint, in the second questionnaire, the third-semester students had better grades when compared to the seventh-semester ones. A possible explanation to the worst performance of the seventh-semester students is that some of them are already employed or are trainees in software development companies, and hence they may have problems concerning spare time to participate in the contest and also to follow the regular activities of the other undergraduate courses at the university. Another possible explanation is that since the students had already learned the content necessary in their regular courses, they may not have performed a review of these concepts before the artifacts production, trying to use only the knowledge they just remembered. The students were not previously notified about the content and shape of the questionnaires.

Comparing the results from Table 4 and Table 5, data show that although third-semester students presented the highest perceived knowledge improvement (according to Table 4), they presented the same performance, in the average, that the students from the other semesters in terms of objective questions assessment (according to Table 5). Whereas they have presented a better performance for the first questionnaire, they revealed a decrease in knowledge in the objective assessment for the second questionnaire. These two data group can lead to the conclusion that third-semester students have a better perceived knowledge improvement which was not sustained by the second part of the objective assessment results. I.e., although they believe they have learned more than the other students, in fact such learning has not materialized in practice the enough to reach the point of knowledge already acquired by students in more advanced semesters.

5. Lessons Learned

In a general way, the professors responsible for implementing this contest considered that it has produced good results for the involved students, although some limitations can be pointed out. As good results, two main ones can be highlighted: (i) the enrolled students have demonstrated a knowledge improvement in terms of both subjective and objective measurements; and, (ii) the responsible professors could notice deficiencies in the current curriculum structure of the IS program being applied for the students, since some students presented some important deficits during the contest which were not expected.

An important limitation of the contest was the unwell dimensioned scope of the system to be developed which caused a frustration in both the students and the professors involved in this activity since no team could finish all the contest phases. This problem may have done some teams gave up during the partial deliveries. On the other hand, this is a common issue present in the software industry regarding the software development projects for which the persistent teams were faced up to in a simulation type: projects whose scope is larger than the estimated time for its development.

Another limitation of this contest was the low number of enrolled teams, caused probably because it was a not mandatory activity inside the curriculum structure of the IS program. Another reason is that the IS area, in a big city as an important financial center as São Paulo in Brazil, allows students easily getting internship opportunities that leaves them with little free time to participate in extracurricular activities. However, we believe that only with this type of rules – according to the objectives of our contest as defined at this paper Introduction section – we could have got the achieved results, as comparing students from different semesters.

An example of further important consequence of the contest evaluation was the analysis of the curriculum structure of the IS program at EACH-USP. In a free comments field of the questionnaires, a great amount of students informed that the *Software Architecture* area had been only superficially taught in the undergraduate courses. This qualitative information is confirmed by the higher learning improvement perception unit for the questions related to software architecture, as presented in Table 3. This information had already been used to support a curricular change in such a way to present this theme at the courses related to the SE.

Although not directly evaluated, the effect of the PBL methodology can be pointed as one of the best results of this contest. Students, provided with some basic technical knowledge in the IS development field, demonstrated an impressive self-learning power. The responsible professors tried to tutor the students following the PBL principles, taking account that these students had a specific course in the first and second semesters of the IS program related to PBL methodology. This impact could have been better evaluated in the contest assessment so that the application of PBL methodology into a IS program could have been a target here.

Regarding the contest evaluation, we also see some strengths and some weaknesses. We have tried to perform some quantitative analysis, upon some subjective and some objective data, to base our results analysis. However, the small sample does not offer many guarantees of results reliability, although tests to small samples have been applied.

Anyway, our intention was provide some value in the results analysis instead of working only with qualitative assessments, which we believe having achieved. Such quantitative analysis always leads us to see the problem in a new way not though before looking to the numbers. Although some limitations could be raised regarding the quantitative analysis undertaken, they were useful in contributing to present more data to base discussions.

Finally, analyses under different points of view should have been performed so that the findings of the contest could be better understood; however, they would be only feasible if a greater diversity of data had been gathered during the contest running. An example of such further analysis could be a discussion on which aspects can influence the findings on how students interpret their own learning and improvement, comparing those reporting better results with the other ones.

6. Conclusion

This paper presented the evaluation of an IS development contest as a tool for learning, and for learning strengthening, for the students of IS undergraduate program at the EACH-USP. The results obtained indicate that, in a general way, the students perceived a knowledge improvement concerning SE themes after the contest participation. However, more specifically, the achieved results reveal that the students of the initial semesters of the IS program got more benefits when compared with the ones of the final semesters. Being more motivated and having availability to participate in the contest are some possible explanations for this result.

Although there has not had a great success in the final stage of the competition, given the fact that the teams failed to deliver complete systems as specified, the early stages were successful. The obtained results are of two types: immediate help to the participating students in improving their knowledge in areas of deficit; and, obtaining information to analyze the curriculum of the IS program to propose improvements in order to help all the students enrolled.

These evaluation results suggest that methodologies based on the PBL principles are important tools to be used in IS programs. Moreover, the incentive through recognition and rewards may be used as motivating issues to improve student learning in this area. Based on these results, quite recently, a new regular course was defined to be added to the curriculum structure of the IS program at the EACH-USP. In this course, to be offered from the second half of 2010, all of the enrolled students may have the chance to get prizes in order to improve their interest and participation.

As future work, this new regular course will be used to allow these and other students to participate in a better contest, improved with the lessons learned obtained from the first edition. New data should be collected so that further analysis can be conducted aiming at a continuous improvement in this course and in the IS program as a whole.

References

ACM South American Programming Contest (2007). `http://www.ic.unicamp.br/~acmcntst/`.

ACM international collegiate programming contest (2010). `http://cm.baylor.edu`.

Bound, D., Feletti, G. (1998). *The Challenge of Problem Based Learning*, 2nd edition. Routledge.

Bowring, J.F. (2008). A new paradigm for programming competitions. In: *Proc. of the 39th SIGCSE Technical Symposium on Computer Science Education*. Portland, USA, 87–91.

Claypool, K., Claypool, M. (2005). Teaching software engineering through game design. In: *Proc. of the 10th Annual Conference on Innovation and Technology in Computer Science Education*. Caparica, Portugal, 123–127.

Cormack, G., Munro, I., Vasiga, T., Kemkes, G. (2006). Structure, scoring and purpose of computing competitions. *Informatics in Education*, 5(1), 15–36.

EACH Contest – First Information Systems Contest of the EACH-USP (2009). `http://each.uspnet.usp.br/si/SemanaSI/concurso2.html` (in Portuguese).

EACH IS – Undergraduate Program in Information Systems at the School of Arts (2010). Sciences and Humanities of the University of São Paulo. `http://each.uspnet.usp.br/each/cursos.php?pagina=sistemas-informacao` (in Portuguese).

Firebaugh, S.L., Piepmeier, J. (2008). The roboCup nanogram league: An opportunity for problem-based undergraduate education in microsystems. *IEEE Transactions on Education*, 51(3), 394–399.

Gibbs, N.E. (1989). The SEI education program: The challenge of teaching future software engineers. *Communications of the ACM*, 32(5), 594–605.

Gotel, O., Kulkarni, V., Say, M., Scharff, C., Sunetnanta, T. (2009). A global and competition-based model for fostering technical and soft skills in software engineering education. In: *Proc. of the 22nd Conference on Software Engineering Education and Training*. Hyderabad, India, 271–278.

Hogan, J.M., Thomas, R. (2005). Developing the software engineering team. In: *Proc. of the 7th Australasian Conference on Computing Education, Newcastle*. Australia, 203–210.

IEEE (1998). *Recommended Practice for Software Requirements Specifications*, IEEE Standard 830.

Jazayeri, M., Mandrioli, D. (2009). SCORE: The first student contest in software engineering. In: *Proc. of the 31st International Conference on Software Engineering Companion*. Vancouver, Canada, 487–488.

Jonassen, D.H. (2001). *What is Problem Solving?* `http://media.wiley.com/product_data/excerpt/79/07879643/0787964379.pdf`.

Kvam, P.H., Vodakovic, B. (2007). *Nonparametric Statistics with Applications to Science and Engineering*. Wiley-Interscience.

Massey, A.P., Ramesh, V., Khatri, V. (2006). Design, development, and assessment of mobile applications: the case for problem-based learning. *IEEE Transactions on Education*, 49(2), 183–192.

Paulik, M.J., Krishnam, M. (2001). A competition-motivated capstone design course: The result of a fifteen-year evolution. *IEEE Transactions on Education*, 44(1), 67–75.

Ribeiro, P., Simoes, H., Ferreira, M. (2009). Teaching artificial intelligence and logic programming in a competitive environment. *Informatics in Education*, 8(1), 85–100.

Schlimmer, J.C., Fletcher, J.B., Hermens, L.A. (1994). Team-oriented software practicum. *IEEE Transactions on Education*, 37(2), 212–220.

Sherrel, L.B., Mills, D.L. (2008). Introducing software engineering processes via games and simulations: A trip-lets initiative. *Journal of Computing Sciences in Colleges*, 23(4), 133–139.

Törngren, M., Grimheden, M., Adamsson, N. (2007). Experiences from large embedded systems development projects in education, involving industry and research. *ACM SIGBED Review*, 4(1), 55–63.

Yang, C., Liu, Y. (2009). Teaching reform and practice on the software engineering course. In: *Proc. of the 1st International Conference on Information Science and Engineering*. Nanjing, China, 3470–3473.

Zagar, M., Bosnic, I., Orlic, M. (2008). Enhancing software engineering education: A creative approach. In: *Proc. of the 2008 International Workshop on Software Engineering in East and South Europe*. Leipzig, Germany, 51–58.

M. Fantinato was born in Santa Fé – PR, Brazil, in 1977. PhD in computer science (2007) and ME in electrical engineering (2002) at the State University of Campinas, Campinas – SP, Brazil; and, BS in computer science (1999) at the State University of Maringá, Maringá – PR, Brazil. Currently, he is an assistant professor in the School of Arts, Sciences and Humanities at the University of São Paulo, São Paulo – SP, Brazil, since 2008. His main research interests are business process management, service-oriented computing, electronic contracts, software product line, and software testing.

M.L. Chaim was born in Piracicaba – SP, Brazil, in 1965. PhD (2001), ME (1991) and BS (1987) in electrical engineering at the State University of Campinas, Campinas – SP, Brazil. Currently, he is an assistant professor in the School of Arts, Sciences and Humanities at the University of São Paulo, São Paulo – SP, Brazil, since 2005. His main research interests are software testing and debugging, software maintenance, software development methods, and empirical software engineering.

M. Morandini was born in Jundiai – SP, Brazil, in 1968. PhD in production engineering (2002) at the Federal University of Santa Catarina, Florianopolis – SC, Brazil; MS in computer science (1996) at the University of São Paulo, São Carlos – SP, Brazil; and, BS in computer science (1992) at the Paulista State University, São José do Rio Preto – SP, Brazil. Currently, he is an assistant professor in the School of Arts, Sciences and Humanities at the University of São Paulo, São Paulo – SP, Brazil, since 2006. His main research interests are usability project and evaluation and software testing.

S.M. Peres was born in Mandaguari – PR, Brazil, in 1974. PhD in electrical engineering (2006) at the State University of Campinas, Campinas – SP, Brazil; ME in production engineering (1999) at the Federal University of Santa Catarina, Florianópolis – SC, Brazil; and, BS in computer science (1996) at the State University of Maringá, Maringá – PR, Brazil. Currently, she is an assistant professor in the School of Arts, Sciences and Humanities at the University of São Paulo, São Paulo – SP, Brazil, since 2007. Her main research interests are education in computing, artificial intelligence and data analysis.

E.F. Tuesta was born in Lima, Peru. PhD in electrical engineering (2002) and MS in statistics (1999) at the University of São Paulo, São Paulo – SP, Brazil; and, BS in statistics (1994) in Lima, Peru. Currently, he is a professor in the School of Arts, Sciences and Humanities at the University of São Paulo, São Paulo – SP, Brazil, since 2005.

Varžybų taikymas siekiant pagerinti mokymąsi informacinėse sistemose: tarpdisciplininis ir neauditorinis požiūris

Marcelo FANTINATO, Marcos Lordello CHAIM, Marcelo MORANDINI,
Sarajane Marques PERES, Esteban Fernandez TUESTA

Akademinėje aplinkoje varžybos, kaip vienas iš mokymo metodų, yra naudojamos, kad imituotų tam tikras profesines situacijas, reikalaujančias iš dalyvių daugiau aktyvumo, nei įprastinio mokymosi metu. Šiame straipsnyje aprašoma studentams skirtų tarpdisciplininių varžybų patirtis. Pateikiamos varžybų taisyklės, kurios susideda iš keleto reikalavimų komandų dalyviams ir organizaciniam komitetui. Straipsnio autoriai yra sudarę varžybų galutinio vertinimo balo struktūrą. Aptariami varžybų rezultatai, kurie rodo, kad didelė dalis dalyvių pastebėjo jų žinių progresą po šių varžybų.