Bringing Real-life Practice in Software Project Management Training Through a Simulation-based Serious Game

Alejandro Calderón and Mercedes Ruiz
Department of Computer Science and Engineering, University of Cádiz, Cádiz, Spain
alejandro.calderon@uca.es, mercedes.ruiz@uca.es

Keywords: Software Project Management, Simulation, Serious Games, Bloom's Taxonomy.

Abstract: Nowadays, we can observe a lack of tools that allow teaching and assessing in Software Project Management in a more real-life way. In this paper, we discuss the need of developing tools to teach in the scope of Software Project Management, and how we can achieve that learners test their knowledge and cover the needed educational objectives, in a risk-free environment through a serious game. Hence, we propose ProDec, a simulation-based serious game to teach and assess in software project management. This tool is an attempt to surpass the limitations found in similar proposals. The paper also maps the stages of the game lifecycle to the levels of Bloom's taxonomy to show how the game helps to achieve different levels of educational objectives.

1 INTRODUCTION

After the software crisis in the beginning of the 1970s, experts began to be aware about the importance of software engineering as a means to provide a set of methods, tools and procedures for the development of quality software, within the constraints of cost and time. However, the 2011 edition of the CHAOS report found that only 37% of all Information Technology (IT) projects succeeded in that they were delivered on time, within budget, with all the required features and functions. IBM research on the reasons for IT project failure concluded that 54% of IT projects failures are a direct result of poor project management.

The importance of teaching software project management for IT learners has always been supported by organizations such as the Association for Computing Machinery (ACM) and IEEE-Computer Society in their joint task force curricula. In the Curriculum Guidelines for Undergraduate Degree Programs in Computer Science curricula recently released (ACM/IEEE, 2013), these organizations have not only highlighted the importance of this matter, but they have also emphasized the need of teaching software project management in a highly practical way, where learners can test their knowledge in real-life scenarios. By a highly practical learning, it is intended that future practitioners acquire professional practice *during*, and not after, their studies.

However, despite the importance of training in software project management, many authors conclude that software project management subjects are still basically taught following a highly theoretical pattern and, as a consequence, learners do not show much interest in them (Ibrahim, 2011).

Compared with other degrees, such as Medicine or Aeronautics, IT learners start their professional life with a serious lack of real-life practical skills. As a result, new professionals need to develop their experience by working in real projects, where the effects of an inadequate plan or a bad decision can lead to project failure or the loss of significant profit.

A serious game is a game with the purpose of training or educating users. They can help in situations like the one described above, as tools to acquire experience and motivate learners, given their engaging nature. Moreover, simulation-based serious games allow us to bring real-life scenarios into the learning process in a risk-free environment.

In this paper, we extend a previous work on ProDec (Calderón & Ruiz, 2013), which is a simulation-based serious game for software project management training.

In the scope of this work, we address the following research questions:

- RQ1: What are the main weaknesses of current serious games for teaching software project

management and how can ProDec overcome them?

- RQ2: What is the educational effectiveness of ProDec according to Bloom's taxonomy?

In order to answer these questions, we have performed the following steps:

1. We have searched and analyzed the related work on different proposals of serious games in software project management.
2. We have defined a set of criteria to compare these proposals and identified their strengths and weaknesses.
3. We have evaluated ProDec using this same set of criteria.
4. We have mapped the educational requirements defined for each level of Bloom's taxonomy with the different stages of ProDec lifecycle to find out the educational effectiveness of ProDec according to this well-known taxonomy.

This paper is structured as follows: Section 2 presents the works related to our proposal and provides a comparison of similar proposals Section 3 briefly reviews Bloom's taxonomy since that framework has been used to evaluate the coverage of learning objectives of our proposal; Section 4 describes the serious game developed; Sections 5, 6, and 7 show the evaluation of our proposal using Bloom's taxonomy; Section 8 provides information about the state of ProDec's evaluation process. Finally, Section 9 summarizes the paper and presents our conclusions and future work.

2 ANALYSIS OF CURRENT PROPOSALS

There are several serious games in the field of software engineering education as Caulfield, Xia, Veal, and Maj show in their systematic review of the literature (Caulfield et al., 2011). However, this study also shows that the serious games that focus on software project management are scarce and have a quite specific scope. If we focus on the serious games developed for the area of software project management, the following ones are well-known examples:

- SIMSOFT (Caulfield et al., 2011) is a serious game materialized as a printed game board, that shows the players the flow of the game, and a Java-based board, where the players can see the current and historical state of the project and adjust the project's settings. It mainly focuses on human resource management, with an emphasis on how the ability of the staff affects the outcomes of the project.
- DELIVER! (Von Wangenheim et al., 2012) is also based on a printed game board designed to help learners develop the skills needed to measure and control project performance by applying the Earned Value Management technique (Project Management Institute, 2005). One of the main aims of this game is to motivate students in their learning process.
- SimSE (Navarro & Van Der Hoek, 2004) is a serious game completely developed as a software tool that is based on software project simulation. The game supports several development methodologies and focuses on the development of abilities for software process management.
- SESAM (Drappa & Ludewig, 2000) is another serious game developed as a software application that uses simulation techniques to motivate learners in learning software project management. Players take the role of a project manager and must plan and control a simulated project.
- The Project Manager Game (Games by Robc, 2013) is an online serious game where users have to allocate the most suitable staff to particular tasks and complete their project on time within the allocated budget.

In order to perform a comparison of these proposals and also to identify their strengths and weaknesses, we propose the following set of features. These features are based on our analysis and also on the ones used by Caulfield and his colleagues (Caulfield et al., 2011).

a) F1. Coverage of software project lifecycle, that is, the phases of the project lifecycle the game deals with.
b) F2. Coverage of the Project Management Body of Knowledge (PMBOK) (Project Management Institute, 2013), that is, the different processes and techniques described in the PMBOK that the game helps to learn and practice with.
c) F3. Support of revised Bloom's taxonomy (Krathwohl, 2002), that is, the level of coverage of each level of Bloom's taxonomy the game offers.
d) F4. Support for automatic learning assessment, that is, the level to which the game helps trainers to assess the learners automatically.
e) F5. Simulation-based, that is, if the game makes use of simulation techniques, and to which extend.

f) F6. Game flexibility, that is, if the scenarios of the game are static or they can change dynamically during the game play.

Table 1 summarizes the comparison of the serious games previously described. In this table, the columns represent the serious games analyzed and the rows are the set of features already defined

According to the PMBOK, created by the Project Management Institute (PMI), the project lifecycle is defined by five stages: Initiation, Planning, Controlling & Monitoring, Executing and Closing. Those stages give also name to the five groups of management processes that need to be carried out with the help of techniques. All the serious games analyzed focus on a specific stage of the project lifecycle (F1) and/or help to practice a specific process or technique in that stage (F2). None of the games analyzed give learners the possibility of studying the complete lifecycle of a project from the initiating stage, where the project characteristics are defined, to the closing stage, where the project ends and it is the time for analyzing the results and deriving the lessons learnt.

If we compare the educational objectives that can be achieved by using these games with a well-known taxonomy of learning objectives such as Bloom's taxonomy, we can conclude that only SIMSOFT covers all the levels of the taxonomy.

Regarding the assessment of the skills that learners acquire by using the games (F4), all the games analyzed use traditional methods for learners' assessment, and none of them includes an automatic process of gathering and analyzing information about the game play to support an automatic assessment. Furthermore, only two out of the five games analyzed make use of simulation as a means to add realism to the game scenarios (F5). However, the two games that make use of simulation provide scenarios supported by static simulation models. This means that the scenarios the player can play are fixed offering a poor flexibility and leading to an early loss of player's motivation.

From the analysis of the current proposals of serious games for teaching software project management, we found that their main weaknesses are:

- They focus on learning specific techniques of project management or specific stages of the project's lifecycle.
- They do not usually reach all levels of Bloom's taxonomy.
- They do not allow to assess learner's new skills automatically.
- They do not offer flexibility.

Taking this analysis into account, we can conclude that further research is needed to overcome the weaknesses found in the application of serious games for software project management. For this reason, in this paper, we propose ProDec, a serious game designed to overcome these weaknesses.

Table 1: Comparison of serious games for software project management.

Serious Game	SIMSOFT	SESAM	SimSE	DELIVER!	The Project Manager Game
Coverage of software project lifecycle (F1)	Planning, Controlling & Monitoring	Planning, Execution, Controlling & Monitoring	Planning, Execution, Controlling & Monitoring	Monitoring	Planning
Coverage of PMBOK (F2)	Staff management	Staff management	Staff management	Earned Value Analysis	Staff management
Coverage of Bloom's taxonomy (F3)	All levels	Knowledge	Knowledge	Application	Application
Support for automatic assessment (F4)	Not allow to obtain automatic assessment reports				
Simulation-based (F5)	No	Yes	Yes	No	No
Game flexibility (F6)	Statically scenarios provided by the game			Static board scenario	Random scenarios

3 BLOOM'S TAXONOMY

Bloom's Taxonomy (Bloom et al, 1956) is a widely accepted classification of learning objectives within education. It constitutes a common framework for learning proposals' assessment and comparison. In the world of serious games, this taxonomy is also used for these aims. For this reason, we have also chosen Bloom's taxonomy to assess the educational objectives that our proposal can reach.

Bloom's taxonomy refers to a classification of the different objectives that educators set for learners. It divides educational objectives into three domains: Cognitive, Affective, and Psychomotor. If we focus on the Cognitive domain, where skills revolve around knowledge, comprehension, and critical thinking on a particular topic, we find six levels sorted in a hierarchy. These levels, in ascending order, are shown below:

- Knowledge, the subject is able to recall previously learnt information. They recognize information, ideas, facts, dates, etc. in an approximate way as they have learnt.
- Comprehension, the subject is able to demonstrate the understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions, and stating the main ideas.
- Application, the subject is able to solve problems in new situations by applying acquired knowledge, facts, techniques and rules in a different way.
- Analysis, the subject is able to examine and break information into parts by identifying motives or causes, and they are able to make inferences and find evidence to support generalizations.
- Synthesis, the subject is able to create, integrate, combine ideas, pose and propose new ways of doing. They are able to apply previous knowledge and skills to produce something new or original.
- Evaluation, the subject is able to make judgments about information, validity of ideas or quality of work according to personal opinions based on a set of criteria.

A revision of Bloom's taxonomy (Anderson & Krathwohl, 2001) establishes that in the cognitive domain there are six levels but not all are sequential in a hierarchical order. This revision proposes that the three lower levels are sorted in a hierarchical way, but the other three, the higher levels, are placed in parallel levels.

Like Caulfield, Xia, Veal, and Maj show in their serious games review (Caulfield et al, 2011) that the majority of authors in this scope are satisfied with placing their educational objectives at the basic levels of Bloom's taxonomy, even some of the proposals only reach the Knowledge one. The only exception that we found is SIMSOFT that reaches the higher levels of the taxonomy.

In the following sections we describe the different stages of ProDec and we perform an analysis of Bloom's taxonomy levels that our proposal satisfies.

4 PRODEC OVERVIEW

ProDec is a serious game to teach and assess learners in software project management. Its main goal is that learners acquire decision-making skills on problems that can appear within a software project lifecycle. Consequently, it helps learners start their professional career with some practical abilities for their profession.

The aim of the game is to successfully manage a software project. The game is over when the project significantly overruns either the approved budget or the allocated time. The player wins when they are able to complete the project within the time and cost limits. Besides, ProDec not only focusses on problem solving during the executing and controlling stages of a software project, but it also invites players to fully plan their own project and then monitor and control its progress by simulating its execution.

At the same time, ProDec helps trainers in the assessment of the skills that learners must acquire by playing the game. To do this, ProDec accepts the assessment criteria as an input of the trainer, and automatically: a) gathers data about these criteria during the game plays, b) analyzes the data collected, and c) shows an assessment report for the learner and the trainer.

ProDec is a game to be played in teams, so that it can also help to develop some soft skills in project management such as leadership and communication skills. Actually, ProDec does not teach learners the basic principles of software project management. Before playing, the players need to acquire these principles in lectures. Therefore, ProDec is a tool to be used in advanced stages of an academic course. Although the game is thought to be used by teams, it can also be played individually.

4.1 Lifecycle

ProDec offers players two ways of playing, a full game and a quick game. In a full game, players begin a project from scratch while, in a quick game, players can select one of the scenarios previously uploaded by the trainer. In this second case, players can only practice their project monitoring and controlling skills. Disregarding the mode of playing, players go through different stages of the project lifecycle with the goal of ending a project successfully. Within the game, there are three phases of a game play's lifecycle, which are Onset, Execution and End phases.

The following sections further describe the functionality associated with each phase of the game and the level of coverage of the learning objectives according with Bloom's taxonomy.

5 ONSET PHASE

ProDec's Onset phase is the first contact that learners have with the game. If the players select to play a quick game, they have to go through the different project scenarios available in the game and read and evaluate each project's features. In this case, learners need to know the main concepts and principles about software project management and need to understand the information and data that the game shows, so that they can get an idea about the difficulty of each proposed project scenarios. So, in this phase, learners need to *remember* and *comprehend* the information provided about the software projects to begin the play successfully. In this way, Knowledge and Comprehension levels of Bloom's taxonomy are covered.

On the other hand, if players select to play a full game, they follow a process that guides them in making the software project plan. This process is made of five sequential stages and allows learners to provide all the data needed to create a new software project plan. The stages that make this process are the following: project information, size estimation, project team definition, tasks definition and risks analysis.

5.1 Project Information

Project Information is the first stage of the process of creation of a software project plan. In this stage, learners have to enter the general information of the project about its scope and features, such as the salary of the workers, the length of the project, the numbers of use cases, etc., that are necessary to begin the size estimation stage. In order to do this task, learners need to know and understand the different concepts about software project information that the game uses. If learners do not know or understand properly these issues, it is likely that they will end up in a failed game, as it happens in real life, where a project misinterpretation leads to a failed project. The training about the concepts used in the phase of the game has to be offered during the lessons taken before playing. As in this stage, in all the next stages of this process learners need to know, properly understand and remember the concepts learnt in the lectures, so that the data they provide to the game is consistent. Hence we can see that Knowledge and Comprehension levels of Bloom's taxonomy are covered by the game.

5.2 Size Estimation

In this stage, players provide the estimate of the size of the project starting with the total number of use cases. Then, learners need to calculate and enter the size of each use case. To estimate the size of each use case, learners use Albrecht's Function Point Analysis (Albrecht, 1979). A function point is a unit of measurement to express the amount of functionality of an information system. In this stage of the game, learners need to *apply their knowledge* to calculate the different size of the use cases before entering the data to continue. Consequently, the Application level of Bloom's taxonomy is covered.

Besides, in this stage learners apply Albrecht's technique and get the results of under- or over-estimating in real-life project scenario, which is a different situation to the traditional one focused on blackboard activities.

5.3 Project Team Definition

During this stage, learners design their project team by defining its members. For each team member, players have to select some features for their personality and past work experience. Currently, the personality of an employee is made of two traits that the player has to select from a range of twenty two available ones. The set of personality traits included in the game is based on the sixteen personality factors described by Cattell (Cattell et al. 1988). In this stage, the game helps learners to think about the different outcomes of mixing personalities in a team and the importance of achieving a good team synergy in a successful project.

5.4 Tasks Definition

Once the size has been estimated and the project team designed, the following stage asks players to define the project tasks. In this stage of the process, players define the project tasks, and enter, for each of them, the time data, the budget allocated, and its predecessor tasks. Consequently, this stage asks the player to provide the information gathered in a PERT diagram (Moder, 1983) of the project. PERT technique is recommended in PMBOK's Time and Cost knowledge areas and it is included in the Project Management Professional (PMP) exam. This function allows learners to apply their knowledge about PERT diagrams in real-life scenarios and to analyze its features within a project unlike more traditional approaches where knowledge is exclusively acquired by solving individual exercises during the lectures. Besides, in this stage, players have also to allocate tasks to the team members. To do this successfully, they need to analyze all the information entered about the tasks description and the personality traits and work experience of the team members with the aim of allocating the most suitable staff to each task. It can be seen that the Analysis level of Bloom's taxonomy appears in this stage of the game.

5.5 Risk Analysis

The last stage of the process that players follow to make the software project plan is the risk assessment and analysis. As its name indicates, in this step, the players perform the quantitative analysis of the risks that can appear along the execution of the project. In a quantitative risk analysis, players need to enter the risk probability and the loss magnitude for every risk identified. The risk probability is the likelihood of occurrence of a risk. The loss magnitude is the potential loss the project may suffer in the case that the risk appears and it is not properly controlled. Later, these risks will be transformed into events of the simulation model. The probability that these risks occur is defined by the risk probability. The loss magnitude establishes the time delay that the tasks being performed at the moment when the event occurs suffer.

Based on the knowledge acquired in the lectures and the information entered about the project, the players are required to analyze the project within its scope with the aim of making a proper list of the risks to be taken into account. Again, we can see how the Analysis level of Bloom's taxonomy is covered.

Along all the onset phase, learners have to analyze all the elements of each stage to get information and synthesize this information into new ideas and decisions to make. Moreover, given the social character of the game, before making a decision such as allocating staff to a task or defining the probability of a risk, players need to analyze the information of the project to argue, defend, discuss, evaluate, negotiate and agree the best decision within their team to make a good project plan. This feature allows the learner to work at the highest levels of Bloom's taxonomy which are Synthesis and Evaluation. Hence by playing a full game, ProDec provides a full coverage of Bloom's taxonomy.

6 EXECUTION PHASE

The second step consists on executing the project created or selected in the previous phase. To support this project lifecycle phase, ProDec automatically generates a source code file with the equations of a discrete-event simulation model that simulates the project described in the first phase of the game. In addition, the file also contains the source code of the user interface for the specific simulation model generated. It is important to notice that this feature helps ProDec achieve a high level of flexibility given that the number of different projects that can be simulated is unlimited.

To achieve this flexibility, we have performed a reverse engineering process. We have analyzed how the simulation software used implements the elements of a discrete-event simulation model, and then, we have generalized the procedure with the aim of building the simulation models in an automatic way. Working like this, ProDec is able of simulate any project plan the players create overcoming the lack of flexibility found in other proposals previously described.

Once the source code of the simulation model is generated, the simulation model is launched and the players start managing the project. During the simulation of the project execution, the game shows the players several screens where the progress of the project is presented and different actions are provided to control the project. In this phase, learners practice two main concepts. On the one hand, they put into practice their knowledge about the Earned Value Analysis (EVA) for monitoring the progress of the project.

On the other hand, the learners practice their decision-making skills by correcting the potential

deviations of the progress of the project from the goal of ending the project within the time and budget. If a corrective action is needed, players select it from a set of actions such as hire a team member or reorganize the project team.

In this phase of the game, players need to analyze the information presented in different screens for controlling and monitoring the progress of the execution of the project. For this, learners use their knowledge about software project management to understand the information about the progress of the project provided by the game. They also have to analyze, create, discuss, argue, evaluate and negotiate within their team to agree on the best decisions in each moment. In this stage, the game also presents a full coverage of the six levels of Bloom's taxonomy.

7 END PHASE

The last phase consists on the players' assessment. By using the information that ProDec has been recording during the game play and the assessment criteria established by the instructor, ProDec generates an assessment report of the learners describing their level of achievement. These information records come from several sources within the game such as the project plan with the initial estimates, the project monitoring data and the kind of decisions that the players made during the play.

The assessment criteria are provided by the instructor in the form of a rubric. An assessment criterion links the information recorded in the rubric with the information recorded during the game. By using a labelling system the labels describing the skills of an assessment criterion are matched with the records of the game that contain the information needed to assess such criterion.

As a consequence, ProDec is able to perform the learners' assessment by analyzing the information recorded during the game and applying the assessment criteria set by the instructor, concluding with the generation of a detailed assessment report that describes the skills acquired by the players.

The generated report allows players to analyze the course of the game play with the aim of learning about their experience. At this point, learners can learn from their own mistakes and can analyze the events occurred along the game to get new knowledge and generate new ideas for future plays.

In this stage, the highest levels of Bloom's taxonomy are also covered.

8 CONCLUSIONS

In Section 2, we presented a comparison between several serious games attending to a set of features considered to be of importance in a game for software project management training. We showed that none of the analyzed games fully satisfies all the mentioned features. In this paper, we proposed ProDec, a simulation-based serious game for software project management, which aims to teach, assess and motivate learners in learning and practicing the principles of software project management as well as improving some important soft skills, like project leadership, in a risk-free environment provided by project. Thus, learners can experiment their abilities with real life scenarios without costs or risks.

ProDec has been developed with the aim of having a tool that satisfies all the criteria that other serious games in the field do not cover. Therefore, ProDec is a serious game that:

a) Opposite to the majority of the serious games in the field, which are focused on a specific stage of the project lifecycle or the training of a specific technique, allows learning along the whole project lifecycle. To achieve this, players can start a software project plan from scratch, improve their skills in decision-making, analyze the log of games played in the past, etc. ProDec offers a sufficient coverage of all the levels of Bloom's taxonomy, from the Onset stage until the End stage of the game lifecycle.
b) ProDec also allows trainers to assess the learners' skills through the information reports that the tool generate during the game play. Besides, at the end of a play, ProDec automatically provides an assessment report with detailed information about the events occurred in the game. Learners can then analyze their game experience to achieve new knowledge and improve their abilities in software project management.

Finally, ProDec is a simulation-based serious game. This means that it uses simulation to execute the software projects, planned by learners and instructors, so that players can practice decision-making in the scope of project controlling and monitoring processes. As a remarkable feature, ProDec generates in real time the source code of the specific discrete-event simulation model for the created or selected project together with the appropriate user interface. This feature makes ProDec a completely flexible tool regarding the

unlimited project scenarios that can be simulated, and, therefore, played.

To sum up, along this paper we have shown how ProDec is a flexible tool, covers the levels of Bloom's taxonomy, allows players to take contact with all the lifecycle of a project and helps professors to assess learner's skills, overcoming the weaknesses identified in current proposals, answering to our first research question. Besides, we have analyzed how ProDec reaches the six levels of Bloom's taxonomy by analyzing the player's behavior ProDec demands at its different lifecycle stages, answering, this way, our second research question.

Our aim is to create a tool to support the effective practical training of the processes of software project management. We believe that this kind of tool is needed to prepare the new practitioners for their professional life in the best possible way. For this reason, we are currently working to improve the features of ProDec to increase learners' soft skills such as project leadership, motivation, engagement and competitiveness. To achieve these goals, we are also studying the benefits of integrating our game with social networks and adopting gamification strategies for enriching the learning process.

Moreover during the first semester of 2014 the tool is being evaluated with real learners. The lessons learnt from this process will be used to measure the effectiveness of the learning and assessment processes when using ProDec and to improve the game.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Science and Technology with ERDF funds under grant TIN2010-20057-C03-03.

REFERENCES

ACM/IEEE-CS, 2013. Computer Science Curricula 2013.
Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. December 20, 2013 The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer.

Albrecht, A. J., 1979. *Measuring application development productivity*. Proceedings of the joint SHARE, GUIDE, and IBM Application Development Symposium, Monterey, California, IBM Corporation, pp. 83-92.

Anderson, L. W., and Krathwohl, D. R., 2001. *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives.* Longman, New York.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., and Krathwohl, D. R., 1956. *Taxonomy of educational objectives: the classification of educational goals*. Handbook I: Cognitive Domain. New York, Longmans, Green.

Calderón, A., and Ruiz, M., 2013. *ProDec: a serious game for software project management training.* The 8th International Conference on Software Engineering Advances. ICSEA, 565-570 Venice. Italy.

Caulfield, C., Veal, D., and Maj, S.P., 2011. *Teaching software engineering project management-A novel approach for software engineering programs*. Modern Applied Science, 5(5), 87-104.

Caulfield, C., Xia, J. C., Veal, D., and Maj, S. P., 2011. *A systematic survey of games used for software engineering education.* Modern Applied Science, 5(6), 28-43.

Cattell, R. B., Eber, H. W., and Tatsuoka, M. M., 1988. *Handbook for the sixteen personality factor questionnaire (16PF)*. Institute for Personality and Ability Testing.

Drappa, A., and Ludewig, J., 2000. *Simulation in software engineering training.* Proceedings of the 22nd International Conference on Software Engineering, pp. 199-208. ACM.

Games by Rocs, 2013. *The project management game.* http://thatpmgame.com/ [retrieved: December, 2013].

Ibrahim, I., 2011. *Teaching project management to IT students: methods and approach*. 2nd International Conference on Education and Management Technology. IPEDR. Vol. 13, 185-191. IACSIT Press.

Krathwohl, D. R., 2002. *A revision of Bloom's taxonomy: An overview.* Theory into practice. 41(4), 212-218.

Moder, Joseph J., 1983. *Project management with CPM, PERT, and precedence diagramming*. 3rd ed. New York: Van Nostrand Reinhold.

Navarro, E. O., and Van Der Hoek, A., 2004. *SimSE: An interactive simulation game for software engineering education*. Proceedings of the Seventh IASTED, International Conference on Computers and Advanced Technology in Education.

Project Management Institute, 2013. *A guide to the Project Management Body of Knowledge (PMBOK® Guide)*. Fifth Edition.

Project Management Institute, 2005. *Practice standard for earned value management*.

Von Wangenheim, C. G., Savi, R., and Borgatto, A. F., 2012. *DELIVER! – An educational game for teaching earned value management in computing courses.* Information and Software Technology, 54(3), 286-298.