A Case of System Dynamics Education in Software Engineering Courses

Ángel García-Crespo, Ricardo Colomo-Palacios, Juan Miguel Gómez-Berbís, Fernando Paniagua-Martín

Abstract—This paper presents a teaching initiative introduced in the Carlos III University of Madrid in which System Dynamics has been applied for teaching project management in the context of Computer Science grade. The initiative was designed in order to achieve two objectives: effectively deliver the lecturing materials relevant to simulation to students and, using the experimental models constructed, emphasize the human aspects of Software Engineering to students, and thus develop their competences as Software Development project managers.

Index Terms—System Dynamics, Simulation Education, Project Management, Software Engineering, Human Factors.

I. INTRODUCTION

A project is a temporary endeavor undertaken to create a unique product, service or result [1]. Project Management is a common task in different engineering fields, and, consequently, teaching Project Management is part of several engineering degrees.

A software development project is the kind of project engineers often face. Boehm [2] highlighted the importance of the management of software development projects by affirming that poor management can increase the software’s costs faster than any other factor. Additionally, recent investigations prove that it is advisable to use interactive means and System Dynamics [3] for the engineer’s training. In particular, the large body of literature available related to a software engineers’ education is relevant to the current research. Software development is tackled in diverse branches of engineering as a core subject. Many universities educate Computer Engineering, Software Engineering or Information Systems students in Software Project Management by means of conference type lessons that do not provide practical education, which, for these kinds of projects, is as necessary as difficult to tackle. Simulation is halfway between the theoretical class and practical experience. Simulation is adopted by diverse professional areas, such as the military, entertainment or civil engineering. Software Project Management is highly complex, comparable to airplane’s piloting in difficult conditions [3].

Therefore, taking into account, on one hand, the lack of practical education for project managers and, on the other hand, the benefits of using simulation tools in engineering education, we have carried out an initiative to handle the problem of the lack of practical training in project management in engineering education. In the framework of an optional subject in the 5th and last year of the Bachelor in Computer Science degree at the Carlos III University of Madrid, with particular regard to the area of human resources management, a learning experience using System Dynamics for the simulation of situations with regard to the role of project manager has been conducted.

This subject can be viewed in the context of the Software Engineering curriculum as part of a body of teaching subjects which constitute the curriculum, which range from programming, software design and specification of requirements, to the fundamental concepts of project management. Given this setting, the initiative within the Project Management subject provides a reinforcement mechanism for teaching the managerial skills a software engineer requires for developing software products.

The paper, after providing a theoretical introduction to System Dynamics in Section 2, describes the framework of the initiative in Section 3, displays the teaching results in Section 4, and concludes in Section 5 with main conclusions and improvements to the model and its application.

II. SYSTEM DYNAMICS EDUCATION

System Dynamics originated in the late fifties when M.I.T.’s Jay Forrester, commissioned by General Electric, studied the use of a plant located in Kentucky. From his analysis the article “Industrial Dynamics a Major Breakthrough for Decision Makers” [4] emerged, which was afterwards extended in the work “Industrial Dynamics” [5]. System Dynamics’ purpose is to ease the understanding of the relation between the system’s behaviour over time and its structure, politics and decision rules [6]. The initial application of SD was to the problems arising in the arena of corporate management: unstable production/employment cycles and declining market share, for
example. Industrial dynamics, a term coined by Forrester, was transformed into System Dynamics once it began to be used in different domains that constituted a continuous system: urban development, sociology and economics. Currently, System Dynamics has numerous applications in scientific, social and business fields [7], for which the emergence of visual software packages has turned out to be crucial (VenSim, iThink, Powersim, Studio...). These software packages simplify the models’ encoding tasks using graphic elements that substitute the pseudo-FORTRAN sentences of the first tools (Dynamo, SimScript, GPSS...).

System Dynamics’ application in the management of software development projects has its origin in the use for general management of System Dynamics, initiated by Roberts [8]. Taking as a reference Roberts’ research, the first application of System Dynamics to the software development process is the model developed by Abdel-Hamid and Madnick [9]. This model depicts the management elements for a small or medium sized software development project, using the cascade life cycle model. It has been useful as a base for numerous subsequent models, which try to complement the general model by means of its application in a particular organization [10], [11], [12] or to analyze and solve specific management problems [13], [14], [15], [16].

The “Simplified Model of Software Project Dynamics” [17] should be specifically mentioned. The model, which has been developed in Spain by researchers from University of Seville, is obtained by simplification of Abdel-Hamid and Madnick’s model [9], reaching a 51% reduction of the different factors. Applied in this way, given its simplicity, it is especially useful when it comes to showing new management staff for software development projects the basic behaviour of a project with these characteristics. The Software Engineering Body of Knowledge, SWEBOK [18], has recognized System Dynamics’ importance for software development projects. This crucial work includes a reference to [19] in the “Software Engineering Management” knowledge area.

The potential of simulation models for the training of managers has long been recognized [20]. The environments of the type “flight simulator” force managers to face up to real situations. This allows managers to undergo almost real experiences without the risks introduced by the real world. The specific use for training in software projects is backed up by fully documented experiences. Diverse Australian local consultancy firms currently use a simulator of the type “War Game” for software development, which has been supported by the Australian Computer Society since 1994, in order to train its project managers [21].

In higher education domain, the presence of System Dynamics engineering studies has been documented [22] and, specifically in Software Engineering fields, diverse authors have successfully implemented System Dynamics’ use for university education in Software Engineering both in Europe [20] and in the United States [23], [3]. However, the proven value of modelling for the training of managers of software development projects has not been enough to be included in Software Engineering 2004 [24]. This volume is part of Computing Curricula ACM and IEEE educational guidelines, and, although the Software Engineering Body of Knowledge [18] introduces a reference to System Dynamics, Software Engineering 2004 does not contain any topic, or even any reference to this fully proven topic for project management education. System dynamics’ teaching in university environments has a limited range. The teaching of system dynamics covers a range of domains, both engineering and those related to management of natural or biological resources, and the simulation of human behaviour.

System Dynamics is generally applied to Computer Engineering education to illustrate a continuous system’s simulation by means of Forrester and Causal diagrams. Some examples of such type of teaching are, within Spanish cases, the lecturing initiatives carried out in Málaga University, Huelva University, Alicante University or Almería University, among other Spanish universities.

Software development courses are present in numerous engineering programmes, and are mandatory in Computer Science grades. Diverse initiatives since the middle 80s have aimed to emphasize Software Engineering’s importance in engineering programmes [25], [26]. The presence of subjects related to the Software Engineering discipline in Computer Science, Computer Engineering, Information Systems, Information Technology courses and even in Software Engineering programmes itself, is regulated by recent international curriculum initiatives backed by IEEE and ACM. Software Engineering Management, which is the subject of this article, is one of the most important and intensively covered knowledge areas in the whole of Computing Curricula effort.

III. ACADEMIC FRAMEWORK OF APPLICATION

“Software Production Control” is a four-monthly optional course of the fifth (final) year, constituting 6 credits of the Computer Engineer degree in the Carlos III University of Madrid. The subject’s purpose is to provide the students with mechanisms for excellent performance in a Project Manager role by means of identification, definition and evaluation of the crucial parameters for the management of a software project. Both the qualitative and quantitative parameters are extracted from global concepts of the standardization rules and from maturity models for software development projects. The purpose of their application is to provide the manager with a set of factors that reflect the project’s state and that allow him to control its progress.

The subject has an eminent practical nature. Once the theoretical framework has been delivered in detail, the basic concepts of System Dynamics and of specific models for software development projects are tackled: Abdel Hamid’s models [9] and the Simplified Model of Software Project Dynamics [17]. Pestel, author of “Mankind at the Turning Point” in 1974 [27], stated that the System Dynamics technique can be learned in less than twenty-four hours. Considering this
maxim, a plan has been designed where the student must develop using VENSIM, starting with the development of simplified examples related to the dynamics of software development projects, and subsequently, its partial control system for software development projects based on a subset of indicators, previously established.

The final mark depends on diverse factors, particularly the model’s implementation using the VENSIM tool and the report, which proposes a solution to a real problem or situation that the lecturer describes, which diverse factors of the model affect. The numeric criteria of the different factors that determine the student’s mark are the following:

- Modelling in VENSIM (65%)
- Report about modelling (25%)
- Oral Presentation (10%)

The different groups of students have developed different models dealing with the managerial problems of software development projects, paying special attention to productivity factors, such as teamwork productivity, motivation, the project member’s capacities and knowledge, and the distinction between working hours and productive hours. The diverse aspects that have been simulated using the VENSIM tool are depicted in Figure 1, being extracted from different publications related to the management of software development projects [28], [29] and from the students’ experience in software development projects during different stages of the degree, or professional experience they may have had. In particular, lecturers are careful to assign at least one student with employment experience to each of the groups when determining the composition of the groups. The objective is to assign members according to the relative motivation which the lecturer considers would be present in an enterprise, influenced by the working atmosphere. Lecturers suggested to each student group to carry out an informal survey between a subset of workers of their enterprises in order to establish the different values of the constants in a real environment, based on the results of the survey. The variable’s initial values which represent the working climate are the result of the surveys conducted by the students in their working environment.

The task of the students is to accumulate data which represents distinct components of the model, for example, establishing the average productivity of a working team based on historical data available to the organization, as well as modeling this data using VENSIM. Additionally, once the model has been developed, the students carry out simulations of real environments, benefiting from the learning gained by examining the effects of modifications to the model, and testing the viability of the model in a real scenario. The results of these tests are described in the subsequent sections.

As has been previously pointed out, the focus on the aspects related to productivity in the 2006/07 academic year and the interconnection with factors such as recruitment or staff’s mobility assigned to the project have not impacted on the model’s growth, and will be dealt with in future revisions. The number of students registered in the subject made it possible to create two working groups devoted to the model’s development and allowed a supervision and a student-lecturer collaboration, that helped both the model’s quality and a reduced absence rate during the subject’s teaching.

The experimentation with the variations in the different factors impacting the management of a software development project means an experience that represents for the students their first situational test as Project Leader, and confronts them with situations which would not be possible to represent in another setting.

IV. PRODUCTIVITY AND SOCIOLABORAL MODEL IN SOFTWARE PROJECT MANAGEMENT

In the development of the model, the afore mentioned Abdel Hamid and Madnick [9], and Simplified Model of Software Project Dynamics [17] have been taken for reference, simplifying its complexity if possible, but including in more depth productivity and sociolaboral aspects. These aspects are related to motivation, working environment, and both personal and the team’s productivity, as shown in Figure 1. All of them are relevant for the training of project managers, being necessary for the development of “Soft Skills” in managers. The previous models, which try to model the set of aspects that comprise a software development project, due to their general focus, are much less detailed in aspects such as the characterization of staff (where one and two categories of staff are established respectively, the categories reflect the experience of each staff member), or do not consider the sociolaboral aspects such as motivation or working climate. The work of the students was focused on the modeling of productivity and sociolaboral aspects, based on the data previously obtained.
Figure 1 Vensim model
The method by which the students were instructed to construct the model is considered part of the educational strategy. In order to realize this strategy, the students were told to work alone on one particular part of the model, subsequently collaborating with their group in order to report on their progress and combine their efforts. The part of the model which each student designs is examined by the lecturer. Once all of the different versions of the model have been developed, a committee is established to integrate all of the sub-models.

The system developed is focused on the calculation of productivity. This productivity is determined by the group’s productivity (where, among other factors, are motivation and hygienic factors, the group’s composition and skills) and the availability of productive hours. This latter aspect is influenced by diverse factors regarding working policy and personal factors of the individual. The control of fluctuation and variations in productivity represents the challenge of economic and human resources’ management for the student.

A typical case the model considers is the decrease of any of the motivational factors, typically salary, working conditions or professional degree, which make the productive hours go down, alongside a negative effect on productivity, and consequently, the costs. Another contingency the student must model, face and manage is the choice between a short term loss in productivity due to training in working time, and the medium term benefits gained from the improvement of capacities and, therefore, of the productivity. A final circumstance that is considered important to be taken into account is the necessity to rely on expert staff, which are more expensive, while the required hours/manpower can be carried out by another type of workers, with lower cost, but with negative impact on the productivity.

The model (Figure 1) portrays four variables that determine the system’s behaviour:

- Remaining work
- Completed Work
- Staff
- Cost

Variables, according to the values of different constants, enable the determination of a project’s state, and their behaviour can be simulated by the student’s actions using the simulation tools, in order to calculate costs and model staff modifications. The actions are performed under the premise that the deadline is fixed.

The model has 88 additional variables, among the so-called auxiliary variables, whose values do not depend on their values in previous states of the simulation, and constants, values that cannot be modified during the execution of a conventional simulation. In the latter case, and with the purpose of being used as a “flight simulator”, these types of objects are used to experiment with their values in different situations.

There are numerous feedback loops, affecting mainly the Workforce level variable. The feedbacks are designed with the purpose of making modifications to the decrease in the “Remaining work” level and therefore also in the staff “Needs”, and thus in the “Workforce” level. Other components included in the feedback loops are the elements regarding productivity, which, as has been previously pointed out, is affected by the composition of the work force.

With the aim of testing the validity of the model, two types of simulations have been performed. The objective of the first simulation is to use the model to determine the difference in performance when comparing a working day of 35 hours with one of 40 hours. It is possible to test the model on a 40 hour...
week initially, as well as examining performance when simulating 35 hours per week (the working day demanded by trade unions). The model’s behavior due to the variation of the productivity and of the team’s composition is illustrated in Figures 2, 3, and 4 through the examination of “Cost”, “Workforce” and “Remaining Work” level variables.

In the mentioned figures what can be seen is, firstly, the logical increase in the work force, and, consequently, in the project’s costs, which increase by 40%, because of the parallel decrease in productivity alongside the team’s increase. Secondly, owing to the time limitation of the project, the almost invariable level of the “Remaining Work” variable can be seen in both simulations.

The objective of the second simulation was to analyze the reliability of the model in comparison with the performance of other parametric models. The COCOMO II model [30] was taken as a reference. COCOMO (COnstructive Cost Model) is the most developed algorithmic model since it was first created. The most recent version of COCOMO available, COCOMO II 2000, was used by the students for the fitting of distinct factors, and subsequently compared with previous models. Due to the appropriateness of the model developed for human capital, it was proposed by Colomo-Palacios, Ruano-Mayoral, Gómez-Berbís and García-Crespo [31] to use the COCOMO II 2000 model, however, with some minor modifications to the COCOMO II model to improve its performance. With respect to the COCOMO II 2000 model with modifications suggested in [31], it fits with a variability of less than 10% in effort and 8% in time period estimations.

I. STUDENTS PERFORMANCE RESULTS

The construction of the model has realised a double objective: it can be used as a learning tool for developing the technical skills required for performing simulations, and as a testing platform to monitor the variation in distinct elements of the model in a software development environment. Therefore, the results obtained can be discussed in the context of the fulfilment of two distinct objectives.

In relation to the effects of the educational strategy on student learning, the development of the model in the teaching environment exhibited specific results in the values of particular parameters of the class as a whole. The marks of the students increased by 14% in comparison with those results obtained in the same subject in the previous year, during which the students studied theoretical models in the literature, and carried out partial implementation of models.

Another outcome which supports the previous result is a 3% decrease in the level of absenteeism of the class compared to the previous year. A final effect which was evident, which has not been empirically tested and may be due to qualitative causal relations, is a 17% increase in the number of students who registered for the course in the subsequent year, compared with the previous year.

A further qualitative result could be observed regarding the simulation of the variations in different elements which comprise the model. As previously discussed, the students made the important realization of the similarity between this tool and other established tools such as COCOMO II, as well as similar applications of the tools. This result was determined by personal feedback to the lecturer from the students. This result contributed to, on the one hand, increasing student’s confidence, and on the other hand, the realization of the students of the validity and realibility of such tools as valid instruments in professional environments. The possibilities for future research which the results initiated will be discussed in a subsequent section.

II. CONCLUSIONS

The understanding of the different parameters a project manager must know and control, and their interrelation, was easily delivered to and understood by the students. However, in particular stages of development, students saw the tool as an aim, and not as a means for analysing model factors. However, once the model was constructed and its simulation carried out, the student’s interest increased, and they could see how the variation of any of the factors influenced situations which were previously unexpected, which they had to manage.

The focus on personal aspects for the simulation’s performance constitutes a new development in the use of these types of models. It also exhibits an opportunity for such models to be used in other engineering projects, not necessarily related to software development, but with which they have aspects in common, such as productivity management, deadline, cost and the composition of the working team.

In particular, the model’s use has meant for the students their first immersion in the complex project management world, encouraging participation and increased interest, and a decrease in absenteeism. The “Action Learning” capacities of the simulators are regarded as having a great impact in the competent training of Engineers, who need real management capacities and real experience in planning and project monitoring.

Finally, it is essential to discuss the relevance of this initiative in the context of the Computer Science degree. The minor
impact of simulation in Computer Science professional fields is in stark contrast with the apparent relevance which curriculum design in such disciplines give to simulation and System Dynamics. Creating professionals with knowledge and simulation skills in the Software Development domain improves the abilities of professionals in the labor market. Additionally, the formation of these knowledge workers raises a challenge for universities, whose task is to supply human capital markets. This has a double outcome: it satisfies the requirements for skilled workers, but it also stimulates innovation and application of knowledge which contributes to the advancement of various disciplines, including Computer Science, Software Engineering, and Information Systems.

Regarding future research, in the context of the development of System Dynamics in the course, the refinement of the model is proposed, with the objective of testing its applicability in more complete environments. In the pedagogical context, the effect of the utilization of System Dynamics on student performance should be empirically measured.

The improvement in the marks of the students discussed above instigated a possibility for future research, which is to further examine the values of the marks of each of the classes of students and apply established statistical metrics, such as a T-test for two independent groups of observations, in order to determine whether or not the values represent a statistically significant difference between the two years. A similar statistical analysis could be conducted to measure other parameters to determine statistically significant differences, for example, between the marks based on the gender of the students. An appropriate method for such an analysis is one way analysis of variance (ANOVA). Using this method, a number of other opportunities for further research emerge, such as examining the effect of different sources of variation, for example, the age of the students.

As well as the parameters mentioned above, specific qualitative factors of the students could be measured with appropriate metrics such as a Likert scale, for example, student satisfaction in the context of particular skills which are part of the teaching objectives of the course. These could include factors such as team work competencies and understanding of the teaching material delivered.

REFERENCES


Angel García-Crespo is the Head of the SofLab Group at the Computer Science Department in the Universidad Carlos III de Madrid and the Head of the Institute for promotion of Innovation Pedro Juan de Lastanosa. He holds a PhD in Industrial Engineering from the Universidad Politécnica de Madrid (Award from the Instituto J.A. Artigas to the best thesis) and received an Executive MBA from the Instituto de Empresa. Professor Garcia-Crespo has led and actively contributed to large European Projects of the FP V and VI, and also in many business cooperations. He is the author of more than a hundred publications.
publications in conferences, journals and books, both Spanish and international.

**Ricardo Colomo-Palacios** has been a Faculty Member of the Computer Science Department at Carlos III Technical University of Madrid since 2002. His research interests include Software Process Improvement, Software Project Management and Information Systems. He received his PhD in Computer Science from the Universidad Politécnica of Madrid (2005). He also holds a MBA from the Instituto de Empresa (2002). He has been working as software engineer, project manager and software engineering consultant in several companies including Spanish IT leader INDRA.

**Juan Miguel Gomez-Berbís** is a Visiting Professor at the Computer Science Department of the Universidad Carlos III de Madrid. He holds a PhD in Computer Science from the Digital Enterprise Research Institute (DERI) at the National University of Ireland, Galway and received his MSc in Telecommunications Engineering from the Universidad Politécnica de Madrid (UPM). He was involved in several EU FP V and VI research projects and was a member of the Semantic Web Services Initiative (SWSI). His research interests include semantic web, semantic web services, business process modelling, b2b integration and, recently, bioinformatics.

**Fernando Paniagua-Martín** has been a Faculty Member of the Computer Science Department at the Carlos III Technical University of Madrid since 2005. Currently, he is completing his PhD in Computer Science in the Universidad Carlos III de Madrid. He also holds a Master in Computer Science and Technology. He has been working as software engineer and project manager in several companies. His research interests include Software Engineering, Audio-visual accessibility on the Web, Web 2.0 and Computer Supported Cooperative Work.