Effective Software Project Management Education through Simulation Models: An Externally Replicated Experiment

D. Rodríguez[1], M. Satpathy[1], and D. Pfahl[2]

[1] Dept. of Computer Science, The University of Reading, Reading, RG6 6AY, UK
{d.rodriguezgarcia, m.satpathy}@rdg.ac.uk
[2] Fraunhofer Institute for Experimental Software Engineering (IESE)
Sauerwiesen 6, 67661, Kaiserslautern, Germany
pfahl@iese.fhg.de

Abstract. It is an undeniable fact that software project managers need reliable techniques and robust tool support to be able to exercise a fine control over the development process so that products can be delivered in time and within budget. Therefore, managers need to be trained so that they could learn and use new techniques and be aware of their possible impacts. In this context, effective learning is an issue. A small number of empirical studies have been carried out to study the impact of software engineering education. One such study is by Pfahl *et al* [11] in which they have performed a controlled experiment to evaluate the learning effectiveness of using a process simulation model for educating computer science students in software project management. The experimental group applied a Systems Dynamics simulation model while the control group used the COCOMO model as a predictive tool for project planning. The results indicated that students using the simulation model gain a better understanding about typical behaviour patterns of software development projects. Experiments need to be externally replicated to both verify and generalise original results. In this paper, we will discuss an externally replicated experiment in which we keep the design and the goal of the above experiment intact. We then analyse our results in relation to the original experiment and another externally replicated experiment, discussed in [12].

1 Introduction

The most important objective of software project management is to use the available resources effectively so that the tasks and subtasks of the development process are kept in schedule and within budget without sacrificing on product quality. Many interacting factors throughout the software life cycle can have impact over the cost and schedule of a project and the quality of the product. To monitor and control software development projects, management experience and knowledge on how to balance the various influential factors are required. To address these issues, process simulation techniques have been applied to the domain of Software Engineering (SE) during the last decade Abdel-Hamid and Madnick [1], etc. But although the potential of simulation models for the training of managers has long been recognised [6, 9],

F. Bomarius and H. Iida (Eds.): PROFES 2004, LNCS 3009, pp. 287–301, 2004.

very few experimental studies involving process simulation as a means for software project management education have been performed.

The results of these experiments indicate that a natural one-way causal thinking could be detrimental to the success of software managers. They must rather adopt a multi-causal or systems thinking. Moreover, they must be aware of (unexpected) feedback to their management decisions. A dynamic model like the Systems Dynamics not only present a white-box view of the influencing factors affecting a certain process attribute of interest (say, cost or effort) but also it often provides some feedback of the possible outcome [1]. A static model like COCOMO [2] only presents a black-box view of the system. These findings highlight the need for new learning and education strategies. The first strategic step for teaching software project management must already be included in the curriculum of students. University education must teach computer science and software-engineering students not only technology related skills but also a basic understanding of typical management phenomena occurring in industrial (and academic) software projects. Pfahl *et al* [11] have performed a controlled experiment that evaluates the effectiveness of using a process simulation model for university education in software project management. In their study, the experimental group applied a Systems Dynamics simulation model while the control group used the COCOMO model as a predictive tool for project planning. The results of the experiment indicate that students using the simulation model gain a better understanding about typical behaviour patterns of software development projects. The result of an experimental study is not usually extrapolated to all possible software environments, especially so when we deal with human subjects. Many uncontrollable sources of variation exist from one environment to another; therefore, more studies need to be conducted in a variety of environments. In addition, replicated studies can help the researchers to combine knowledge directly or via some form of meta-analysis. Since intervening factors and threats to validity can almost never be completely ruled out of a study, complementary studies also allow more robust conclusions to be drawn when related studies can address one another's weak points [14]. Replication of a study means repeating a study based on the design and results of a previous study, whose goal is either to verify or broaden the applicability of the results of the original study. A replication can be internal or external. In an internal replication, the original researchers perform the replication, whereas in an external replication, different researchers conduct the replication. A scientific hypothesis gains increasing acceptance when external replications arrive at the same conclusion.

In this paper, we discuss an external replication of the same experiment (of Pfahl *et al* [11]) which was performed at the University of Reading in England. Another externally replicated study has also been conducted at the University of Oulu, Finland [12]. We discuss here our findings and perform a meta-analysis over the results of the original experiment and the results of the two replications. The organization of the paper is as follows. Section 2 presents the experimental details of the study. Section 3 summarizes the results of the data analysis and a brief meta-analysis over the results of the three experiments. Section 4 discusses the various threats to the validity of the study. Finally, Section 5 concludes the paper.

2 Description of the Experiment

Ours is a replication study; therefore we have tried to keep the same intent and environment as of the original experiment. However, the original experiment suggested some modifications as to the timing of the experiment, and in our experiment we have followed the suggestions. The main objective of developing and applying a simulation-based training module has been to facilitate effective learning about certain topics of software project management for computer science students. This was done by providing a scenario-driven interactive single-learner environment that can be accessed through the internet by using a standard web-browser. The training module used in the study is composed of course material on project planning and control. The core element of the training module is a set of interrelated project management models, represented by a simulation model that was created by using the System Dynamics (SD) simulation modelling method [5]. This model simulates typical behaviour of software development projects.

In order to investigate the effectiveness of computer-based training in the field of software project management using a SD simulation model, a controlled experiment applying a pre-test-post-test control group design was conducted. The subjects who were willing to participate in the experiment had to pass two tests, one before the training session (pre-test) and one after the training session (post-test). The effectiveness of the training was then evaluated by comparing within-subject post-test to pre-test scores, and by comparing the scores between subjects in the experimental group, i.e. those who used the SD model, and subjects in the control group, i.e. those who used a conventional project planning model instead of the SD model. In the study, the well-known COCOMO model [2] was used by the control group since this model is used in many industrial software organisations.

The various possibilities of conducting a training session is described as a three-layered structure. The first layer defines the learning goal, i.e. software project management with focus on project planning and control. The second layer defines the type of project planning model used in the training session, i.e. COCOMO model versus SD simulation model. Finally, the third layer defines the learning mode as another dimension to characterise the training session, i.e. inclusion or exclusion of a web-based interactive role-play. The combination of the distinctions made in layers two and three yield four different treatments. Our empirical investigations compare the effectiveness of two of them: T_A (Group A: SD model-based learning with web-based interactive role-play scenario) verses T_B (Group B: standard COCOMO-based learning without web-based interactive role-play). The following dimensions were used to characterise "effectiveness" of the training session:

1. Interest in software project management issues ("Interest").
2. Knowledge about typical behaviour patterns of software development projects ("Knowledge").
3. Understanding of "simple" project dynamics ("Understand simple").
4. Understanding of "complex" project dynamics ("Understand complex").

In the study, these four dimensions were represented respectively by dependent variables Y.1 to Y.4.

2.1 Experimental Hypotheses

The two hypotheses of the experiment were stated as follows:

1. There is a positive learning effect in both groups (A = experimental group, B = control group). Using the notations in Table 1, this can be formulated as:
   - $score_{post}$ (Y.i; A) > $score_{pre}$ (Y.i; A), for i = 1…4
   - $score_{post}$ (Y.i; B) > $score_{pre}$ (Y.i; B), for i = 1…4
2. The learning effect in group A is higher than in group B, either with regard to the performance improvement between pre-test and post-test (relative learning effect), or with regard to post-test performance (absolute learning effect). The absolute learning effect is of interest because it may indicate an upper bound of the possible correct answers depending on the type of training (A or B). This expectation can be formulated as follows:
   - $score_{diff}(Y.i; A) > score_{diff}(Y.i; B)$, *for i = 1, …, 4*
   - $score_{post}(Y.i; A) > score_{post}(Y.i; B)$, *for i = 1, …, 4*

Table 1. Terms and definitions of the hypotheses

Term	*Definition*
$score_{pre}$ (Y.i; X)	Pre-test scores for Y.i (i = 1,…, 4) of subjects in group X (X = A or B).
$score_{post}$ (Y.i; X)	Post-test scores for Y.i (i = 1,…, 4) of subjects in group X (X = A or B).
$score_{diff}$ (Y.i; X)	Difference scores for Y.i (i = 1,…, 4) of subjects in group X (X = A or B). $score_{diff}$ (Y.i; X) = $score_{post}$ (Y.i; X) − $score_{pre}$ (Y.i; X)

Note that it is not expected that both relative and absolute learning effect will always occur simultaneously. This reflects on the fact that higher relative learning effects in group A compared to group B are less likely to occur when pre-test scores of group A are significantly higher than those of group B. Similarly, higher absolute learning effects in group A compared to group B are less likely to occur when pre-test scores of group A are significantly lower than those of group B. Standard significance testing was used to analyse expectations. The null hypotheses were stated as follows:

- $H_{0,1}$: There is no difference between pre-test scores and post-test scores within group A and group B, i.e.
  - $score_{pre}(Y.i; A) = score_{post}(Y.i; A)$ *and*
  - $score_{pre}(Y.i; B) = score_{post}(Y.i; B)$ *for all i = 1, …, 4.*
- $H_{0,2a}$: There is no difference in relative learning effectiveness between group A and group B, i.e.
  - $score_{diff}(Y.i; A) = score_{diff}(Y.i; B)$ *for all i = 1,…, 4.*
- $H_{0,2b}$: There is no difference in absolute learning effectiveness between group A and group B, i.e.
  - $score_{post}(Y.i; A) = score_{post}(Y.i; B)$ *for all i = 1,…, 4.*

2.2 Subjects

Our replication study (henceforth the Reading experiment) was conducted during a university term with 11 second year undergraduate students doing the software engineering module (out of a total of 180). One of the authors is the instructor of the module and he invited the students to participate in the experiment. A total of 30 students responded but only 11 turned up on the day of the experiment. Finally, the treatment was divided randomly among the students, just depending on which computer they selected. The personal characteristics of the subjects have been summarised in Table 2. We have also included the personal characteristics of the subjects in the original experiment and those of the first replication. The original experiment was conducted at the University of Kaiserslautern in Germany (henceforth KL experiment) and its first replication was performed at the University of Oulu in Finland (Oulu experiment). The subjects of the KL experiment were graduate computer science students enrolled in the advanced software engineering class. In the Oulu experiment, the subjects were graduate and post-graduate.

Table 2. Personal characteristics of the subjects

	KL students	***Oulu***	***Reading***
Average age [years]	27.0	31.3	23.20
Share of women	11 %	50 %	9%
Share of subjects majoring in Computer Science	100 %	67 %	82%
Preferred learning style(s):			
• Reading (with exercise)	89 %	33 %	18%
• wb-based training	11 %	8 %	33%
• in-class lecture (with exercise)	22 %	25 %	72%
• working group (with peers)	33 %	42 %	81%
Opinion about most effective learning style(s):	- not asked -		
• reading (with exercise)		25 %	18%
• web-based training		17 %	33%
• in-class lecture (with exercise)		33 %	72%
• working group (with peers)		67 %	81%

2.3 Treatments

The training sessions for both the groups was composed of the following four scenario blocks; they were the same as the original experiment [10]:

- Block 1 - PM Introduction: General introduction into the main tasks of software project managers and the typical problems they have to solve with regard to project planning and control.
- Block 2 - PM Role Play: Illustration of common project planning problems on the basis of an interactive case example in which the trainee takes over the role of a fictitious project manager.
- Block 3 - PM Planning Models: Presentation of basic models that help a project manager with planning tasks, namely a process map, and a predictive model for effort, schedule and quality.
- Block 4 - PM Application Examples: Explanation on how to apply the planning models on the basis of examples that are presented in the form of little exercises.

Treatment of the Experimental Group. The experimental group passed all scenario blocks. The SD model was used as the predictive model in scenario blocks 3 and 4. In addition, the SD model was integrated into the interactive role-play offered by scenario block 2. The SD model used in the training session consists of five interrelated sub-models [10].

Scenario block 2 (PM Role Play) has been designed to help the trainee understand the complex implications of a set of empirically derived principles that typically dominate software projects conducted according to the waterfall process model. The set of principles used in the block scenario was distilled from the top 10 list of software metric relationships published by Boehm [3]. In order to make the trainee understand the implications of these principles (and their combinations), a role-play is conducted in which the trainee takes the role of a project manager who has been assigned to a new development project. Several constraints are set, i.e. the size of the product and its quality requirements, the number of software developers available, and the project deadline. The first thing to do for the project manager (in order to familiarise with the SD simulation model) is to check whether the project deadline is feasible under the resource and quality constraints given. Running a simulation does this check. From the simulation results, the project manager learns that the deadline is much too short. Now, the scenario provides a set of actions that the project manager can take, each action associated with one of the principles and linked to one of the model parameters. Soon the project manager learns that his department head does not accept all of the proposed actions (e.g. reducing the product size or complexity). Depending on the action the project manager has chosen, additional options can be taken. Eventually, the project manager finds a way to meet the planned deadline, e.g. by introducing code and design inspections (one of the principles discussed by Boehm [3]). The role-play is arranged in a way that the project manager can only succeed when combining actions that relate to at least two of the principles of Boehm. At the end of the role-play, a short discussion of the different possible solutions is provided, explaining the advantages and disadvantages of each.

Treatment of the Control Group. The control group passed only scenario blocks 1, 3, and 4. The predictive model used in scenario blocks 3 and 4 was the intermediate COCOMO model [2].

Differences between Initial Experiment and Replications. Since almost all of the participants of the KL experiment stated that they did not have enough time for working through the materials, more time was reserved for the treatment during our replication study. While the initial experiment was conducted on two days with one week of time in between, the Reading experiment was conducted on one single day. The Oulu experiment also adopted the same changes.

2.4 Experimental Design

For evaluating the effectiveness of a training session using SD model simulation, a pre-test-post-test control group design was applied. This design involves random assignment of subjects to an experimental group (A) and a control group (B). The subjects of both groups pass a pre-test and a post-test. The pre-test measures the performance of the two groups before the treatment, and the post-test measures the

performance of the two groups after the treatment. By studying the differences between the post-test and pre-test scores of the experimental group and the control group, conclusions can be drawn with respect to the effect of the treatment (i.e. the independent variable of the experiment) on the dependent variable(s) under study.

2.5 Experimental Variables

During the experiment, data for three types of variables are collected. The dependent variables (Y.1 … Y.4) have been discussed earlier. The lone independent variable (X.1) is the type of treatment. Z1 (Personal background), Z2 (Time consumption/time need) and Z3 (Session evaluation) are the three variables that represent potentially disturbing factors. The conceptual model assumes that the independent variable and the disturbing factors affect the dependent variables [12].

Independent Variables. The independent variable X.1 can have two values: T_A, applied to the experimental group A, and T_B applied to the control group B. The difference between T_A and T_B is basically determined by two factors. The first factor is the training scenario according to which the course material is presented. The second factor is the planning model that is used to support software project management decision-making. With regard to the scenario, the main difference consists in the application of scenario block PM Role Play for treatment T_A. As a consequence of performing the scenario block PM Role Play, interaction of the trainee with the training module will be high whereas treatment T_B will only trigger low interaction of the trainee with the training module. With regard to the model that is used during the training session, treatment T_B exclusively relies on a black-box model providing point estimates, such as COCOMO. In contrast to this, use of SD model in T_A facilitates insights into behavioural aspects of software projects.

Dependent Variables. The dependent variables Y.1, Y.2, Y.3, and Y.4 are determined by analysing data collected through questionnaires that all subjects have to fill in, the first time during the pre-test, and the second time during the post-test. The value of each dependent variable will then be equal to the average score derived from the related questionnaire. The contents of the questionnaires are as follows:

- Y.1 (“Interest”): Questions about personal interest in learning more about software project management.
- Y.2 (“Knowledge”): Questions about typical performance patterns of software projects. These questions are based on the empirical findings and lessons learned summarised in Boehm’s top 10 list of software metric relations [3].
- Y.3 (“Understand simple”): Questions on project planning problems that require simple application of the provided PM models, addressing trade-off effects between no more than two model variables.
- Y.4 (“Understand complex”): Questions on project planning problems addressing trade-off effects between more than two variables, and questions on planning problems that may require re-planning due to alterations of project constraints (e.g. reduced manpower availability, shortened schedule, or changed requirements) during project performance.

Disturbing Factors. The disturbing factors remain the same as the original experiment. The contents of the respective questionnaires are as follows:

- Z.1: Questions about personal characteristics (age, gender), university education (number of terms, major, minor), practical software development experience, software project management, literature background and preferred learning style.
- Z.2: Questions on actual time consumption per scenario block, and on perceived time need.
- Z.3: Questions on personal judgement of the training session (subjective session evaluation).

Table 3. Time distribution for various stages during the experiments

	KL Experiment	***Oulu/ Reading***
Introduction to experiment	5'	5'
Background characteristics	5'	5'
Pre-test		
Interest	3'	5'
Knowledge about empirical patterns	5'	5'
Understanding of simple project dynamics	10'	10'
Understanding of complex project dynamics	12'	15'
Introduction to treatments	5'	5'
Random assignment of subjects to groups	5'	5'
Treatment	45'	80'
Post-test		
Interest	3'	5'
Knowledge about empirical patterns	5'	5'
Understanding of simple project dynamics	10'	10'
Understanding of complex project dynamics	12'	15'
Time need & subjective session evaluation	5'	10'
Total	130'	180'

2.6 Experimental Procedure

The duration of the phases in the KL and the Reading experiment are as they are in Table 3. The Oulu experiment also had similar plans as that of Reading. They are similar because both followed the changes suggested by the KL investigators. After a short introduction during which the purpose of the experiment and general organisational issues were explained, data on the background characteristics (variable Z.1) was collected. Then the pre-test was conducted and data on all dependent variables (Y.1 through Y.4) was collected, using questionnaires. Following the pre-test, a brief introduction into organisational issues related to the treatments was given. After that, the subjects were randomly assigned to either the experimental or control group. Then each group underwent its specific treatment. After having concluded their treatments, both groups passed the post-test using the same set of questionnaires as during the pre-test, thus providing data on the dependent variables for the second time. Finally, the subjects got the chance to evaluate the training session by filling in another questionnaire, providing data on variables Z.2 and Z.3. The time frames reserved for passing a certain step of the schedule was identical for the experimental and control groups. However, more time was reserved during the replication as

compared to the initial experiment. This was done in accordance with the recommendations of the original experiment [11]. Of the eleven students participating in the first replication, 6 were assigned randomly to the experimental group (A), and 5 to the control group (B).

2.7 Data Collection Procedure

The data collection procedure of our study remains same as the original study. We briefly discuss it here. The raw data for Y.1 to Y.4 were collected during pre-test and post-test with the help of questionnaires. Each answer in the questionnaire is mapped to the value range R = [0, 1] assuming equidistant distances between possible answers, i.e. "fully disagree" is encoded as "0", "disagree" as "0.25", "undecided" as "0.5", "agree" as "0.75", and "fully agree" as "1".

The raw data for disturbing factors were collected before pre-test (Z.1) and after post-test (Z.2 and Z.3). In order to determine the values of factor Z.1 ("Personal background") information on gender, age, number of terms studied, subjects studied (major and minor), personal experience with software development, and number of books read about software project management was collected. The values for factor Z.2 are normalised average scores reflecting the "time need" for reading and understanding of the scenario blocks 1, 3, and 4, for familiarisation with the supporting tools, and for filling in the post-test questionnaire. For group A, the variable Z.2' includes also scores related to scenario block 2. The values for factor Z.3 ("Session evaluation") are based on subjective measures reflecting the quality of the treatment.

2.8 Data Analysis Procedure

In a first step of the statistical analysis a t-test was used to investigate the effect of the independent variable X.1 on the dependent variables Y.1 to Y.4. For testing hypothesis $H_{0,1}$, a *one-way paired t-test* was used. For testing hypotheses $H_{0,2a}$ and $H_{0,2b}$, the *one-way t-test for independent samples* was used [13]. A prerequisite for applying the t-test is the assumption of normal distribution of the variables in the test samples. Checking for the normality assumption showed that no normal distribution of the variables in the test samples could be assumed. On the other hand, the outlier analysis showed that all data points lie within the range of ±2 standard deviations around the samples' means,.

Researchers should perform a power analysis [4] before conducting a study to ensure the experimental design will find a statistically significant effect if one exists. The power of a statistical test is dependent on three different components: significance level α, the size of the effect being investigated, and the number of subjects. Low power will have to be considered when interpreting non-significant results. Usually, the commonly accepted practice is to set $\alpha = 0.05$. Since sample sizes were rather small in the initial experiment and in our replication, and no sufficiently stable effect sizes from previous empirical studies were known, it was decided to set $\alpha = 0.1$. This was also the case with the Oulu experiment. *Effect size* is expressed as the difference

between the means of the two samples divided by the root mean square of the variances of the two samples [13]. For this exploratory study, effects where $\gamma \geq 0.5$ are considered to be of practical significance. This decision was made on the basis of the effect size indices proposed by Cohen [4].

3 Experimental Results

Data was collected from 11 subjects. The column "Pre-test scores" of Table 4 shows the calculated values for mean, median, and standard deviation of the raw data collected during the Reading experiment. The column "Post-test scores" shows the calculated values for mean, median, and standard deviation of the raw data collected during the post-test. The column "Difference scores" shows the calculated values for mean, median, and standard deviation of the differences between post-test and pre-test scores. Possible reasons for unexpected outcomes are discussed in a later Section.

Table 4. Scores of dependent variables

	Pre-test scores				***Post-test scores***				***Difference scores***			
Group A (6 subj.)	*Y.1*	*Y. 2*	*Y. 3*	*Y. 4*	*Y.1*	*Y. 2*	*Y. 3*	*Y. 4*	*Y.1*	*Y.2*	*Y. 3*	*Y. 4*
Mean($score_{pre}$(A))	0.63	0.40	0.33	0.22	0.68	0.80	0.81	0.61	0.05	0.40	0.48	0.39
Median($score_{pre}$ (A))	0.55	0.40	0.29	0.17	0.70	0.80	0.79	0.58	0.05	0.40	0.57	0.42
Stdev($score_{pre}$ (A))	0.22	0.18	0.20	0.23	0.22	0.13	0.12	0.14	0.10	0.18	0.27	0.33
Group B (5 subj.)	*Y.1*	*Y. 2*	*Y. 3*	*Y. 4*	*Y.1*	*Y. 2*	*Y. 3*	*Y. 4*	*Y.1*	*Y.2*	*Y. 3*	*Y. 4*
Mean($score_{pre}$ (B))	0.69	0.52	0.29	0.43	0.75	0.68	0.60	0.53	0.06	0.16	0.31	0.10
Median($score_{pre}$(B))	0.70	0.60	0.29	0.33	0.70	0.80	0.57	0.67	0.05	0.00	0.43	0.00
Stdev($score_{pre}$ (B))	0.17	0.23	0.14	0.22	0.16	0.18	0.12	0.22	0.07	0.26	0.23	0.30

Table 5 shows the calculated values for mean, median, and standard deviation of the raw data collected for the disturbing factors. In the initial experiment, there could be observed a difference between students in the experimental group (A) and the control group (B) regarding experience with software development (Z.1). This difference was neither observed in the Reading experiment nor in the Oulu experiment.

Table 5. Scores of Disturbing factors

Group A	***Z.1***	***Z.2***	***$Z.2_{B2}$***	***Z.3***	***$Z.3_{B2}$***
Mean $_{df}$	0.36	0.17	0.14	0.51	0.59
Median $_{df}$	0.33	0.00	0.00	0.50	0.58
StDev $_{df}$	0.06	0.27	0.22	0.17	0.10
Group B	***Z.1***	***Z.2***		***Z.3***	
Mean $_{df}$	0.3	0.05		0.65	
Median $_{df}$	0.33	0.00		0.63	
StDev $_{df}$	0.07	0.11		0.14	

3.1 Hypothesis $H_{0,1}$

Table 6 shows the results of testing hypothesis $H_{0,1}$ using a *one-way tailed paired t-test* is used to compare the means of the pre-test and post-test scores within each

group (A and B). Column one represents the dependent variable, column two the effect size, column three the degrees of freedom, column four the t-value of the study, column five the critical value for $\alpha = 0.10$ (the t-value has to exceed the critical value to be statistically significant), and column six provides the associated p value. By looking into the fourth and fifth columns of Table 6, we can check that group A achieved significant results for dependent variables Y.2, Y.3 and Y.4, and group B for dependent variables Y.1 and Y.3. Therefore the null hypothesis $H_{0,1}$ can be rejected for these cases at $\alpha = 0.10$. It is to note that for group A, the dependent variable Y.2 support the direction of the expected positive learning effect in both groups, however without showing an effect size of practical significance. In addition, for group B, values for dependent variables Y.2 and Y. 4 also support the direction of the expected positive learning effect, with and without practical significance respectively. Our analysis corroborates the result of the KL and Oulu experiments as regards to the variables Y.2 and Y.3 for group A, and for Y.3 in group B.

Table 6. Result for post-test vs. pre-test

Group A (6 Subjects)					
Variable		***Df***	***t-value***	***Crit t $_{0.90}$***	***p-value***
Y.1	0.48	5	1.17	1.48	0.15
Y.2	2.24	5	5.48	1.48	0.00
Y.3	1.79	5	4.39	1.48	0.00
Y.4	1.19	5	2.91	1.48	0.02
Group B (5 Subjects)					
Variable		***Df***	***t-value***	***Crit t $_{0.90}$***	***p-value***
Y.1	0.92	4	2.06	1.53	0.05
Y.2	0.61	4	1.37	1.53	0.12
Y.3	1.34	4	2.99	1.53	0.02
Y.4	0.33	4	0.74	1.53	0.25

3.2 Hypothesis $H_{0,2a}$

Table 7 shows the results of the testing hypothesis $H_{0,2a}$ using a *one-tailed t-test for independent samples*. For significance level $\alpha = 0.1$, the score difference between post-test and pre-test for the dependent variables Y.2 and Y.4 are significantly larger in group A as compared to group B, and thus hypothesis $H_{0,2a}$ can be rejected for these variables. It can also be noted that the values of variable Y.3 support the direction of the expected relative learning effect, showing a medium to large effect size. The value for variable Y.1 does not even support the direction of the expected relative learning effect. We achieve significant result for variable Y.4; however, for the KL and the Oulu experiments, the value of Y.4 does not support the direction of the hypotheses.

Table 7. Result for performance improvement

Group A versus B					
Variable		***df***	***t-value***	***Crit t $_{0.90}$***	***p-value***
Y.1	-0.11	9	-0.18	1.38	0.57
Y.2	1.10	9	1.81	1.38	0.05
Y.3	0.64	9	1.06	1.38	0.16
Y.4	0.91	9	1.51	1.38	0.08

3.3 Hypothesis $H_{0,2b}$

Table 8 shows the results of testing hypothesis $H_{0,2b}$ using a *one-tailed t-test for independent samples*. For significance level = 0.1, the post-test scores of variable Y.3 are significantly larger for the experimental group A as compared to the control group B, and thus hypothesis $H_{0,2b}$ can be rejected for this variable. It can also be noted that the values of variables Y.2 and Y.4 support the direction of the expected absolute learning effect, however, only with a small effect size. The values for variable Y.1 does do not even support the direction of the expected absolute learning effect. As regards to Y.3, we achieved a significant result. In both the KL and Oulu experiments, the values of Y.3 even did not support the direction of the hypothesis.

Table 8. Results for Post-test performance

Group A versus B *Variable*		*df*	*t-value*	*Crit. $t_{0.90}$*	*p-value*
Y.1	-0.35	9	-0.57	1.38	0.71
Y.2	0.21	9	1.30	1.38	0.11
Y.3	0.13	9	2.93	1.38	0.01
Y.4	0.28	9	0.73	1.38	0.24

3.4 Qualitative Results

In addition to filling in the pre-test and post-tests and the questionnaires about potential disturbing factors, the participants of the case studies had the chance of making comments or improvement suggestions, and could raise issues or problems that they encountered during the treatments. Comments and statements mainly supported the findings of the quantitative analyses. Positive comments about its usefulness as a whole were made in both groups. Negative comments or problem statements mainly addressed the difficulty of understanding the whole amount of information (both groups) and mainly the structure of the complex system dynamic model in the experimental group. In a lone case in group A, there was a concern with the lack of time for getting acquainted with the tools and for working through the treatments. This was an important objection in the KL experiment, and it was a relatively minor issue with the Oulu experiment.

3.5 Analysis, Summary, and Discussion

Table 9 shows the main results of all the 3 experiments as regards to the testing of null hypotheses $H_{0,2a}$ and $H_{0,2b}$, respectively. Meta-analysis techniques [8] are used for comparing and combining results from different studies. The benefit of meta-analytic procedures is that by combining the results of a number of studies, one can increase the power of the statistical analysis. This enables one to identify effects that could escape the scrutiny in a single study with much lower statistical power. Meta-analytic techniques are based either on p-values or effect sizes. To make a step in this direction and include both, p-values as well as effect sizes, in the discussion, the hypothesis testing results of each study were classified as follows:

- Statistical significance (sta. sig.): null hypothesis could be rejected at significance level $\alpha = 0.1$.
- Practical significance (pract. sig.): null hypothesis could not be rejected but effect size $\gamma \geq 0.5$.
- Positive effect (+): no practical significance could be observed but $\gamma > 0$. The number in parentheses indicates how many subjects would have been needed to achieve statistical significance with the given effect size.
- No effect or negative effect (-): t-value ≤ 0.

Table 9 shows that null hypothesis $H_{0,1}$ could only be rejected in all experiments for variable Y.3 (both for the experimental and the control groups). In addition, for the experimental group, $H_{0,1}$.could be rejected in all cases for Y.2 and in one case for Y.1. For the control group, $H_{0,1}$ could be rejected in two cases for Y.1 too.

Table 9. Summary of individual results of $H_{0,1}$

Variables	*Group A*			*Group B*		
	KL	*Oulu*	*Reading*	*KL*	*Oulu*	*Reading*
Y.1	stat. sig.	+	+	-	stat. sig.	stat. sig.
Y.2	stat. sig.	stat. sig.	stat. sig.	+	–	+.
Y.3	stat. sig.	stat. sig.	stat. sig.	stat. sig.	stat. sig.	stat. sig.
Y.4	+	–	stat. sig.	+	–	+

Table 10 shows that null hypothesis $H_{0,2a}$ could only be rejected in all cases for variables Y.2. A significant result was achieved in one case for variable Y.1. Regarding null hypothesis $H_{0,2b}$ statistical testing yielded statistically and practically significant results for variable Y.2. In the KL and the Oulu experiments, there is no indication that the experimental group performs better than the control group with regard to understanding of simple and complex project dynamics (variables Y.3 and Y.4). However, in the Reading experiment, a better performance for the experimental group has been obtained for these variables. The role-play scenario explicitly states project management principles that were not so clearly specified for the control group. On the other hand, this task imposed additional time pressure on the subjects in the experimental group, which might have resulted in low scores for questions related to dependent variables Y.3 and Y.4 in the KL and the Oulu studies. This was not observed in the Reading experiment.

There is a major difference between Reading experiment and the other two in relation to the experience of the subjects. Reading students were in the middle of a course on Software Engineering, and only a few weeks before they were introduced to principles of project management. We believe that such issues related to project management were fresh in their mind, and that might have been the reason why the results were better as regards to Y.4 and hypotheses $H_{0,1}$ and $H_{0,2a}$.

Table 10. Results of $H_{0,2}$

Variables	*KL Experiment*		*Oulu*		*Reading*	
	$H_{0,2a}$	$H_{0,2b}$	$H_{0,2a}$	$H_{0,2b}$	$H_{0,2a}$	$H_{0,2b}$
Y.1	stat. sig.	+(1000)	–	+(56)	–	+
Y.2	Pract. sig.	stat. sig.	stat. sig.	stat. sig.	stat. sig.	pract. sig.
Y.3	–	–	–	–	pract. sig.	stat. sig.
Y.4	–	–			Stat. sig	+

4 Threats to Validity

Construct Validity. It is the degree to which the variables used in the study accurately measure the concepts they purport to measure. The related issues remain same as the KL experiment [11]. We highlight the points here.

1. The mere application of a SD model might not adequately capture the specific advantages of SD models over conventional planning models
2. Interest in a topic and evaluation of a training session are difficult concepts that have to be captured with subjective measurement instruments.
3. It is difficult to avoid "unfair" comparison between SD models and COCOMO, because SD models offer features that per definition are not available for COCOMO. Since exclusively subjects of the experimental group perform scenario block 2, subjects of the control group might be disadvantaged.

Internal Validity. It is the degree to which conclusions can be drawn about the causal effect of the independent variable on the dependent variables. Potential threats include selection effects, non-random subject loss, instrumentation effect, and maturation effect. These issues also remain same as the original experiment:

1. A selection effect was avoided by random assignment of subjects.
2. Non drop-out of subjects has been avoided by the experimental design.
3. The fact that the treatments of group A and B were different in the number of scenario blocks involved and, as a consequence, in the time available to perform each scenario block, may have induced an instrumentation effect.

External Validity. It is the degree to which the results of the research can be generalised to the population under study and other research settings. The two possible threats are:

1. The subjects participating in the experiment were all students in computer science or related fields at an advanced level. Any generalisation of the results with regard to education of novice students, or even with regard to training of software professionals should be done with caution.
2. Adequate size and complexity of the applied materials might vary depending on previous knowledge about SD modelling and COCOMO.

5 Conclusions and Future Work

The empirical studies presented in this paper investigated the effect of using a System Dynamics (SD) simulation model to assist software project management education of computer science students. The treatment focused on problems of project planning and control. The performance of the students was analysed with regard to four dimensions, i.e., interest in the topic of software project management (Y.1), knowledge of typical project behaviour patterns (Y.2), understanding of simple project dynamics (Y.3), and understanding of complex project dynamics (Y.4). The

findings of the current replicated study corroborates the finding the first two experiments in the sense that using SD models increase the interest of the subject in software project management and also improve a students' knowledge of typical behaviour patterns. Hence, SD models represent a viable path for learning multi-causal thinking in software project management. This was supported by the subjective evaluation of the role-play scenario involving simulation with the SD model, which received very high scores.

Future work will include its replication for two reasons. A further replication should consider the examination of cause/effect relationships. And second, each empirical study exhibits specific threats to validity, which can only be ruled out by replication. We also intend to further analyse other dynamic methods such as Bayesian networks (BN) [7]. An experiment of comparing SD and BNin the context of software project management education is a part of our future work.

Acknowledgements. The research was supported by the Spanish Research Agency (CICYT- TIC1143-C03-01) and The University of Reading.

References

[1] T. K. Abdel-Hamid and S. E. Madnick, *Software Project Dynamics: an Integrated Approach*. Prentice Hall, 1991.
[2] B. W. Boehm, *Software Engineering Economics*. Prentice-Hall, 1981.
[3] B. W. Boehm, "Industrial software metrics top 10 list", in *IEEE Software*, 1987, pp. 84-85.
[4] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences. Second Edition*. Academic Press, 1988.
[5] J. W. Forrester, *Principles of Systems*. Norwalk, CT: Productivity Press, 1961.
[6] A. K. Graham, J. D. W. Morecroft, P. M. Senge and J. D. Sterman, "Model-supported case studies for management education," *European Journal of Operational Research*, vol. 59151-166, 1992.
[7] F. V. Jensen, *An Introduction to Bayesian Networks*. UCL Press, 1996.
[8] J. Miller, "Applying meta-analytical procedures to software engineering experiments," *Journal of Systems and Software*, vol. 54, no. 1, pp. 29-39, 2000.
[9] J. D. W. Morecroft, "System dynamics and microworlds for policymakers," *European Journal of Operational Research*, vol. 35301-320, 1988.
[10] D. Pfahl, M. Klemm and G. Ruhe, "A CBT module with integrated simulation component for software project management education and training," *Journal of Systems and Software*, vol. 59, no. 3, pp. 283-298, 2001.
[11] D. Pfahl, N. Koval and G. Ruhe, "An Experiment for Evaluating the Effectivenesss of Using a system Dynamics Simulation Model in Software Project Management Education," Metrics Symposium, London, UK, 2001, pp. 97-109.
[12] D. Pfahl, O. Laitenberger, J. Dorsch and G. Ruhe, "An Externally Replicated Experiment for Evaluating the Learning Effectiveness of Using Simulations in Software Project Management Education," *Empirical Software Engineering*, vol. 8, no. 4, pp. 367-395, 2003.
[13] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press, 1997.
[14] F. Shull, V. Basili, J. Carver, M. J.C, G. H. Travassos, M. Mendonca and S. Fabbri, "Replicating Software Engineering Experiments: Addressing the Tacit Knowledge Problem," ISESE, Nara, Japan, 2001.