Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

DECISION MAKING SUPPORT IN CMMI PROCESS AREAS USING MULTIPARADIGM SIMULATION MODELING

Daniel Crespo
Mercedes Ruiz

University of Cadiz
11003 - Cadiz, SPAIN

ABSTRACT

Estimates of task duration or the amount of resources needed in software projects are often very inaccurate. To avoid this problem, project management must be effective and dynamic, that is, being proactive rather than reactive. Among the tasks needed in this approach, reassigning resources, hiring new personnel or adapting estimates to new situations can be found. In this paper we propose to apply multiparadigm simulation modeling in the scope of two process areas of one of the most used software process maturity frameworks such as CMMI, with the aim of supporting decision making and determining the optimal values of cost and schedule according to the management needs. The paper describes the model built and a case study with the simulation outputs.

1 INTRODUCTION

Now, more than ever, organizations want to develop products and services in an optimal, fast and inexpensive way. At the same time, in the 21st century high technology sector, almost every organization has found itself developing products and services of increasing complexity. The problems these organizations face nowadays need solutions that require the involvement of the entire company and an integration approach. In essence, these organizations are product and service providers that need to manage their development activities under an integrated approach as part of their way towards business goal achievement.

CMMI (Capability Maturity Model Integration) consists of best practices for the development and maintenance activities of products and services (CMMI Product Team 2006). CMMI can be used as: a) a best practice collection in process improvement activities, b) a framework to prioritize and organize activities, c) a support to coordinate multidisciplinary activities to properly build a product and/or d) a way to align improvement process goals with organization business goals. This paper proposes the application of multiparadigm simulation modeling to the CMMI Project Planning and Project Monitoring and Control process areas. The aim of the model is to support the design and execution of the Specific practices (SP) described in the process areas aforementioned. The model integrates discrete event simulation and agent-based simulation approaches and has been validated using the data from the International Software Benchmarking Standard Group (ISBSG) v. 10 project repository (ISBSG Repositories 2012).

The paper is organized as follows. Section 2 offers a brief description of the CMMI process areas involved and shows the issues associated to put them into practice. Section 3 presents main project management techniques and references some current works studying this problematics. Section 4 explains the main features of the proposed multiparadigm simulation model. Section 5 describes the different scenarios used for simulation and summarizes the obtained results. Finally, Section 6 contains conclusions and future work.

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

2 CONTEXT DESCRIPTION

A process area is a cluster of related best practices in an area, which when implemented collectively, satisfy a set of goals considered important for making significant improvement in that area (CMMI Product Team 2006). Each process area consists of goals and practices. There are two categories of goals and practices: generic and specific. Specific goals and practices are specific to a process area. Generic goals and practices are a part of every process area.

In this section the CMMI process areas used in this study are described and analyzed focusing in the problems associated to their design and implementation.

The purpose of Project Planning (PP) is to establish and maintain plans that define project activities. The main goals belonging to this area follow: a) establish estimates, b) develop a project plan and c) obtain commitment to the plan.

In general, the difficulties inherent to this area include determining estimates of work products and task attributes, defining the life cycle for the project, determining cost and effort estimates, establishing budget and schedule, identifying project risks and establishing a plan for project resources.

On the other hand, the purpose of Project Monitoring and Control (PMC) is to provide an understanding about the progress of the project so that appropriate corrective actions can be taken when the project performance deviates significantly from the plan. This process area aims at two objectives: a) monitor project against plan and b) manage corrective action to closure.

Taking into practice this area presents several problems which can be summarized in two main groups: a) monitoring project planning parameters (cost, schedule, etc.) in order to detect significant deviations from the project plan and b) monitoring the risks that may appear with the purpose of taking corrective actions.

3 RELATED WORK

The following studies can be included among the current work proposing the application of computational techniques in project planning: In Drabble 1995 artificial intelligence has been used to design a new project planning approach that can handle projects conducted by geographically separated organizations. Data mining has been applied in Yousef et al. 2006 and Prasad, Arsiwala, and Singh 2010 in order to analyze success factors in a software project and estimate and improve success probability respectively. A new project planning method is proposed in Wang and Han 1997 using linear programming.

Research on the application of simulation techniques within CMMI framework has been done in order to address both higher levels achievement and process design and optimization. Namely, simulation techniques based on CMMI ontology has been used in Lee et al. 2008 and Lee, Wang, and Chen 2008 to develop an intelligent estimation agent-based on CMMI ontology for PP and PMC process areas. System dynamics simulation has been used in Birkholzer et al. 2005 to design a model capable of reflecting underlying strategies for advancing or maintaining an organization's processes which can be used by all stakeholders involved in software development to better understand the various aspects of software engineering. Chen, Zhou, and Luo 2010 presents a process optimization method that can be applied to CMMI Level 4 and Level 5. Li et al. 2011 proposes a software process model with risk management and cost control modules to help improve software process risk management. In Raffo and Wakeland 2008 simulation techniques for process design and documentation have been introduced in order to improve their comprehension. Moreover, it shows how to perform bottom-up cost estimations in a project using process simulation. Finally, Miller, Pulgar-Vidal, and Ferrin 2002 suggests potential utilities of simulation to achieve higher levels in CMMI.

Apart from CMMI supporting, simulation techniques have been used also in Joslin and Poole 2005 with the aim of adapting an agent-based simulation model for project planning used to plan experimental activities of NASA's Mars Rover. Madachy Madachy 1994 came up with a dynamic simulation model for software projects risks evaluation. In Host, Regnell, and Tingstrom 2008 a framework for simulation of requirements engineering processes is presented. It is also possible to combine different simulation approaches, which is usually called hybrid or multiparadigm simulation. Hybrid models based on discrete events and system dynamic models have been used in Wakeland, Martin, and Raffo 2004 to evaluate changes to a software development process.

Regarding resource allocation studies in a software project, Antoniol, Di Penta, and Harman 2004 address this problem from the point of view of massive maintenance projects, in Shan, Jiang, and Huang 2010 a genetic algorithm that carries out this task is proposed and in Sebt et al. 2009 linear programming techniques have been used to develop a resources allocation algorithm. Tsai, Moskowitz, and Lee 2003 deal with resource selection problem for choosing the best project team by using Taguchi's parameter design.

The main contribution of our study consists in using a multiparadigm simulation model integrating discrete event and agent-based simulation approaches to model concrete parts of the development process in software projects. The project coding process is simulated by an agent-based model, thus obtaining a more realistic vision and a much more accurate estimation when modeling the project team behavior from the point of view of the people involved. This model supports the specific practices of the PP and PMC CMMI process areas emphasizing schedule fixing, effort estimation, adequate life cycle election, project total cost estimation and resources number needed to develop the project.

4 MODEL DESCRIPTION

In this section the problem considered for the case study and the multiparadigm simulation model proposed are described.

4.1 Problem Description

For the purpose of our study, we have considered a Software Factory certified in CMMI Level 2 that develops custom software for different customers. Aside from Management and other departments that have no direct influence on the considered process, the organizational structure of the company includes the following roles:

- Manager : Responsible for supervising the work that is being done, controlling the production and acting as a link between the company and the customers.
- Project manager: Responsible for planning the work assigned to their team and leading the analysts.
- Analyst: Responsible for software analysis and design. Each analyst is in charge of one or more programmers.
- Programmer: Their goal consists of developing the software designed by the analysts.
- Tester: The tester performs software tests.

The proposed model aims at solving the problems faced by the organization when it comes to designing and implementing the CMMI Level 2 process areas related to project management that can be benefited from simulation techniques (Raffo and Wakeland 2008), namely PP and PMC. To achieve this goal, the amount of resources needed is studied in order to complete the project attending to cost and schedule constraints and maximizing the profit margin obtained by the organization at the same time. On the other hand, the resource allocation process is closely analyzed by implementing different strategies for resource allocation and simulating different corrective actions in case of detecting significant deviations. From the point of view of the project life cycle, code and design errors generation is studied and corrective actions procedures are simulated.

4.2 Simulation Model Development

Following Kellner's proposal for describing simulation models (Kellner, Madachy, and Raffo 1999) and Martinez and Richardson's methodology (Martinez and Richardson 2001), the simulation model developed is described below. The model implementation and the simulation runs have been performed using

Anylogic™simulation software with the Enterprise Library. The model logic is written in Java and the databases have been designed and implemented using MySQL.

4.2.1 Purpose and Scope of the Model

The proposed model aims to support the design and construction of the CMMI PP process area providing tools that help to carry out the specific practices proposed in the area. The scope of the model covers the whole execution of a particular project in the organization modeled implementing one or several CMMI process areas and being PP one of them.

4.2.2 Output Variables

The main variables providing information about model purpose are the following:

- *End date*: Date for project completion.
- {*Coordinators, Analysts, Programmers, Testers*} *Effort*: Number of hours each role has worked in the project.
- *Cost*: Cost of the project.
- *Profit margin*: Ratio of profitability obtained in the project. Profitability is the result of subtracting direct costs of the project to the economic benefits obtained.
- *Defects*: Number of defects found by the users of the software product during its first month of usage.

4.2.3 Input Parameters

Input parameters allow to set up different scenarios for simulation. In order to be as meaningful as possible, the current version of the model has 45 input parameters. However, for the sake of clarity in this paper we only describe the input parameters that play a relevant role in this case study. These parameters have been grouped into the following five categories:

- Project parameters:
  - *Budget*: Approved budget for the project.
  - *Size*: Software functional size measured in adjusted function points.
  - *Schedule*: Estimated time for the complete execution of the project.
  - *Phases number*: Number of phases in the project when using a incremental life cycle.
  - *Software and hardware investment*: Budget approved for hardware and software investments.
  - *Programming language*: Type of language used.
- Task parameters:
  - *Complexity*: Estimated task complexity (low, medium or high).
  - *Type*: Type of task (coding, design, etc.).
  - *Priority*: Task can have a low, medium or high priority. This parameter determines the start order of the tasks in compromised situations.
  - *End date*: Task deadline.
  - *Phase*: Phase to which a task belongs in the project using an incremental life cycle.
- Programmer parameters:
  - *Skills*: Technical competences of the programmer, such as particular technologies or programming language knowledge.
  - *Teamwork influence*: Encouragement felt by a programmer when working in a team.
  - *Adaptation rate*: Ease or difficulty in adapting to new tasks.
  - *Learning rate*: Learning capacity of a programmer when he is being taught about new concepts or techniques.

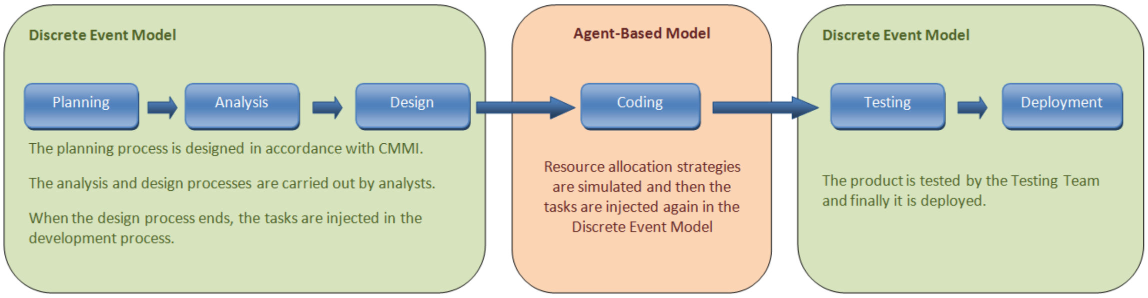

Figure 1: Model architecture.

  - *Commitment*: Degree of commitment of the programmer with the organization.
  - *Proactivity*: Proactivity of the programmer.
  - *Programming language skills*: Skill degree of the programmer when writing code.
  - *Development methodology skills*: Skill degree and knowledge of the development methodology used by the organization.
- Inspection Process parameters
  - *InspectionEffort*: Effort needed for work reviewing.
  - *InspectionEfficiency*: Ratio of efficiency of the inspectors.
  - *ReworkEffort*: Rate associated to solve errors, measured in hours.
  - *AmplificationImpact*: Rate referring to the impact of the overlooked errors on the following phases of the inspection.
- Control parameters
  - *Strategy*: Sets the resource allocation strategy that will be used.
  - *Life Cycle*: Shows the life cycle that will be used in the project.
  - *Inspection*: Enables/Disables the Inspection process.
  - *Generate Errors*: Simulates code and design errors during project execution.

The values used for the programmer parameters are obtained after a conversion of qualitative data stored in the organization. Specifically, project managers perform regular evaluations of the team members according to their performance and acquired knowledge and skills. Table 1 shows the relationship between the grades considered in the evaluations and the quantitative values used in the proposed model. Thus, the values given are modeled as percentages representing the grade acquired by a programmer in a particular competence or skill.

Table 1: Equivalence between qualitative and quantitative values of programmer parameters.

Grade	A	B	C	D	E	F
Quantitative grade	1.0	0.8	0.6	0.4	0.2	0.0
Equivalent percentage	100	80	60	40	20	0

4.2.4 Process Abstraction

Figure 1 shows the architecture of the model. The model consists of two main submodels: a) A discrete event simulation model, to simulate the PP process area together with the processes of the project life cycle that results from the plan already developed, and b) an agent-based simulation model, to in-depth simulate the coding process of the project life cycle aforementioned.

A description of each of these submodules follows:

Discrete event model

First, this model simulates the set of specific practices of the PP process area as described in Section II. As a result, a project plan is obtained. To test the suitability and viability of this plan, the model simulates the main processes of the execution of the software project deriving from that plan. The processes simulated of the project life cycle are the following:

1. *Analysis and Design process*: During the analysis phase, analysts are responsible for performing the analysis of the system to be developed. The model determines the time needed to develop this phase, which eventually will depend on the number of tasks, their complexity and the resources available. In the design phase, the design of each task of the project is made. Once again, the time needed to carry out this phase will depend, mainly, on the resources available and the number and complexity of tasks involved.
2. *Coding process*: As stated before, this process has been modeled following the agent-based simulation approach. Under this approach, programmers are modeled as agents whose behavior is determined by a state diagram. During this phase, each programmer will develop the tasks that have been assigned to him. This model is further explained in the following sub-section.
3. *Unit Testing process*: Whenever a phase of the project ends, the work products developed are tested. In this phase, testers need to design and execute the tests. The time needed to develop this phase also depends on the resources, the number and the complexity of the tasks involved.
4. *Integration Testing process*: Once the project phases are over, it is time to test the integration of all the components of the software product. The model takes into account that the time needed to develop this phase and the results obtained depend also on if and how many inspection activities have taken place during the project execution. When the simulation of this process ends, the model simulates the rework activities performed by the programmers and the testers performing unit and integration tests.
5. *Deployment process*: Once the product has been approved for release, it is deployed in an environment similar to the one of the customer.

Agent-based model

The agent-based model is used to simulate the project coding process. The agents are the programmers of the organization and each one of them is described according to the parameters mentioned in Section 4.2.3. During a real project development, the estimated coding time for a task rarely coincides with the time employed. Actually each person produces at a different level depending on certain skills and characteristics inherent to them. This paper proposes a more realistic estimation of a software product development by means of calculating the expected productivity of each resource individually. Productivity is a factor that may vary over time. When a task is getting started, it is common that the productivity is lower due to the familiarization process with new commitments. Also new personnel incorporation into a project that already has been started has a negative effect over productivity since part of the time is being invested in helping newcomers.

For each work hour, actual production time comes from equation 1.

$$Productivity = GCW * C.G. + TCW * C.T. \tag{1}$$

Where *GCW* and *TCW* are the weights associated to Generic and Technical competences respectively, and *G.C.* and *T.C.* comes from equations 2 and 3 described below:

$$C.G. = TWIW * TWI + AdW * Ad + ProW * Pro + CommW * Comm \tag{2}$$

$$C.T. = PLW * PL + MetW * Met \tag{3}$$

Where *TWIW*, *AdW*, *ProW*, *CommW*, *PLW* and *MetW* are the weights associated to Teamwork Influence (*TWI*), Adaptation (*Ad*), Proactivity (*Pro*), Commitment (*Comm*), Programming Language Proficiency (*PL*) and Methodology Proficiency (*Met*) respectively. When using the model within an organization, it is necessary to calibrate the values of these parameters using historical data available in the organization.

The model also allows to study the effects of different types of resource allocation strategies within the coding process. In this study, two strategies are analyzed:

- Strategy 1: While there are idle resources, they are assigned to new tasks of the pending tasks set.
- Strategy 2: While there are idle resources, determine if a task needs an extra support in order to finish on time. If so, assign the resource to the running task. If not necessary, the resource is assigned to a new task of the pending tasks set.

4.3 Model Verification and Validation

The equations used in the model have been validated by means of unit and integration tests. Regarding the validation of the model we have followed the recommended procedure described in Sargent 2011 using several validation techniques including Operational Graphics and Animation, Event Validity, Extreme Condition Tests, Face Validity, Traces and Sensitivity Analysis. The model has been calibrated using the data provided by ISBSG ver. 10 project repository, with data that can be licensed and used to estimate, benchmark and improve the planning and management of projects. The repository has close to 6,000 projects from 21 countries, across 15 major industry types.

5 CASE STUDY

Once the model has been successfully verified and validated it can be used to simulate different scenarios. In this study, we present the results of three different experiments. The first one allows us to analyze the effect of different resource allocation strategies in the coding phase of the project. The second experiment is designed to determine the optimal resource number required to accomplish the project. The third experiment helps to analyze the pros and cons of executing an inspection process within the project. For each experiment, a description of the parameters involved and their values, and the simulation output are presented and discussed.

5.1 EXPERIMENT 1. Analyzing the Effect of Resource Allocation Strategy

The model has been simulated assuming that the analysis and design processes have been developed up to a point in which it is possible to begin the coding process. Following the different groups of input parameters used in this experiment are shown together with the initial values of the parameters in each group.

Project Parameters

Taking into consideration projects and teams from medium-sized software factories, the following values have been used for the parameters: *FunctionalSize*: 500. *Language Type*: 3GL. *Programmers Number*: 7. *Coordinators Number*: 2. *Analysts Number*: 2. *Testers Number*: 5. *Schedule*: 50 days. *Budget*: 90,000€. *Phases number*: 3.

Tasks Parameters

The parameter values relating to the coding tasks are stored in a database containing estimation data. This database contains information based in historical data from the organization under case study.

Programmer Parameters

Since the project coding process has been defined by means of an agent-based model, the parameters defining the programmers of the organization (agents of the model) must be provided. These are collected in a database for this purpose. For the sake of clarity, only an example of the values set for a single programmer is given in the following: *Skills*: B. *Teamwork influence*: B. *Adaptation*: C. *Learning*: C. *Proactivity*: A. *Commitment*: A. *Programming language skills*: B. *Development methodology skills*: B.

Decision Parameters

The default parameter values belonging to this group follows: *Strategy*: 1. *Life Cycle*: Incremental.

Before beginning the resource allocation activity, the tasks are arranged according to the sequence established in the project plan. The resource allocation strategy (RAS) number 1 looks for the most adequate resource to carry out a particular task. In order to determine the suitability of an assignment, it looks for an idle resource that has the skills needed for the task. If no resource is found with the specific skills desired, the task will be assigned to any idle resource.

RAS number 2 differs from the previous one in that it is possible to allocate additional personnel to a single task as long as the model predicts that the task will not be accomplished on time. In order to make this prediction, the model estimates the probability of finishing each task on time periodically. To estimate this probability the model uses the current value of the resources performance for each task and assumes that this level will remain uniform all over the remaining time to develop each task. If as a result of such estimation it is deduced that the task cannot be completed within the estimated time, a new resource will be assigned to the task as long as such allocation do not harm in excess the overall performance and quality of the project. Additionally a task will never have more than four resources assigned and adding a new resource into a task in progress will increase the probability of getting software defects. Such probability increment is parameterized and it has been fixed in 5% for this particular experiment.

In this experiment the RAS described above have been simulated using the parameter configuration aforementioned. The results are shown in Table 2.

Table 2: Results obtained after simulation.

	Total Cost (€)	**Duration (Days)**	**Profit Margin**	**Defects**
RAS 1	56,320	64	33,680	1
RAS 2	54,400	61	35,600	4

According to the results obtained, RAS number 1 seems better than RAS number 2 in terms of cost and duration. Nevertheless, the defects number during the first month of use increases slightly. It can be concluded that, as long as a control exists over the allocated resources number to a single task, is preferable using this strategy unless the customer is willing to extend the project duration and to assume an appreciably higher cost in benefit of final quality of the project.

This experiment shows how the proposed model helps in the context of the CMMI PMC process area. Namely, it is possible to predict the effect of the implementation of corrective actions as well as monitoring project performance parameters. In this particular case, resources allocation is an important part of decision making since the results obtained after choosing one or another strategy may be different.

5.2 EXPERIMENT 2. Determination of the Optimal Number of Resources

The goal of this experiment is to determine the optimal number of resources of each role to develop the project. To that end, the deadline of the project must be considered and the obtained profit margin will be maximized. The parameters used for this experiment and their values follow:

- Number of simulations: 500.
- Input parameters: Input parameters described for Experiment 1.
- Control parameters for the optimization experiment: *numProgrammers* (varies from 1 to 7), *numTesters* (varies from 1 to 5), *numCoordinators* (varies from 1 to 2) and *numAnalysts* (varies from 1 to 2).

The optimization starts from a set of control parameters suggested by the project manager that could come from historical data of the organization or other kind of estimations. Table 3 shows the difference between the initial values of the parameters and the ones obtained by the experiment. Figure 2 illustrates

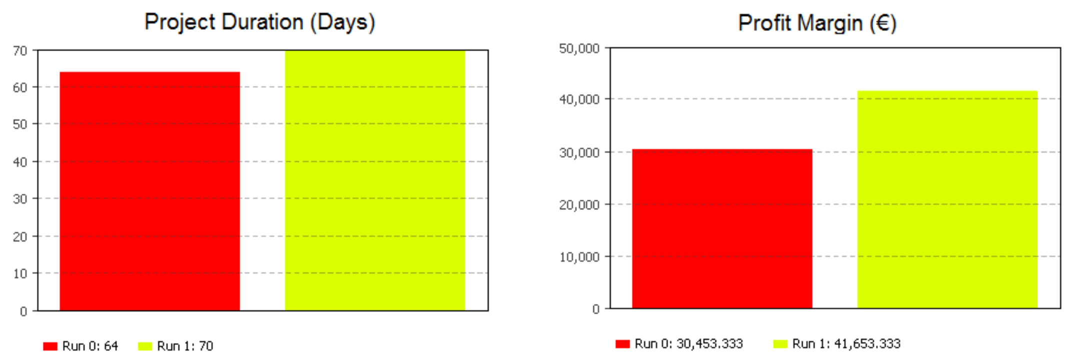

Figure 2: Comparison between execution with a priori estimated parameters (Run 0) and execution with optimal parameters (Run 1).

the schedule and profit difference using the initial values for the parameters and the values calculated as a result of the experiment.

In order to carry out this experiment, the OptQuest™optimization engine has been used. OptQuest™Engine uses metaheuristic, mathematical optimization, and neural network components to guide the search for best solutions to decision and planning problems of all types. The model tells OptQuest™Engine the quality of each solution generated during the search, by calling an evaluator that can take any form.

Table 3: Comparison between initial and calculated values for parameters.

	Initial values	Optimal values
numCoordinators	2	1
numAnalysts	2	2
numProgrammers	7	5
numTesters	4	3
Team Size	**15**	**11**

In this experiment it is shown that the determination of the optimal number of resources for a particular project is crucial. In fact, small deviations in the determination of this number represent significant increases in the project cost. The proposed model is useful to support in an optimal way some of the specific practices covered by CMMI PP process area: establish the budget and schedule, define a plan for project resources, obtain commitment to the plan, reconcile work and resource levels and contribute to the elaboration of the project plan.

5.3 EXPERIMENT 3. Runs Comparison Experiment using the Inspection Process

The aim of this experiment consists in the determination of pros and cons when using the Inspection Process (IP) during project execution. To that end, project execution has been simulated two times enabling and disabling IP respectively, and enabling the error generation simulation in design and code in both scenarios. The obtained results are shown in Table 4. Figure 3 a) depicts the effort distribution among the different project roles modeled during execution disabling the IP as well as the number of hours spent in rework. Figure 3 b) shows the same type of information for the project execution enabling the IP.

Table 4: Results after enabling/disabling IP.

	Without IP	With IP
Duration (days)	99	70
Cost (€)	87.680	77.093

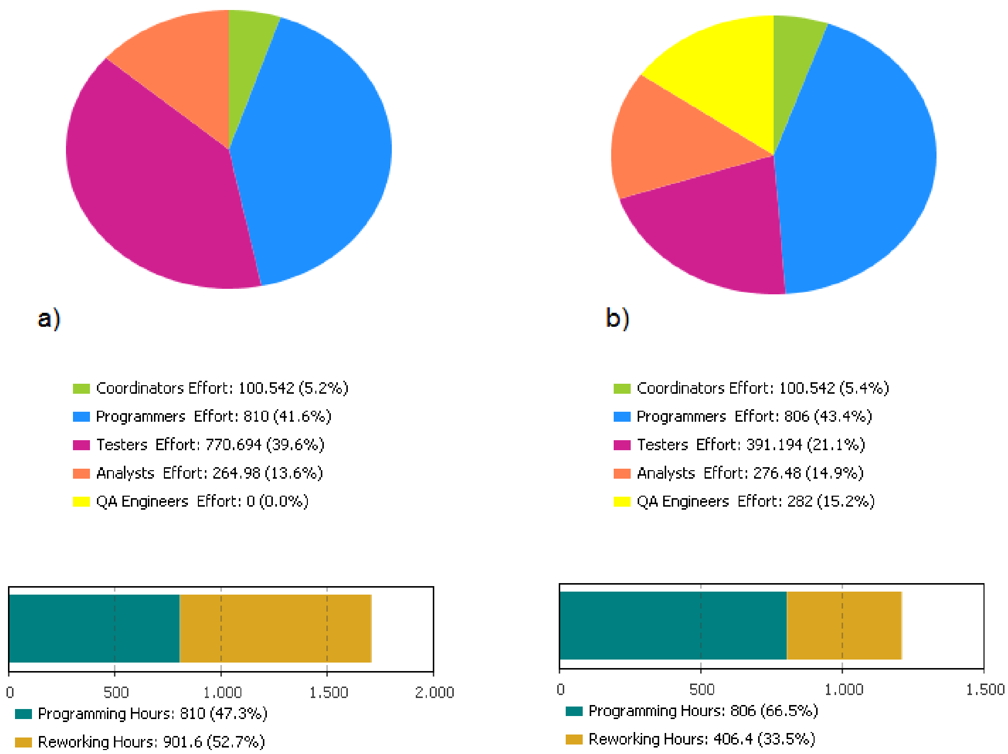

Figure 3: Effort distribution disabling IP (a) and enabling IP (b).

According to the results obtained, it follows that using the IP is very advantageous. By reviewing the product while it is being developed, an important percentage of errors is detected. If such errors go undetected until the Integration Testing process, important delays may be caused due to the difficulty of detecting the error source and correcting it without producing collateral effects. The correction of most of these errors during project execution produces important benefits in terms of project cost and schedule.

This experiment shows how the proposed model supports some specific practices of both CMMI PP and CMMI PMC process areas, specifically: determine estimates of effort and cost, identify project risks, monitor project risks, analyze issues and take/manage corrective actions.

6 CONCLUSIONS AND FUTURE WORK

In this study, simulation techniques have been applied to project management within a process framework in order to achieve process improvement. Specifically, a multiparadigm simulation model in the realm of CMMI PP and PMC process areas has been developed in order to support the specific practices of the areas emphasizing schedule fixing, effort estimation, adequate life cycle election, project total cost estimation and resources number needed to develop the project. This has been done by simulating the development process of a project from planning to deployment starting from project information known by the organization which has been modeled.

Due to the large number of possibilities that simulation techniques may offer, there may be numerous improvements and contributions to the proposed work. The purpose of our future work follows:

- Keep on calibrating and validating the model.
- Simulate different alternatives in particular process area construction in order to determine which one fits better to the organization, studying the process behavior over time.

- Simulation-support for CMMI Level 4 process areas building and quantitative project management simulation.
- Simulate the organizational structure.
- In-depth simulation of the testing team.
- Enhance the model to simulate other CMMI process areas, such as Integrated Project Management, Organizational Process Performance and Quantitative Project Management.
- Help as a training tool for software engineering students and practitioners.

ACKNOWLEDGMENTS

This research is partially funded by the Ministry of Education and Science of Spain and European FEDER funds under project TIN2010-20057-C03-03.

REFERENCES

Antoniol, G., M. Di Penta, and M. Harman. 2004. “Search-based techniques for optimizing software project resource allocation”. In *Proceedings of Genetic and Evolutionary Computation–GECCO 2004*, edited by K. Deb, R. Poli, W. Banzhaf, H. Beyer, and E.Burke, 1–11.

Birkholzer, T., C. Dickmann, J. Vaupel, and L. Dantas. 2005. “An interactive software management simulator based on the CMMI framework”. *Software Process: Improvement and Practice* 10 (3): 327–340.

Chen, T., B. Zhou, and W. Luo. 2010. “A Process Optimization Method for High Maturity Process Improvements”. In *Proceedings of Management and Service Science (MASS), 2010 International Conference on*, 1–4. IEEE.

CMMI Product Team 2006, August. “CMMI for Development, Version 1.2”. Technical Report CMU/SEI-2006-TR-008, Software Engineering Institute.

Drabble, B. 1995, October. “Artificial intelligence for project planning”. In *Proceedings of the IEEE Colloquium on Future Developments in Projects Management Systems*, 1–5.

Host, M., B. Regnell, and C. Tingstrom. 2008. “A Framework for Simulation of Requirements Engineering Processes”. In *Proceedings of Software Engineering and Advanced Applications, 2008. SEAA’08. 34th Euromicro Conference*, 183–190. IEEE.

ISBSG Repositories 2012. “ISBSG Repositories”. Accessed Sep. 11, 2012. http://www.isbsg.org/.

Joslin, D., and W. Poole. 2005, December. “Agent-based simulation for software project planning”. In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 1059–1065. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kellner, M. I., R. J. Madachy, and D. M. Raffo. 1999, April. “Software Process Simulation Modeling : Why? What? How?”. *Journal of Systems and Software* 46 (2/3): 91–105.

Lee, C.-s., M.-h. Wang, and J.-j. Chen. 2008. “Ontology-based Intelligent Decision Support Agent for CMMI Project Monitoring and Control”. *International Journal of Approximate Reasoning* 48 (1): 62–76.

Lee, C.-S., M.-H. Wang, Z.-R. Yan, C.-F. Lo, H.-H. Chuang, and Y.-C. Lin. 2008. “Intelligent Estimation Agent Based on CMMI Ontology for Project Planning”. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 228–233.

Li, J., M. Li, D. Wu, and H. Song. 2011. “An Integrated Risk Measurement and Optimization Model for Trustworthy Software Process Management”. *Information Sciences*.

Madachy, R. J. 1994. *A software project dynamics model for process cost, schedule and risk assessment*. Ph. D. thesis, University of Southern California.

Martinez, I. J., and G. P. Richardson. 2001. “Best practices in system dynamics modeling”. In *Proceedings of the 19th International Conference of the System Dynamics Society Atlanta, Georgia*, 1–23.

Miller, M., F. Pulgar-Vidal, and D. Ferrin. 2002, December. "Achieving higher levels of CMMI maturity using simulation". In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, Volume 2, 1473–1478. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Prasad, A., J. Arsiwala, and P. Singh. 2010, oct.. "Estimating and improving the probability of success of a software project by analysing the factors involved using data mining". In *Proceedings of the 2010 International Conference on Artificial Intelligence and Education (ICAIE)*, 391–394.

Raffo, D. M., and W. Wakeland. 2008, January. "Moving Up the CMMI Capability and Maturity Levels Using Simulation". Technical Report CMU/SEI-2008-TR-002, Software Engineering Institute.

Sargent, R. 2011, December. "Verification and validation of simulation models". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 183–198. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sebt, M. H., A. Akrami, M. T. Banki, and V. Shahhosseini. 2009. "Optimized Allocation of Expert Human Resources to Project". In *Proceedings of 2009 Third Asia International Conference on Modelling & Simulation*, edited by D. Al-Dabass, R. Triweko, S. Susanto, and A. Abraham, 386–391.

Shan, X., G. Jiang, and T. Huang. 2010, August. "The Optimization Research on the Human Resource Allocation Planning in Software Projects". In *Proceedings of the 2010 International Conference on Management and Service Science*, 1–4.

Tsai, H., H. Moskowitz, and L. Lee. 2003. "Human resource selection for software development projects using Taguchi's parameter design". *European Journal of Operational Research* 151 (1): 167–180.

Wakeland, W., R. Martin, and D. Raffo. 2004. "Using design of experiments, sensitivity analysis, and hybrid simulation to evaluate changes to a software development process: a case study". *Software Process: Improvement and Practice* 9 (2): 107–119.

Wang, Y., and S. Han. 1997, July. "A multiple criteria 0-1 linear programming method of project planning and investment decision making". In *Proceedings of Innovation in Technology Management - The Key to Global Leadership. PICMET '97: Portland International Conference on Management and Technology*, 394.

Yousef, A. H., A. Gamal, A. Warda, and M. Mahmoud. 2006. "Projects Success Factors using Data Mining Identification". In *Proceedings of The 2006 International Conference on Computer Engineering and Systems*, edited by A. Salem, M. El-Kharashi, and A. El-Din, 447–453.

AUTHOR BIOGRAPHIES

DANIEL CRESPO is a Ph.D. student in the Department of Computer Languages and Systems at the University of Cadiz, Spain. He received a Master's Degree in Computer Science at the University of Cadiz. In 2010 he obtained a research grant from the Ministry of Education of Spain with the aim of studying the application of simulation in software engineering management. Since 2011 he is a member of the Software Process Improvement & Formal Methods Research Group of the University of Cadiz. He is mainly interested in simulation-based decision support applied in Software Engineering. His email address is dani.crespobernal@alum.uca.es.

MERCEDES RUIZ holds a PhD in Software Engineering from the University of Seville, Spain. She is currently an Associate Professor at the University of Cadiz, Spain, where she leads the Software Process Improvement and Formal Methods Research Group. Her research interests are: Simulation-based decision support applied in Software and Services Engineering, as well as simulation-based learning and assessment. She is the author or co-author of numerous papers in international conferences and journals. She is member of ACM and of the Spanish Chapter of the ACM-SIGCSE. Her email address is mercedes.ruiz@uca.es.