
Performance Analysis of Software Processes Supported by Simulation: a Resolution Problem Process Case Study

Dawilmar Guimarães Araújo^{a1} - Nilson Sant'Anna^b - Germano Souza Kienbaum^b

^aPosgraduate student (LAC-INPE), Lecture (UNITAU)

^bLecture (LAC-INPE)

Abstract: Results expected by organizations do not match the efforts spent to obtain processes defined in environments of software development. In the real world, one justifies this fact by considering the lack of adequate instruments and efficient programs of implementation and follow-up of the performance of these processes, together with the fact that the task is far from being trivial. The search for the achievements of these processes, the measurement and analysis of performance are referred to as a practical-key for the maturity and quality of the processes according to major market frameworks. In this scenery of problems and challenges, this work presents a proposal that supports the definition, measurement and analysis of performance of software processes, and the improvement of these. The proposal is characterized by the agreed use of software modeling and simulation of processes with the performance analysis, the latter subsidizing the former. This work contains two well distinguished parts: a bibliographical review concerning key areas related to software processes, and a case study of one of the processes of framework SPICE. Viability of the proposal, a study on appropriate indicators of performance, a theoretic-scientific approach as contribution to the referred areas and an instrument that supports the definition and management of software processes are some of the results of this case-study. One concludes that a capable software process is a software satisfying the needs of the customer, and that it must be duly adjusted to the reality of the customer organization.

Keywords: Software Engineering, Modeling and simulation of process software, Process software technology, Process management, Analysis of processes performance

1. Introduction

Results expected by organizations do not match the efforts spent to obtain processes defined in environments of software development. One of the points of

¹ Postgraduate student. Laboratory for Computing and Applied Mathematics, Brazilian National Institute for Space Research, São José dos Campos, Brazil
C. Postal 515 – 12245-970 – São José dos Campos - SP – Brazil
Lecture – University of Taubaté (UNITAU). Taubaté-SP-Brazil.
email:dawilmar@lac.inpe.br, dawilmar@unitau.br

this difference is mainly in the characteristics of complexity and use of permanent human resources in EDS-Environment of Development of Software.

On the other hand, it is worth noting the increased dependence of software in all the business-oriented types, as well the growing demand for more complex systems. The companies are dealing with problems with strong emphasis in the management of processes, in the improvement, and the maturity process of software [06].

The results of the efforts found in literature to supply this scenario, maturity models and improvement of processes carried through for the organizations can be seen. As an international example the ISO/IEC-15504 can be mentioned as a result of the Project SPICE (Software Process Improvement Capability dEtermination), framework CMMI (Capability Maturity Model Integration) of SEI (Software Engineer Institute), and particularly to the Brazilian national environment, the MPS.Br model (Improvement of Process of Brazilian Software), that is a model of improvement and evaluation of process of recognized software that endeavors to take care of its necessities and to be nationally and internationally used as a model of the software organizations [07].

The adoption of models and norms as guides are sufficiently promising, however they say what to do and not how to do it. In order to justify and the support the management for its action, proposals for the use of tools of management of processes and analysis that support quantitative methods are highly important.

Therefore, this work presents some results of experiments of a proposal that supports the definition, analysis of performance and improvement software processes. Such proposal is characterized by the combination of the modeling and simulation of processes of software with the performance analysis, the one being subsidized by the other.

This work reflects part of the research carried out by the Software Engineering Group of Software of the LAC/INPE (Laboratory for Computing and Applied Mathematics/Brazilian National Institute for Space Research), in an environment of processes to support the activities of management of the Software Engineering (modeling, execution, simulation and accompaniment) in an integrated way.

2. On the software process

This section is introduced presenting two definitions for software process that evidence elements for the interest of this work.

"The process or a set of processes, used for an organization or project to plan, to manage, to execute, to monitor, to control and to improve the activities related to the construction of software" [08].

"A process of composed software consisting of a set of processes capable of leading the organization involved with the construction of products of software with quality and predictive costs, efficiently and managed with the possibility of constant improvement" [10].

A software process is understood or interpreted by a set of elements, from its definitions and from its relationship namely: activity, task, roles, actors, products and resources. A software process typically is identified by a collection of

activities. An activity is identified by its position in the context of the process and by its relationship with other activities. Which are usually described by rules of the organization and of the process itself. An activity is part of the process that is defined according to its tasks. Abilities or roles are written and defined for the accomplishment of the tasks through its actors. Actors produce products (objective) of the process when playing their roles, their by consuming resources.

The quality of a software product is determined by the quality of the process used for its development and maintenance [05].

The citations presented stress the importance to legalize and to improve the organization processes of software, so as to gain quality and productivity. A process is the main element of an environment of software development since the quality of a software product is directly related to the quality of the process used for its development.

3. Software process simulation

The Modeling and quantitative simulation as tools for the improvement process, had been initially used in the area of conventional manufacture. However, these tools had been applied to the processes of development of software only recently [04].

Simulations of software processes are found of general form in three classic models: event-discrete, continuous and hybrid. For the first model, event-discrete simulation, it is understood that the interest events happen promptly in time and it does not matter what happens between them. Thus, for example, to initiate a codification, to finish the code, to initiate an interpretation and many others can easily be characterized in time. How the second model, of continuous simulation, aims at capturing the dynamic aspects that the previous model does not do. Models of continuous simulation aim at the capture and agreement of phenomena as for example, the activities of the software processes are developed by human agents and consume a certain time that incidentally causes modification in their answers as to productivity, product quality that it generates etc. Factors such as stress or motivation can diminish or increase these answers. How the third model, called hybrid model that appears when there are manipulation of the variables (continuous and discrete) of the two types of models involved in the study and that are considered jointly. Also, when from any modification that it makes in the one or the other model and that they modify the conceptual structure of the original model [04].

4. Performance and capability of the software process

The capability of the software process describes the gamma of results expected by the organizations, that can be reached with the application of the software process.

The performance of the software process represents the reached real results in its execution. Thus, the performance of the software process is focussed in the results, obtained while the capability of the process focussed on results expected [03].

To obtain the capability of a process is somewhat critical. On the basis of the attributes of a specific project, in the environment characteristics and the context in which the process is led, its real performance may not reflect the capability of the total process of the organization. For example, radical alterations in the application

or in the technology employed can place the of the project staff in the situation of apprentices, which it makes capability and the performance of its project diminish in relation to the capability of total process of the organization [03].

The capability agreement must follow that of the performance. In other words, the way to the capability is in first place to obtain the capacity of the process by means of the agreement of the performance of the process in controlable and predictive conditions.

5. Establishing the quantitative indicators of performance

The measure represents the result, obtained with the use of the metric. Metric is the translation in a model of aspects of interest. Indicator the measurable reference of the metric. Thus, for example the indicator of error tax is interpreted by the metric: number of errors is counted in modules divided by the numbers of lines of the module. Metric is a translation or interpretation of the aspects involved in obtaining the indicator. By using the indicator in a real case, taking effective steps, one will obtain a numerical value that represents the aspect translated in the indicator.

The suggestion of indicators of this work was carried out from discussions and analyses of the Software Engineering Group of Software of the LAC-INPE, and it is not intention to make sufficient here, to only show how the group work of the is being conducted as well as the first experiment of this proposal.

Three aspects had been contemplated for the process used in this case study, as shown in Table 1. By means of the indicators an index of planned productivity can be obtained.

Table 1. Indicators of performance, metric and objective global for evaluation of processes.

Indicators	Interpretation (metric)	Objective of the organization
Productive Capacity of the process or total resource in hours (PC)	Nhd [Number of hours and developers involved *numbers of days planned for the project]	it translates the availability of hours destined to the project without considering the occurred stoppages
Planned Productivity (P_p)	Utilization * Efficiency ($\xi\mu$)	It reflects the relation of effective productivity for all planned capacity of the project
Utilization (μ)	Time effectively spend/number total of available hours	Utilization of an operation, translates the percentile relationship between the worked hours and the hours effectively destined in the period of the project
Efficiency (ξ)	Estimated time/effective time spent in the work.	related to the resources foreseen and accomplishit for the execution of an amount of production in a period of the project

It can be said that it is generally accepted that, the future of the metric of software is in the use of relatively simple metric combining different aspects that may permit some types of estimates and evaluations.

6. The problems resolution process

During the execution of a process of software construction, many facts harmful to the project can occur, either foreseen by the management or not. Any way actions must be taken so that these facts do not hamper the normal course of the project, mainly those that may cause compromise of quality, the cost and schedule.

Occurrence is defined as any event or fact that must be considered as a potential element of risk for the project and that needs evaluation for the group responsible for such task.

The process of solution of problems must be initiated from the moment where any person involved in the project finds a problem. For this purpose some procedures they must be followed [02]:

- To open Occurrence - the "reporter", any person involved in the project and that finds a problem, must fill an occurrence report, supplying information on the problem;
- To analyze Occurrence - the analyst, after being notified of the occurrence must verify the correct fulfilling of the information supplied by the reporter and thus analyze the occurrence. Once analyzed, the analyst must to confirm information on the problem to verify the possible impact, the product to be corrected and if the occurrence is justified;
- To solve the Occurrence - the "developer" or solution provider (person who solves the of occurrence), after being notified must solve the described problem in the occurrence report and thus supply the information of the solution. The solution must be directed for revision and validation;
- To review the Occurrence - the "reviewer" must look forward the information supplied and compare them with the criterion of closing, which will be closed if the criteria is met, otherwise, the occurrence must remain open indicating re-working necessity. The "developer", person who will solve the must then be notified to do the re-work.

7. Executing the model of process of resolution of problems

The initial model for experimentation was constructed in the event-discrete environment of simulation. This model is shown in Fig 1.

This section illustrates a hypothetical analysis that was conducted. The presented analysis rapidly shows the evaluation that could be made as an initial revision of a favorable and feasible idea of change of process.

The used data are hypothetical and aim at representing the information from real estimates of the management based on actual data. The shaped process typically reproduces the routine of a company used for the study. The company produces, throughout three phases of the cycle of life of the project of software around 500 problems. The considered effort was of 6,5 [h/h/d] (hour-man-day), the remaining 1,5 hours was destined to the necessary stoppages (routines meeting

of the team of the day, equipment configuration etc). The process of Resolution of Problems adopted has an activity of analysis, with an actor for this activity; three activities of troubleshooters, being an actor for each one of the three different phases of the project; and one activity of editor, with an actor for this activity. In phase 1 requirement survey is included, analysis and structural project, in phase 2 codification, implementation of all the products and tests, finally in phase 3 they are the final tests and the implantation of the product. A typical band of problem raised in the order of 10%, 60% was adopted respectively for each phase, 30%. It is pointed out that the two first phases possess constant interactivity, being that it was considered for phase 1, 10% of problems of primary order (initial), the remaining problems being concentrated on phase 2. For this example, each detected revision between the interactions of phases 1 and 2 approximately 30% of the defects that are in this point are attenuated for phase 2 which increased significantly its tax to 60%. The activities of the troubleshooters are efficient enough, closing in the average of 85% of the problems in the first interaction. Therefore, 15% stand after being revised by the editors still for copyholders, conditioned to the troubleshooters.

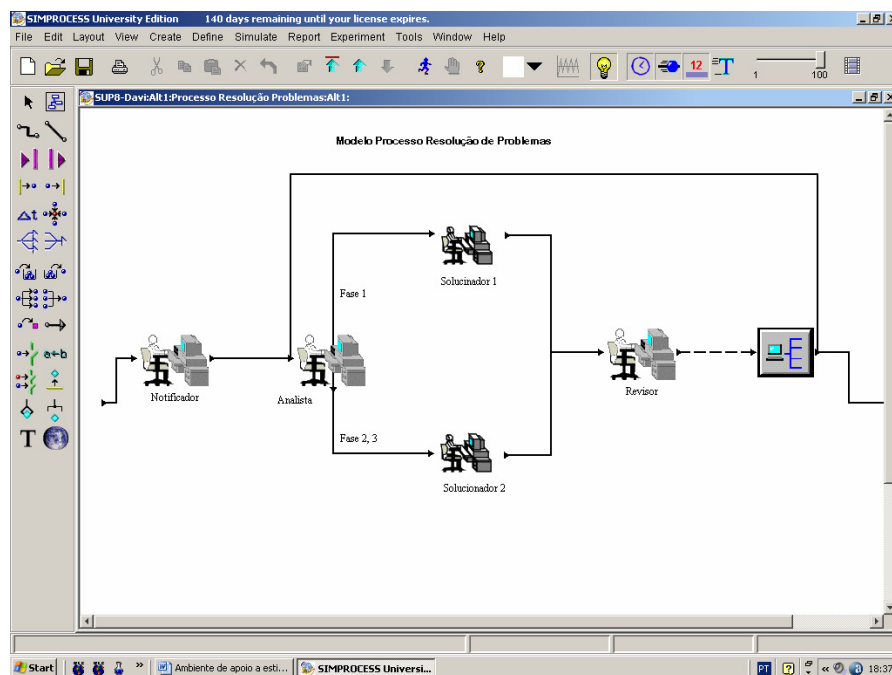


Figure 1. Screen of the logical model of process resolution of problems in the Simprocess® simulator environment. [11]

For this example, the done question was, "considering aggregation of roles of the actors during two first fase, which we would be the impact in the performance process" The additional data in the simulation as to the effort in the accomplishment, ability in deciding the problem, type of the problem (classification as to the phase critical index) and others were limited to in the codes

of the environment, making possible an ample gammut of freedom to modify parameters in the experimentation and in the analyzis of results.

7.1 – Experiments and results

For this work two modalities of simulation had been considered. In the first one the team (human resources) kept it constant in relation to the format of the organization as to its activities i.e., kept the process "as it is"; and in the second modality, the roles that contemplate the activity had been aggregated, aiming at a multi-functional team, i.e. modified the process for "as it would be if". In both cases the indices of productivity, utilization, efficiency and average time of resolution of problems.

Deficiencies of the model of simulator and the executors of processes could have been perceived clearly, as:

- One does not have a standard time for the activities (in function of the three aspects: the individual, the type of the activity, the complexity of the activities);
- no statistics information of any type is offered, as for example, probability distribution in agreement (agreement of the estimates of the managers of the company in study). Aiming the objectives of the research of the Software Engineering Group of the LAC-INPE, is definition of a model of gauging of parameters for these and other aspects, which includes these deficiencies by means of experimental results [01].

Results in this case study emphasizes evidences, such as:

- processes must necessarily be treat continuously, and aspects like efficiency and income must be analyzed jointly (as shown in Table 2, some partial results);
- the lack, or non-availability of a resource in a given moment may to cause the reduction of the performance, which reflects directly on the performance of the process.
- the simulation revealed to be a relatively simple resource to organize and to understand the use of systematic resources.

Table 2. Comparison of the tried alternatives (calculated from the hypothetical data)

Indicator	Original process	Modified process (adding functionality)
Efficiency (ξ) [%]	~81	~90
Utilization (μ) [%]	~74	~85
Planned Productivity ($P_p = \xi\mu$) [%]	~60	~76
Average time of resolution [h]	4,0	2,5

8. Conclusion

There are various options to increase the effectiveness of the analyses, solution and revision of process. To determine new potential results based on the prediction of the results of the simulation. However to evaluate the potential change in more depth and to evaluate also some alternative strategies of the execution it is interesting to take in to consideration resources you add to the inserted models of quality in the engineering processes.

It is fundamental to point out that the types of to be considered changes depend on the organizational culture. What may work for an organization, most of the time many not work for another one. The simulation is a very efficient tool to explore alternatives. Corroborated by diverse authors [04, 05, 07], however they only assist in the aspects that had been considered in the model. Feelings and decisions still are in the reality of the organization and the actors that control and assume them.

The process that was shaped for approaching this work is particularly useful to explore these questions considered qualifying in the process in study. For each potential option, a result in productivity, utilization, efficiency etc can be predicted. Some of these many not be immediate because of staff training, but others may be immediate with the hiring of expert developers.

The approach to this work appears as in the mentioned areas of this work (modeling, simulation and analysis of performance) being quite interesting, therefore it support the metrology of the engineering of software for analysis of performance of software processes, and for the management of processes of software development.

References

- [01] Araújo, D G; et all. (2006). Apoio a estimativas em projetos de software através simulação. II UNEM-WorkShop Colaboração Universidade-Empresa. Taubaté-SP.
- [02] Cereja Junior, M. (2004). A software process coordination service integrated into the e-WebProject-PSEE. (Thesis). INPE, São José dos Campos-SP, Brazil.
- [03] Gonçalves, J M; Boas, A V. (2001). Modelo de Maturidade de Capabilidade de Software. CMU/SEI-93-TR-24-CMM. V1.1.Telecom & IT Soluction. Campinas-SP
- [04] Kellner, M I; Raffo, D M and Madachy, R J. (1999). Software process simulation modeling: Why? What? How? The Journal of systems and software, 46, (2-3), 91-105, 4/15.
- [05] Raffo, D M; Harrison, W and Vandeville, J. (2002). Software Process decision support: making process tradeoffs using a hybrid metrics, modeling and utility framework. Proceedings of the 14th international conference on Software engineering and knowledge engineering. ACM International Conference Proceeding Series. pgs: 803 – 809. July.
- [06] Raffo, D M; Vandeville, J V; Martin, R H. (1999). Software Process Simulation to Achieve Higher CMM Levels. Journal of systems and software, 46, (2-3), 91-105, apr, 15.
- [07] Salviano, C F. (2003). Introdução aos modelos CMM, ISO/IEC 15504 (SPICE) e CMMI. V Simpósio Internacional de Melhoria de Processo de Software, Recife-PE.
- [08] Sant'Anna, N. (2000). Um ambiente integrado para o apoio ao desenvolvimento e gestão de projetos de software para sistemas de controle de satélites. (Thesis). INPE, São José dos Campos-SP, Brazil.
- [09] Software Engineering Institute (SEI). (1993). CMM. version 1.1. Pittsburgh: Software Engineering Institute. (Technical Report CMU/SEI-93-TR-24).
- [10] SPICE - Software Process Improvement and Capability dEtermination. (1993). Software Process Assessment – Part 2 : A model for process management, Version 1.00.
- [11] Simprocess. Available at: <<http://www.simprocess.com>>. Accessed on: Jan. 15th 2007.