Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2000 Proceedings

Americas Conference on Information Systems (AMCIS)

2000

Moving Toward CMM Levels 4 and 5: Combining Models and Metrics to Achieve Quantitative Process Management

David Raffo
Portland State University, davidr@sba.pdx.edu

Warren Harrison
Portland State University, warren@cs.pdx.edu

Moving Toward CMM Levels 4 and 5: Combining Models and Metrics to Achieve Quantitative Process Management

David Raffo, School of Business Administration, Portland State University, *davidr@sba.pdx.edu*
Warren Harrison, Computer Science, Portland State University, *warren@cs.pdx.edu*

Abstract

The ability to quantitatively monitor and assess software project performance supports the Quantitative Process Management and Software Quality Management Level 4 Key Process Areas (KPAs) of the Capability Maturity Model (CMM). In this paper, we discuss on-going work with an industrial partner to integrate software process metrics with a discrete event simulation model to improve process performance predictions. A flexible metrics repository provides data that are used to generate updated simulation model parameters at predefined project milestones. Model predictions are updated using a flexible metrics repository and are compared to Outcome Based Control Limits (OBCLs) defined for the project. These predictions enable a project manager to take corrective action as necessary with the simulation model providing insight on potential performance impacts of the corrective actions proposed.

Keywords
CMM Levels 4 and 5, Software Process, Project Management, Simulation, Software Metrics Repositories

1 Introduction

Timely and accurate data are necessary to provide an accurate picture of where the project currently is and to make a prediction of where the project is likely to go. The ability to quantitatively monitor and assess software projects helps support the Quantitative Process Management and Software Quality Management Level 4 Key Process Areas (KPAs) of the Capability Maturity Model (CMM) [Humphrey 89, Paulk et al. 93].

In a domain where "gut feel" and subjective estimates are common, software project managers have often looked for tools and an approach to provide quantitative data on current project status and quantitative estimates on potential project outcomes.

In recent work, Raffo developed the Process Tradeoff Analysis (PTA) Method. This method builds on previous process modeling work by Kellner et al. at the Software Engineering Institute (SEI) [Kellner 89, Kellner 90] by developing a quantitative approach to evaluating potential process changes in terms of development cost, product quality, and project schedule [Raffo 96]. The core of the PTA method addresses evaluating process alternatives quantitatively by developing stochastic simulation models of each process alternative. These models explicitly capture process-level details including complex interdependencies among process components. The PTA Method has been applied to real-world process change problems at leading software development firms. [Raffo 96A; Raffo 96B; RVM 99]. This work has predominately been applied to the software project management planning function [Raffo 96A; RVM 99].

The goal of our current research is to develop a "forward-looking" approach that integrates timely process metrics with simulation models of the software development process in order to support the software project management *control* function which has a more narrow and immediate focus. We see the management control function as being intended to keep events on course by identifying and correcting deviations from the plan. It is intended to alert managers to significant deviations from the plan while the project is in process.

The forward-looking approach we are developing provides predictions of project performance and the impact of various management decisions. By combining metrics and predictive models, a more comprehensive picture can be achieved than by using metrics alone. In addition, the predictive models can support managers as they attempt to replan and bring a project back on track. This is a key element of quantitative process management and continuous process improvement as specified by the CMM and other industry standards such as SPICE, ISO, TickIT among others.

Central to this approach is the development of a flexible metrics repository which links corporate databases with software process simulation models. This paper reports on work with a leading software development firm to create an approach that includes a flexible metrics repository and a discrete event simulation model based on the company's software development process.

Capturing project-level issues is a critical feature needed to support planning and control activities. In addition, the state-based and discrete event simulation process models [MR 00, RK 95, Raffo 96A] are stochastic and can provide a quantitative assessment of risk. In order to

1. Support planning decisions related to the project and processes being used,
2. Evaluate alternative tailoring variations of different processes and make choices, and
3. Allocate resources among different process sub-tasks,

more detailed models which capture process-level and product-level issues are needed.

By capturing the details related to actually executing software projects, software process simulation models take a very significant step forward in supporting project planning and control activities. This step forward is attained by modeling the software development process to a finer level of granularity and utilizing lower-level project data.

However, the timeliness of data sources for these models (i.e. data obtained from past projects) has remained the same. In order to provide an accurate picture of current project status, up-to-date project information is needed.

The work presented in this paper describes an approach for integrating feedback in the form of process metrics with discrete event simulation models of the software development process. The next section discusses the metrics repository and how it supports the controlling function by providing up-to-date information in a flexible manner. After that, we present an overview of a discrete event simulation model that has been developed. A distinction is made between the representation used by the discrete event simulation model and other process modeling approaches. This difference makes the discrete event paradigm highly compatible with the metrics repository. We then discuss the controlling activity, outcome-based control limits (OBCLs) and decisions supported by our approach. We conclude by discussing potential benefits and future work.

2 The Metrics Repository

Up-to-date project and process information is necessary to support project management and control decisions about a project. The metrics repository provides feedback by storing the necessary information and provides the critical link between raw project metrics and model parameters. Since the simulation model needs to provide a timely view of the project at all times, the repository must facilitate the collection of data on a "real-time" basis.

The repository is based upon a "transformation view" of the software development process. Artifacts such as specifications, designs and code are "transformed" by the application of a "transformation process" into a new artifact. For instance, a design artifact may be transformed into a code artifact by the application of a "programming transformation".

Artifacts possess certain properties, such as "size", "volatility", "complexity", etc. and the transformation process possesses other properties such as "resources consumed", "errors made", etc. The transformation process as well as the artifacts themselves can be represented in the following simplified entity-relationship diagram (figure 1).

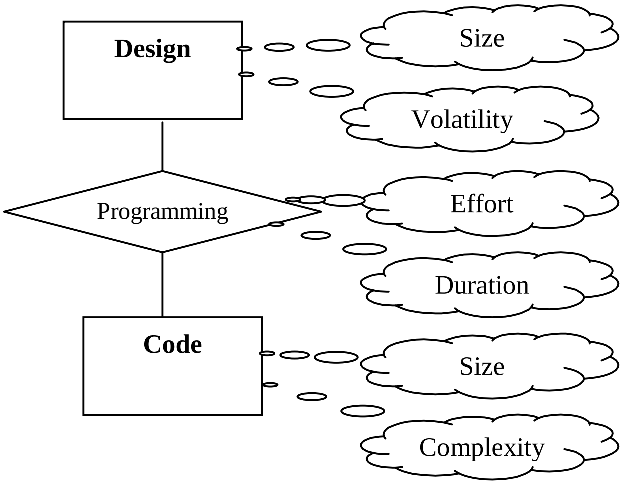

Figure 1 – Entity-Relationship Diagram of a Portion of the Transformation Process

In short, this model denotes that an Artifact is related to another Artifact through some "transforms" relationship. In order to provide an historical record of the state of the project as it progresses through (or is “transformed by”) the development process, we record snapshots of project characteristics each time a *significant* "transformation" occurs. The threshold of *significance* can be adjusted to record snapshots at whatever level of detail is necessary. For instance, snapshots could be taken every time a changed piece of code is checked back into the revision control system, or they could be taken every time the project proceeds to the next phase of the process. The level of significance that is selected will have an impact on the degree to which the repository can be used to provide feedback and support project management and control activities (more detail supports a heightened level of feedback and greater control).

While the repository supports flexibility of data, it also provides a framework for maintaining the current state of the project which provides the necessary foundation for quantitative process management. Depending on the level of granularity and frequency of recording, a project manager should be able to determine the current artifacts being developed, as well as all the pertinent information currently available about preceding artifacts.

The repository is currently implemented using Microsoft Access™ with linkage to the PR-Tracker™ bug tracking system.

3 Discrete Event Simulation Models

This section briefly describes our on-going work with Northrop Grumman Corporation to develop an artifact- or entity-based model that provides a compatible interface to the metrics repository and is linked through parameters that are generated by a set of database queries. These parameters are updated during various significant project *transformations* to provide process feedback that can be used by the model to make improved predictions.

Since mid-1996, Northrop Grumman has been sponsoring research into the use of stochastic simulation models to support software process improvement and quantitative project management issues. The goal of this research is to develop a quantitative simulation model of the software development process that can be used in a forward-looking, predictive fashion to simulate the impact of proposed process changes prior to deployment on software projects. This work is being conducted in collaboration with Portland State University and the Software Engineering Research Center (SERC), a NSF sponsored Industry/University Cooperative Research Center. A new discrete event simulation model has been developed to simulate the activities and artifacts of one of Northrop Grumman’s large-scale software development projects. This model contains cost, schedule and quality data that have been collected from past projects.

Northrop Grumman’s SBMS Melbourne site develops software for airborne radar surveillance and battle management systems. The portion of the software development process modeled consists of traditional software life cycle activities. These activities consist of 71 distinct development steps. The architecture of the simulation model replicates the architecture of the actual software development process in that some activities are executed sequentially, and some concurrently through the use of multiple entities.

When the simulation model is run, parameters for each execution are drawn from populations of data that were collected from the various project teams. Using multiple runs, the model provides the mean and variance of performance results that may be experienced from team to team. Hence, the results of the simulation are stochastic, capturing the inherent uncertainty associated with real-world development. The process model was developed using Extend^{TM} from Imagine That[1].

At SBMS Melbourne the model has been used to analyze several process change problems and to evaluate alternative process configurations for a significant new project bid proposal. The analyses have included flexible “what if” assessments of an initial review of a process change ideas.

One key distinction between the system dynamics and state-based simulation models as compared to a discrete event simulation model is the handling of process artifacts. In the discrete event simulation model, individual artifacts are represented and each artifact is able to retain distinct attributes. In other words, rather than representing a generic code or design artifact, in the discrete simulation model, we represent a particular code module with a certain size, complexity, number of defects, and so forth. It is clear that this added detail is highly compatible with the structure and output of the metrics repository and reflects what actually occurs on a project more accurately. The added detail also provides substantial scope for addressing a number of interesting questions such as: How does the process react if only 20% of the modules contain 80% of the defects? How does the process react if a few code modules are very large or highly complex rather than uniform throughout? What is the effect of a high or low level of fan-out of code modules from design? In short, this representation enables us to look at important questions related to core project management issues.

Using process feedback from the repository enables improved accuracy in the predictions rather than basing the predictions upon initial project estimates of key model parameters (e.g. size and so forth). As will be discussed in the next section, this approach supports active planning and re-planning activities described earlier as part of the project management controlling activities.

4 Combining Metrics with the Discrete Simulation Model to Support Quantitative Process Management and Software Quality Management

The purpose of this section is to illustrate the capability that can be achieved by linking the flexible metrics repository to the discrete event software process simulation model to provide real-time metrics combined with short-term performance prediction. Using this combination, the process provides information to the model in the form of metrics data and model parameters that enable up-to-date predictions of project performance. These predictions create a more sound base (provided by updated project data reflecting the current status of the project) as well as a more accurate trajectory (provided by updated model parameters which are used to predict the future). The model is then used to provide insight regarding which course of action would best achieve the project manager’s objectives of bringing the project back “in control” (see figure2).

[1] Extend is a registered trademark of Imagine That, Inc., San Francisco, California (http://www.imagine_that.com)

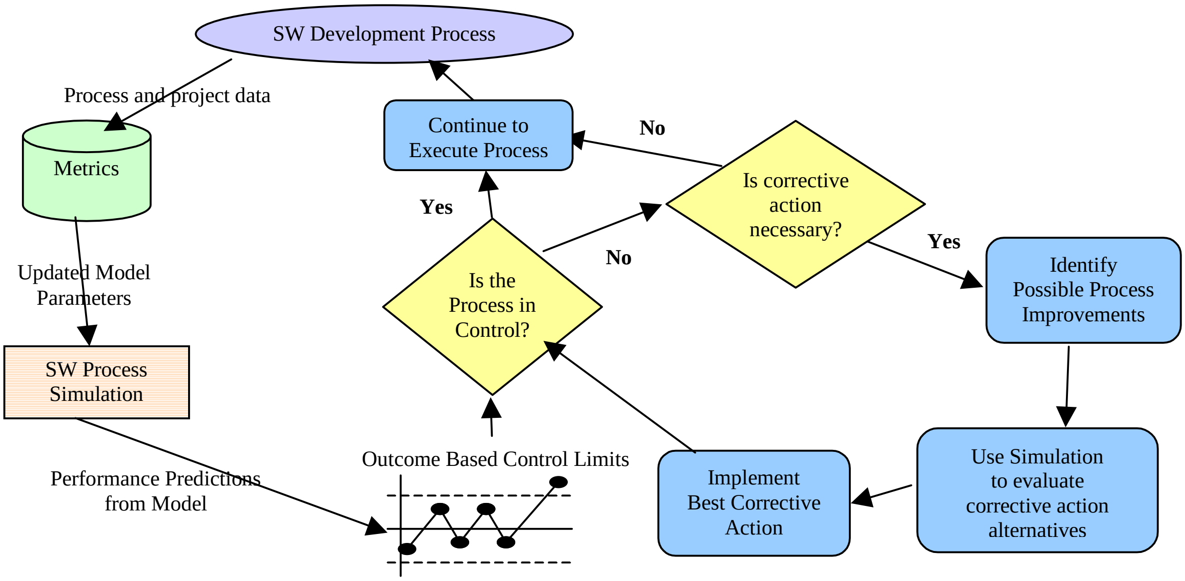
SW Development Process
Process and project data
Metrics
Updated Model Parameters
SW Process Simulation
Performance Predictions from Model
Continue to Execute Process
Yes
No
Is the Process in Control?
No
Is corrective action necessary?
Yes
Identify Possible Process Improvements
Outcome Based Control Limits
Implement Best Corrective Action
Use Simulation to evaluate corrective action alternatives

Figure 2 – Integrating Process Feedback with Simulation Models

We introduce the concept of predicted project performance (i.e. predicted by the model) as being "in-control" or "out of control" – meaning the project "is" or "is not" adhering to the plan within a reasonable amount (reasonable bounds) for the performance measures under consideration (in our model these performance measures are cost, quality, and schedule). This is different from the definition typically used in statistical process control (SPC) where control limits are derived statistically from previous measurements of the process. In SPC, the control limits are determined independently of the desired outcome. We define outcome-based control limits (OBCLs) which are used as guides for assessing whether the project is "in-control" from a project management perspective. OBCLs can correspond to internal performance goals or can reflect contract performance requirements.

The decision as to whether a project is "out of control" requires (a) constant monitoring of the current state of the project and (b) an objective, accurate and meaningful way to compare the current state to the planned state. Software process simulation models address this issue very well. Not only can process models identify changes that will have a significant impact to the project, they also can distinguish deviations from the plan that will *not* affect the project. For instance, although coding on a particular module may begin late, it may have no noticeable impact on the project if it is not on the critical path.

By incorporating up-to-date metrics data with the simulation model, estimated parameters become more accurate, the time frame for the estimate is reduced and more is known about the actual status of the project. In this mode, the model would predict the likely end of project cost, quality, and schedule performance. This performance would be compared to the outcome based control limits. If the project is within the OBCLs, confidence is increased that the current approach will achieve the desired performance targets. This is the primary feedback loop shown in figure 2.

On the other hand, if predicted performance is outside of acceptable bounds, management is alerted that some action needs to be taken. Unlike traditional SPC, the potential out-of-bounds (i.e. out-of-control) situation is recognized ***before*** it occurs, allowing most issues to be prevented rather than "fixed" after the fact. The project manager may have a large number of possible actions that could be taken to bring a project back into control. Information on which options are most likely to be successful, and their relative cost and risk is essential.

When a significant deviation between planned and actual behavior is identified, the project manager can take several steps. First, he can attempt to determine whether the deviation is significant. Computer simulation of the project is used to help predict the final outcome of the project, given the deviation from the plan. The result of the simulation will help the manager decide if the project is truly in trouble.

If the deviation suggests that the project may be in trouble, the project manager can change aspects of the simulation representing various process steps and explore the results of process changes in response to the control deviation. Potential actions to bring the process back under control can be analyzed and compared for effectiveness, risk and cost. This is the secondary loop shown in figure 2.

5 Conclusions

Process feedback is an essential ingredient in process change planning and software project management. In this paper, we discussed on-going work with an industrial partner to integrate feedback from the software development process with a discrete event simulation model to improve process performance predictions. This work is closely tied to our previous process modeling research, which predicts the impact of process changes in terms of cost, quality and schedule.

A flexible metrics repository provides feedback that is used to generate updated simulation model parameters at predefined project milestones. Model predictions using updated parameters and current project data are compared to Outcome Based Control Limits (OBCLs) defined for the project. If the expected performance is outside of the OBCLs, management can be alerted to a potential problem and take corrective action. This is the goal of methods that are based on Statistical Process Control (SPC) but it cannot be achieved using individual metrics alone. The predictive model is then used to evaluate the outcomes of potential management decisions to bring the project back "in control" and help the project meet its pre-determined goals.

This approach directly supports the ability to quantitatively monitor and assess software projects. As a result, this approach supports the Quantitative Process Management and Software Quality Management Level 4 KPAs of the CMM. This also addresses Level 5 KPAs of the CMM related to Continuous Process Improvement.

In future work, we plan to develop a Process Tradeoff Analysis Testbed which will feature the flexible metrics repository and general software process simulation blocks which can be flexibly connected and configured to accommodate a variety of process types and variations. The testbed will support rapid model development and configuration of a company's metrics data into the metrics repository.

6 Acknowledgements

This work has benefited from discussion, comments and participation by Dick Hotz, Joseph Vandeville, and Dean Knudson of Northrop Grumman Corporation. Their contributions are gratefully acknowledged. In addition, the authors are grateful to the Software Engineering Research Center, a National Science Foundation Industry/University Collaborative Research Center for supporting this research effort.

7 References

[Humphrey 1989]Humphrey, Watts, "Managing the Software Process", Addison Wesley, 1989.

[Kellner 89] M. Kellner, "Software Process Modeling Experience," presented at The 11th International Conference on Software Engineering, Pittsburgh, Pennsylvania, USA, 1989.

[Kellner 90] M. I. Kellner, "Experience with Enactable Software Process Models," presented at The Fifth International Conference on the Software Process, Kennebunkport, Maine, USA, 1990.

[MR 00] Martin, R. H. and D. M. Raffo, "A Model of the Software Development Process Using Both Continuous and Discrete Models", *International Journal of Software Process Improvement and Practice,* (In Press)

[Paulk et al. 93] Paulk, M.C., Curtis, W., Chrissis, M.B., Weber, C.V., "Capability Maturity Model for Software, Version 1.1", Technical Report SEI-93-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA., February 1993.

[Raffo 96A] D. M. Raffo, "Modeling Software Processes Quantitatively and Assessing the Impact of Potential Process Changes on Process Performance", Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA, UMI Dissertation Services #9622438, 1996.

[Raffo 96B] David M. Raffo, "Examining the Test Process: Predicting the Impact of a Process Change," Proceedings of the Pacific Northwest Software Quality Conference (PNSQC), help in Portland, Oregon, October 27-29, 1996.

[RK 95] D. M. Raffo and M. I. Kellner, "Using Quantitative Process Modeling to Forecast the Impact of Potential Software Process Improvements", Proceedings of the 10th International Forum on COCOMO and Software Cost Modeling, Held in Pittsburgh, Pennsylvania, October 1995.

[RVM 99] D. M. Raffo, J. V. Vandeville, and R. Martin, "Software Process Simulation to Achieve Higher CMM Levels", *Journal of Systems and Software,* Vol. 46, No. 2/3 (15 April 1999).