

# OPT: Organization and Process Together

Carolyn B. Seaman  
Victor R. Basili  
University of Maryland  
College Park, Maryland

## Position

The role of humans in the software development process must be studied in the context of their organizational configuration. The organizational structure within which the process executes has a profound effect on its outcome. The communication and interaction problems associated with human involvement in development cannot be addressed by process improvement alone. The solution must include organizational improvement as well. Because organizational factors are complex, their analysis is a non-trivial task. Methods are needed for improving both organizations and processes, as well as the relationships between them.

The aim of the OPT approach is the improvement of software development through improvement of organizational structures and processes. The approach is an iterative improvement method. The steps include modeling the relationship between the organization and the process, measuring various properties of this relationship, and choosing organizational and process changes to be implemented.

## 1 The OPT Approach

Some efforts in the study of software development processes have incorporated basic elements of organizational structure in their modeling formalisms [2]. As well, some researchers have examined the role of organizational structure in the management of software development [6]. However, none of these have viewed organizational structure as an object for improvement or as a factor in the effectiveness of the process. Our position is that the effect that an organizational structure can have on a development process is of sufficient magnitude to justify considering these two elements together, as equal parts of a single system.

The OPT approach is based on the combined improvement of organizational structure and development process. The approach is an iterative improvement method based on the Quality Improvement Paradigm [1]. Like the QIP, the OPT approach is a closed-loop improvement cycle. The steps are outlined below:

1. Model the initial relationship between the organization and the process
2. Set high-level project goals, as well as specific organizational and process improvement goals
3. Define constraints which represent management policies and which limit the possible changes
4. Using the model, measure various attributes of the organization/process relationship
5. Based on these measurements, select candidate changes to the organization and process which will contribute to the satisfaction of the stated goals
6. Simulate these changes by applying them to the model, or experiment by applying the changes to some subset of the organization and/or process
7. Re-measure and evaluate the results of the simulation or experimentation
8. Decide which candidate changes are appropriate and update the organizational model accordingly
9. Either iterate back to step 2 to find more candidate changes, or institutionalize the changes in the actual environment.

The result of step 1 is a baseline organizational model, described in the next section. The goal-setting step (number 2) in OPT results in two sets of goals.

The first set contains high-level project goals that concern such issues as quality and productivity. Goals in the second group are defined in terms of attributes of the organization/process relationship. The second set of goals must be chosen so that their satisfaction contributes to the satisfaction of the project goals. Constraints, defined in step 3, specify the conditions that must be maintained while satisfying the goals. Both goals and constraints utilize the OPT metrics, described later, to quantify attributes of the organization/process relationship. The OPT metrics are also used in step 4. They are designed to highlight anomalies, or parts of the environment that are out of the ordinary. This facilitates the selection of candidate changes in step 5. Steps 6 and 7 allow the evaluation of changes before they are implemented. The measurement results of step 7 are used to modify goals, constraints, and candidate changes in subsequent iterations of the OPT cycle.

Central to this process are a set of modeling notations and a suite of metrics. The OPT modeling formalisms are used to build the organizational model which is used throughout the process to capture relevant information. The OPT metrics are designed to quantify this information, and are used to measure attributes of the organization/process relationship, as well as to formulate goals and constraints.

## 2 Organizational Models

Organizational models have three parts:

- an organizational specification, which describes the organizational structure
- a process model, which describes the development process, and
- an architecture specification, which describes the relationship between organization and process

An *organizational specification* represents an organizational structure very simply as a set of nodes and links. The official hierarchy as well as horizontal organizational structures are represented.

The *process model* can be built using any representation that provides certain required capabilities, such as the ability to represent roles. A discussion of process modeling and the various formalisms that are available can be found in a number of references, including [2], [3], and [5].

The *architecture specification* incorporates modeling notations that are unique and new to this work. An architecture specification describes those aspects of the process that impact, or are impacted by, the structure of the organization. This information is an abstraction of information assumed to be available in the process model. Such an abstraction is necessary because most

process models obscure this type of organizational information among other details, making it scattered and often implicit. An architecture specification also provides the bridge between a development process and an organization which allows the two to be considered together as a system.

An architecture specification describes the "agents" that execute the process in terms of the specific "activities" that they perform and the "interactions" between them. Agent definitions include the relevant activities. Each interaction definition describes a piece of information that must pass from one agent to another in order for some part of the development process to execute. Activities, agents, and interactions correspond to objects that appear in most process models (see [4], for example). Specifically, they are mapped to tasks, roles, and artifacts or products, respectively.

## 3 Organization and Process Attributes

Our work thus far has focused on two broad categories of attributes which characterize the relationship between an organizational structure and a process. These categories are responsibility attributes and communication attributes, and were chosen to represent the mutual impact of organization and process. A process can be said to impose certain amounts of responsibility on the members of an organization, so responsibility attributes characterize the effect that the process has on the organization. An organizational structure can either facilitate or hinder the efficient flow of information, or communication, between process activities and participants, so communication attributes characterize the effect of the organization on the process. We have modeled these attributes operationally in order to provide a suite of metrics that can be used to quantify the relationship between a particular organization and process.

*Responsibility* for a particular process activity reflects the extent to which the success or failure of the activity affects the professional success or failure of a participant in the activity. Some responsibility attributes for which OPT metrics have been defined are *commitment*, *type*, and *diversity*. Commitment is the amount of benefit or recognition that someone derives from performing an activity well. Commitment is modeled as a function of task priorities and formality of job descriptions. An OPT metric is defined to quantify it in this way. The metric representing the type attribute takes on the values shared, sole, and managing. Diversity of responsibility is modeled as the number of different process activities for which a person holds some type of responsibility.

Evaluation of the responsibility metrics requires the ability to associate process tasks with members of the

organization. The mappings in an organizational model allow this association. Members of the organization are mapped to architectural agents. An agent is associated with a set of activities, each of which is mapped to a process task. Thus, the metrics can be evaluated by direct inspection of the organizational model.

We have also defined OPT metrics for several attributes of process *communication*, for example the *medium* employed, the *purpose*, and the *organizational distance* between the communicators. Communication media range from verbal messages to documents to meetings. Purpose (e.g. informational or decisional) indicates the importance of clarity and understanding. Organizational distance is the relative position in the organization of those who are communicating. This reflects the difficulty both of communication and of understanding. Organizational distance is modeled to take into account not only the position of the participants in the official hierarchy, but also in any other organizational relationship that is modeled in the organizational specification. Another communication attribute, *interaction effort*, is modeled as a function of the metrics defined for medium, purpose, and organizational distance.

Process communication is represented in an organizational model by the interactions in the architectural specification. Each of these interactions describes some instance of communication that is required by the process. The OPT metrics defined for communication attributes can all be evaluated with the information in these interaction definitions.

## 4 Formulating Goals and Constraints

The metrics described above are used to characterize the environment under study. Another important use is the building of OPT goals and constraints. OPT goals are set during the OPT improvement cycle in order to guide the selection of candidate changes to the organization and process. OPT constraints are also formulated during the OPT cycle, as part of the characterization of the system. They are used to constrain the possible choices of candidate changes.

Goals and constraints are formulated using the metrics described in the last section. Relationships are defined between the metrics which describe the responsibility and communication attributes, as well as metrics which describe other attributes of the development environment. Constraints specify which of these relationships must be preserved. Goals specify which of these relationships must change. How the relationships change, and by how much, is what defines improvement.

For example, suppose that a particular software development environment identifies a problem with slipping deadlines. A decision is made to address this prob-

lem by identifying and focusing on those process activities on the critical path. In step 2 of the OPT approach, the project goal chosen is to increase the commitment of higher-level members of the organization in critical-path process activities. This project goal translates to the following OPT goals, stated in terms of the metrics defined previously:

- Increase the average organizational level of those who hold responsibility of *type* "managing" for activities on the critical path
- Increase the average *commitment* for critical-path activities of members of the organization with a level more than  $x$

At the same time, assume that there is a concern about overloading middle managers. A general policy is drafted stating that the number of different activities that a manager has responsibility for is bounded, but increases with the manager's level in the organization. This translates to the following constraint:

- Any member's *diversity of responsibility* is bounded by  $y$  times the member's organizational level.

This simple example illustrates the way in which OPT metrics are combined with other measurable properties to formulate goals and constraints.

## 5 Conclusions

This paper describes the OPT approach for organizational and process improvement of software development. This method is meant to be part of a continuous improvement program, and is modeled after the Quality Improvement Paradigm. The approach includes mechanisms for modeling the relationship between an organizational structure and a development process, for measuring this relationship quantitatively, and for using this information to plan specific improvements to the environment.

The OPT approach also provides a way to study the role of humans in the software development process by taking into account the organizational structures within which they work. We recognize that organization and process have a profound mutual effect on each other. Accordingly, they should be studied together as a single system.

This paper describes very early results of the work on organizational modeling and analysis. This area promises to provide the software industry with tools for significant change and improvement in the way that software is produced. It is part of the broadening of focus from strictly product-centered improvement approaches to the measurement and improvement of non-product factors. This trend in the field reflects a maturity in software development that will bring further advances in software quality and productivity.

## Acknowledgments

The authors would like to acknowledge the contributions to this work of Gianluigi Caldiera and Walcelio Melo, and the support of IBM Canada Laboratory's Centre for Advanced Studies.

## References

- [1] Victor R. Basili and H. Dieter Rombach, "The TAME Project: Towards Improvement-Oriented Software Environments", *IEEE Transactions on Software Engineering*, Vol. 14, No. 6, June 1988.
- [2] Bill Curtis, Marc Kellner, and Jim Over, "Process Modeling", *Communications of the ACM*, September 1992, Vol. 35, No. 9, pp. 75-90.
- [3] G.E. Kaiser, N.S. Barghouti, and M.H. Sokolsky, "Preliminary Experience with Process Modeling in the Marvel Software Development Environment Kernel", *Proceedings of the 23d Annual Hawaii International Conference on System Sciences, Vol. II-Software Track*, IEEE Computer Society, Washington, DC, 1990, pp. 131-140.
- [4] Christopher M. Lott and H. Dieter Rombach, "Measurement-based Guidance of Software Projects Using Explicit Project Plans", *Information and Software Technology*, Vol. 35, No. 6, June 1993, pp. 407-419.
- [5] K. Narayanaswamy and N. Goldman, "Team Coordination: Information Sharing + Policies", *Proc. of the Int'l Software Process Workshop*, San Francisco, CA, October 16-18, 1991.
- [6] Walt Scacchi, "Managing Software Engineering Projects: A Social Analysis", *IEEE Transactions on Software Engineering*, 10:1, January 1984.