A FRAMEWORK FOR IMPROVEMENT PROGRAMS
Modeling Optimal Supplier - Customer Interactions

Tom Gilchrist, **CQA**, 206-226-3708 (tomg@halcyon.com)
Ron Nelson (74672.2224@compuserve.com)

Presented at the Forth International Conference on Software Quality (ASQC) October 4, 1994, Washington, DC

Introduction

Many companies and organizations have improvement programs that are intended to improve their competitive position in meeting the stated aim of the organization. However, many times the effectiveness of an improvement program falls far short of expectations. This can be very frustrating. Often the cause of ineffective improvement programs can be traced to incomplete and ineffective communications with the customer of the products and services of the supplier.

This paper presents a graphic framework or model of customer/supplier interactions in very mature environments. It is built on a graphic framework presented at last year's 3ISQC conference in our paper, "*A FRAMEWORK FOR MEASUREMENT PROGRAMS.*"[1]. We have found that the ideas presented here can be used by anyone, whether management or practitioner, to improve the effectiveness of measurement and improvement programs.

Our position for this paper is that the success of improvement programs is ultimately tied to the maturity of the consumer of your products and services and the health of the communication paths between supplier and customer.

The models presented in this paper have been used over the past several years in an environment of engineers, engineering management, software developers and business management. We have seen the models used many different ways to interpret different project management programs. It is often used as a shorthand to explain a sometimes foreign and complex process. It is also used as a graphic checklist of components and attributes required for improvement activities. Once the model has been presented and used once, the graphics allow people who are swamped with information, to quickly recall the fundamental concepts. In this way, they can get down to the important issues without having to relearn the basic concepts each time metric, improvement, and communication issues are presented.

However, these models are not intended to represent physical organizational structure. The concept of having a separation of process improvement and process usage is important because they are two separate tasks that need to be managed. They may or may not be done by the same people. Also, the models do not attempt to characterize how organizational memory or the resulting Professional Practices are stored, accessed, maintained, and archived.

This paper explains the models and concepts in much the same way as we do. We call the basic model the "Single Engine Model" and always present it first. We present an extension oriented toward process memory. This extension we call the "Two Engine Model". The "Four Engine Model" is used to describe the importance of having sound customer communications and relations. The model is extensible and we have been amazed at the different ways people use it as a fundamental building block. Because the Single and Two Engine Models where discussed in detail last year, we will only briefly review their most important features.

Single Engine Model

Figure 1 presents the framework we call the "Single Engine Model". It is a complete feedback loop. It is used to illustrate the need to effectively couple actions with measurements.

PROCESS is where the needs and constraints of the customer are translated into deliverables. It is monitored by MEASUREMENT and its behavior is influenced by ACTION.

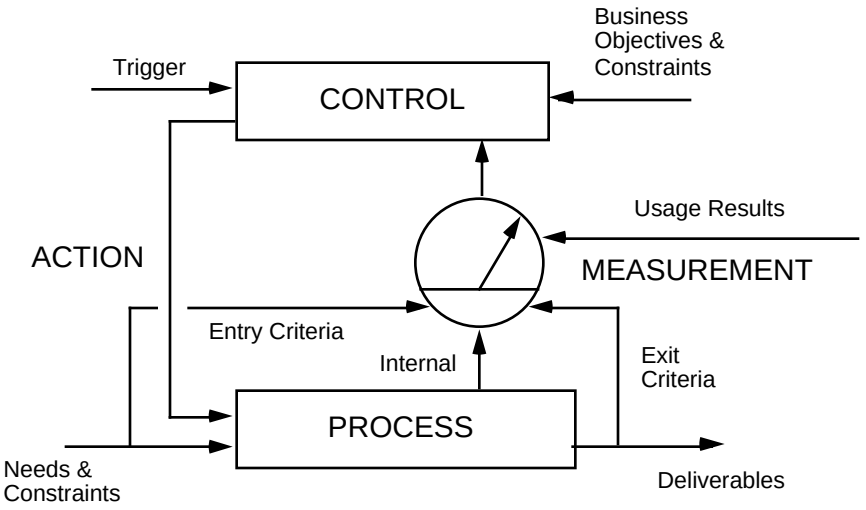

Figure 1. Management with data framework, the Single Engine Model

CONTROL is the decision making mechanism which uses MEASUREMENT and external Business Objectives and Constrains to direct ACTION. The Trigger signal is used by CONTROL to know when to start and stop the engine.

The MEASUREMENT meter indicates that there are a variety of ways to collect evidence regarding the state of the target PROCESS, the product it creates and results or satisfaction of the use of the deliverables.

If any of the components of the Single Engine Model are missing or not working, the model can't effectively operate as a feedback loop. In other words, it can't correct for observed error or performance. This can either be caused by not knowing the current state of the PROCESS (no MEASUREMENT) or not issuing corrective actions (no ACTION).

Organizational Memory

The Single-Engine model of Figure 1 does not explicitly show a "memory" of previous experiences as a basis for improvement. This is typical of project level behavior; a project starts, ends, and forgets. Reinvention is then necessary. Without such formal memory, there may be constant change, but improvement is unlikely and unverifiable. There must be a repository of practices and this is to be improved through experience. Process memory is one way to remember those things that work (so we can do them again), and those things that don't work (so we can avoid them).

One such repository is in the experiences of the practitioners. We refer to this as "individual professionalism". Unfortunately, this is not a reliably shareable memory. An obviously superior repository of lessons learned is a documented statement of Professional Practice for the group or organization. This documented description can be shared between workers and rationally analyzed. However, the notion of organization must be larger than a group of people doing a task or project. Projects have the tendency to come and go. Figure 2 illustrates that an organization needs to be defined in such a way as to retain past experiences (to have a memory). The use of documented professional practices can be an instance of "organizational professionalism".

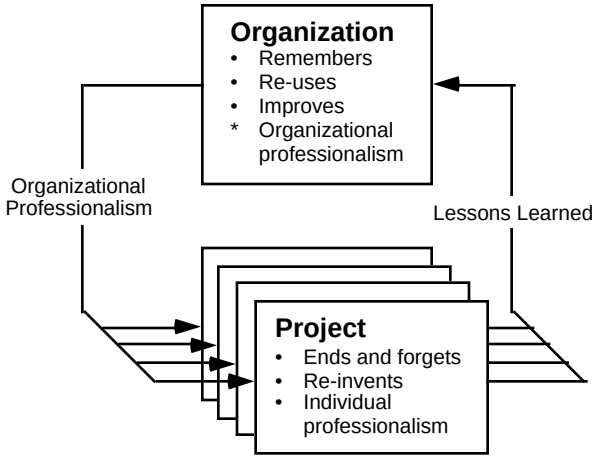

Figure 2. Organizational memory.

The projects in Figure 2 represent those of the past and of the present. Lessons learned, a body of experience based on organizational practices, constantly accumulates in the organizational "memory" of documented professional practices.

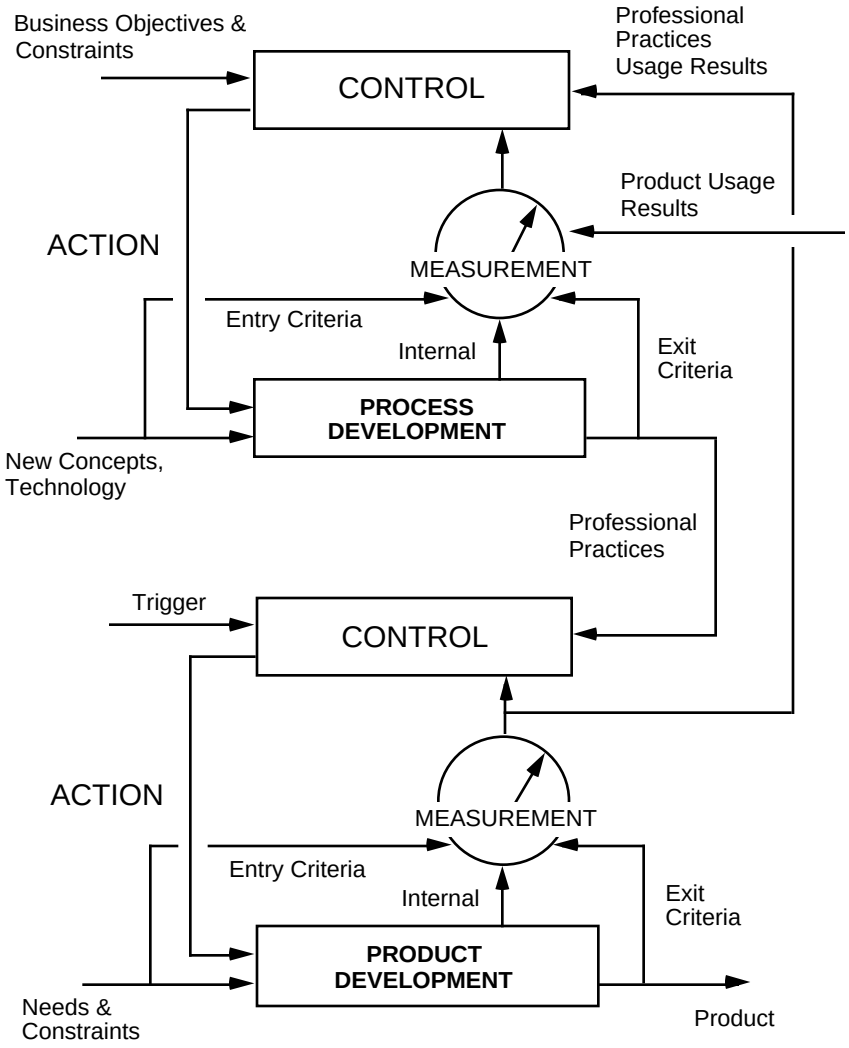

Figure 3. The Two Engine Model

Two Engine Model

The Two Engine Model, shown in Figure 3, consists of a pair of single engine models in feedback communication with each other. Briefly, the bottom engine creates the target **product** for some customer. The top engine creates and improves the **processes** used by the bottom engine. To emphasize the feedback: the top engine communicates **Professional Practices** to the bottom and the bottom engine communicates **experiences** to the top where they are incorporated into the Professional Practices of the future.

The block diagram in figure 2 can be thought of as a simplified Two Engine Model shown in Figure 3. Figure 4 shows this relationship. For the purpose of clarity in this paper, we will use the block diagrams. However, the complete graphic is used when we use the model in detailed discussions. The block diagram also illustrates the leverage of a single organizational improvement effort over **many** development efforts.

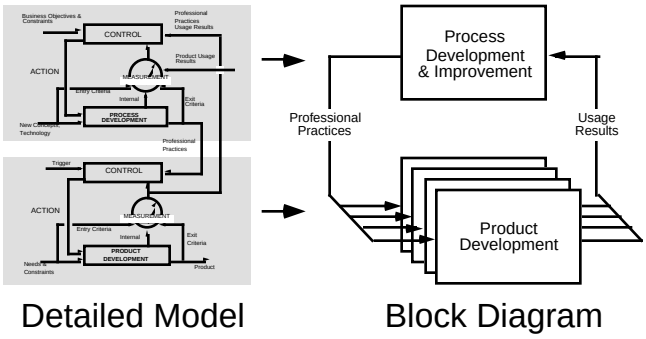

Figure 4. Detailed vs. Block Two Engine Model

Four Engine Model

In organizations trying to improve their process maturity some succeed and some seem

to get nowhere. The success of the software organization is tied not only to its process maturity, but also to the maturity of their customer. If one assumes that any mature organization can be modeled using the Two Engine Model of Figure 2, then one can model the relationship between a software developer and an organization using the software by using two interacting sets of Two Engine Models as shown in Figure 5.

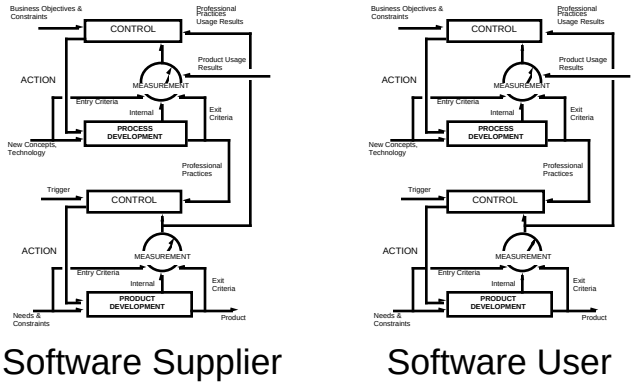

Figure 5. The Four Engine model

An interacting pair of Two Engine Models now becomes what we call the Four Engine Model. This model represents two mature organizations, each producing product and each using the constant improvement of professional practices based on the lessons learned in their use.

To show the connections and relationships between these two high maturity organizations, we will use block diagrams to make the communication and product flows clearer.

Four Engine Connections

While there are a number of interactions that can be described using the Four Engine Model, there are three that demonstrate the importance of the relationship. First, how do software requirements flow from a high maturity software customer to a high maturity software supplier? Figure 6 shows that the "Requirements" flow from the top process improvement engine of the software user to a product development engine of the software supplier. These requirements are based on the memory of past experience of the customer organization.

For instance, based on experience, the customer has determined that it takes X units of resources to do certain tasks when the current Professional Practices of the software user's organization are used. An improvement team had estimated that they can reduce the amount of resources for the process if software is used. This is the information passed to the software development organization. The requirements come from the desire to improve the customer's organizational performance.

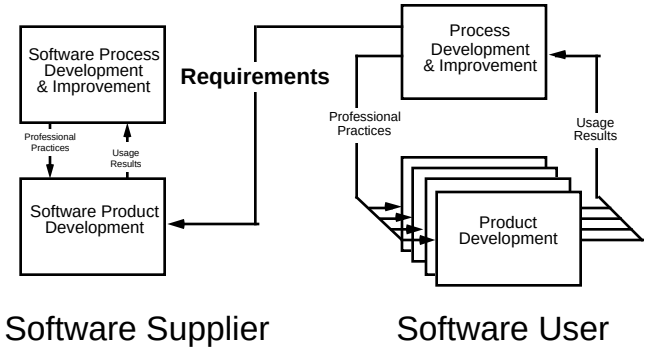

Figure 6. Software Requirements

Of course, not only business requirements flow on this path. Customer required standards, detailed specifications and constraints also flow along this path.

If the software supplier builds the software that will satisfy the customer requirements, what path does the software take? In high maturity organizations, the product flows on the path in Figure 7. The software is developed by the software supplier using his organizations professional practices, delivers the software to the customer's process improvement engine where it is delivered to the customer's value added development processes as **new Professional Practices.**

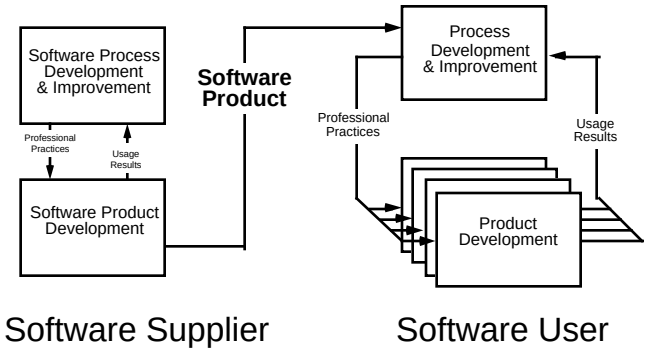

Figure 7. Software Product

The result of the end-use of the software is communicated back to top engine of the mature customer. If changes need to be made to the software, requirements are again communicated to the software supplier as in figure 6.

A third line of communication is used when the software supplier has the goal of improvement and wants an answer to the question "how well do our current professional practices work"? While the data and satisfaction from developers is important, the results of their use in practice can only come from the user of the software, the customer. Figure 8 shows the "Results and Satisfaction" flow from the top improvement engine of the customer to the corresponding top improvement engine of the supplier.

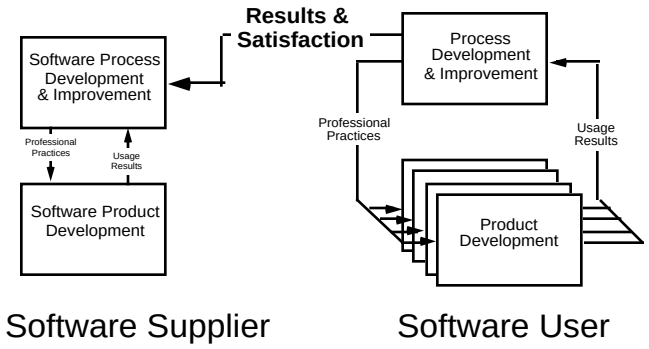

Figure 8. Software Improvement

In the same way that organizational memory is important in the Two Engine Model, remembering the results of usage of the software product is important in accurately answering the question "how well does our software work in meeting the needs of our customers"? Without this formal organizational memory on the customer's side, customer satisfaction and usage information will be limited to either a subset of current usage, or the loud voices of individuals who might not represent true usage information.

Maturity Transitions

In the table in Figure 9, we have summarized how suppliers and users/customers affect each other when they are faced with high or low level maturity partners. The term "Supplier Impact" is meant to characterize negative impact. Regardless of the maturity of the software customer, any combination except where both organizations are at a high level of organizational maturity causes problems with the supplier.

	Supplier Impact	User Satisfaction	Dynamics
Low M Supplier Low M User	High	Low	Personalities Dominate
High M Supplier Low M User	High	Low	Perception of Bureaucratic Behavior
Low M Supplier High M User	High	Low	Somebody will change
High M Supplier High M User	Low	High	Consensus

Figure 9. Maturity Transitions

When a low maturity supplier and user interact, the result is low customer satisfaction and constant problems for the supplier. Since there are few (if any) organizational professional practices, problems are blamed on individuals. In this environment, success (when it comes) is usually attributed to the heroic efforts of a few key individuals. Success and failure is determined by individual professionalism (or lack thereof).

When a high maturity supplier interacts with a low maturity user, the customer is unhappy. Unable to understand why the high maturity supplier does things the way they do, the customer has the perception that the supplier is interested in "bureaucratic nonsense". The customer attributes this behavior to many things. Of course, the supplier sees no end to stream of changes to requirements and the inability for the customer to speak with one voice.

If the tables are turned and the customer is high maturity and the supplier is low maturity, the customer is still unhappy, but for different reasons. Because the customer is accustomed to organizational professionalism, he becomes annoyed at the seemly endless variation of the suppliers support. The software supplier is constantly struggling to live up to the expectations of a customer who seems to "have his act together." Eventually, the mature software user is likely to dump the supplier. In any case, if the parties stay together, somebody will change.

A broken relationship is not always the outcome of partners of different maturity. One of the parties can change. If a software organization really wants to improve their organizational maturity, they should find a customer with a high level of organizational maturity. Of course this is not always easy. Customers who have obtained high levels of

process maturity have probably learned that low maturity venders cause problems.

High maturity software organizations should also stay away from low maturity customers. There is always the possibility that interacting with a low maturity customer will have the effect of eroding the organizations Professional Practices. In any case, when the software supplier and software user are of two different organizational maturities, if they stay together, someone will change.

Four Engines or One Team?

If one trys to connect the flows to a detailed view of the Four Engine Model as in Figure 5, you will see that there are no flows to connect to on the software user side that correspond to "Requirements", and "Software Products". This is because the activity of developing software is really part of the box "Process Development" in the mature customer side of the Four Engine Model.. This notion of the supplier and the user being one team having a common goal is true in high maturity organizations. This type of relationship is built on a foundation of trust and respect and is not easily abandoned by either party. While some organizational and business goals might be different, the central aim of the software supplier is to help the software user succeed and thus remain an ongoing customer of its products and services.

Ownership Roles

Mature organizations and relationships don't happen by chance. Energy and direction are needed to move from low levels of maturity to high. Also, once at a higher plateau of maturity, some energy and control is required to maintain and improve. While the models presented in this paper are not intended to reflect an actual business structure, the role of "CONTROL" in each engine can be thought of as a way to describe the roles and responsibilities of a manager.

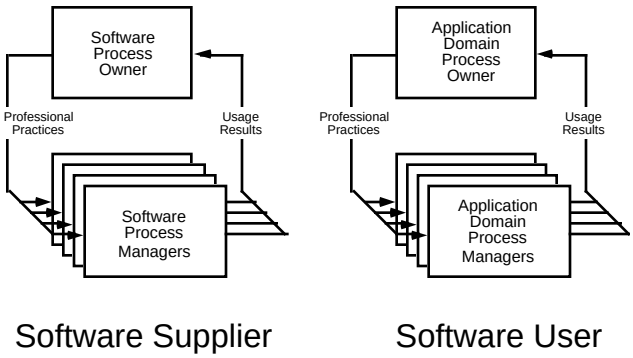

Figure 10. Management Roles

Figure 10 suggests that the top engine of the Two and Four Engine Model, describe the roles and responsibilities of a "Process Owner". These processes are part of the Professional Practices that are used in the bottom engines by "Process Managers".

The working definition of a process owner and a process manager are never clearly described and communicated in immature organizations. The detailed two engine model clearly describes the activities and scope of a process owner.

Conclusion

High maturity supplier - customer interactions need to resemble the Four Engine Model described in this paper. The model incorporates both the idea of the separation of process and product development and the concept of process memory which is institutionalized in documented Professional Practices.

For improvement to happen, an organization needs to be able to evaluate their current maturity and have a framework to build upon. The success of an organization to improve is tied to the maturity of the customers you have. It is not enough just to have customer involvement. Without this infrastructure, lessons learned are never remembered and an organization repeats the same mistakes over and over again and can actually degrading current practices by sapping organizational morale, energy, and resources.

END OF ARTICLE