

A FRAMEWORK FOR MEASUREMENT PROGRAMS

Tom Gilchrist, CQA, 206-226-3708 (tomg@halcyon.com)

Ron Nelson (74672.2224@compuserve.com)

*Presented at the Third International Conference on Software Quality
(ASQC) October, 1993
Lake Tahoe, Nevada*

Introduction

Many companies and organizations have metric programs which are supposed to drive improvement. However, many times the effectiveness of a change program falls far short of expectations. This can be very frustrating. Often the cause of ineffective measurement programs can be traced to a misunderstanding of the purpose of a measurement program and management's role in its effective operation.

This paper presents a graphic framework for understanding the components necessary for the successful use of data in decision making. We have found that the ideas and concepts presented here can be used by anyone, whether management or practitioner, to improve the effectiveness of a measurement program. Our position is that measurement must be justified by the decisions which require it and that these decisions must be clearly visible to all involved in the measurement program.

Data is sometimes requested simply because someone believes that by merely gathering the data, things will improve. This is naive. What we really want is improvement. We want change with the key goal of real, measurable improvement. Knowing when and what to change will separate teams which will succeed and prosper from those which will be out looking for jobs and wondering what went wrong.

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 management with data programs. 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 internalize the basic concepts each time metrics issues are presented.

This paper explains the model and concept in much the same way as we do. We call the basic model the "Single Engine Model" and always present it first. The model is extensible and we have been amazed at the different ways people use it as a fundamental building block. We present an extension oriented toward process improvement; this extension we call the "Two Engine Model".

Measurement Customers

There is a simple, yet very important framework for effective management with data. This is the feedback control concept. Perhaps the best way to motivate our discussion is to depict the behavior of some pathological "management styles" we have often observed over the years.

There have always been those who want to direct change and each has had to express what they wanted to happen. There has typically been little concern with measuring results. The situation is best described by the phrase "Royal Dictates". Figure 1 depicts this behavior. First there is the box labeled "CONTROL". This box represents the agency with the authority to direct a change. This role is sometimes called the process owner or process manager. The "ACTION" represents the change ordered. The bottom box represents the target of the action. In our example it contains those who are to effect the change, the "PROCESS". The "PROCESS" is where the needs of a customer are transformed into deliverables for their end-user.

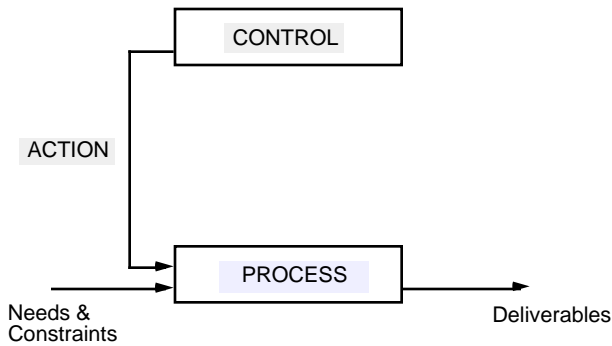


Figure 1, Royal Dictates

The Royal Dictate involves no feedback from the PROCESS to CONTROL. There is no way that CONTROL can reliably determine the outcome of the ACTION or, in many cases, even if anything actually occurred. A lot of pain and confusion can result from Royal Dictates, but improvement is nearly impossible to demonstrate.

Today, managers know that successful organizations have measurement programs. Some conclude that if they are to be successful, they too need one. Figure 2 depicts behavior best described as "measurement for the curious". This arises when CONTROL asks for data about the PROCESS, but has no visible plan for using the data; it appears that CONTROL is merely curious - or worse, has a hidden agenda. The phrase "what gets measured, gets done" is sometimes wrongly equated with this situation. Measurement for the Curious annoys if the data is not used, for its collection involves additional work. It creates suspicion (and possibly program sabotage) if later used for purposes initially unstated.

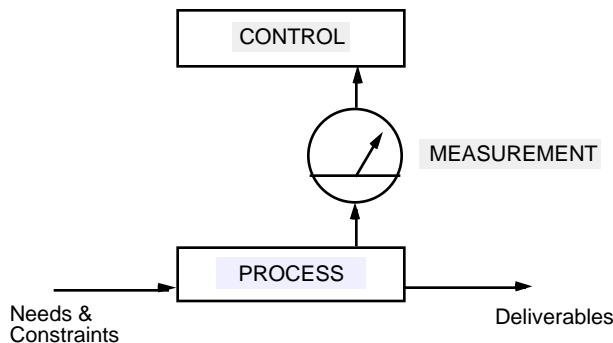


Figure 2, Measurement for the Curious.

In Measurement for the Curious, it often makes sense for PROCESS to spend as little effort as possible and still meet the measurement request. There is little reason to do more.

Most initial measurement programs are really "guessing" programs and the quality of data

gathered is typically poor. The use of the data for decision making has not been planned or communicated.

Single-Engine Model

We need a way to effectively couple actions with measurements. Figure 3 presents the framework we call the "Single Engine Model". It is a complete feedback loop with components and detail not found in Figures 1 and 2. To better understand the model and its implications, we describe each major component.

- **PROCESS**

The fundamental process transforming Needs into Deliverables and results. It is monitored by MEASUREMENT and its behavior is influenced by ACTION directed by CONTROL.

- **MEASUREMENT**

The quantitative evidence regarding the state of the target PROCESS and its products. There are Process and Product measurements and satisfaction of downstream customers who communicate product Value.

- **CONTROL**

The decision making mechanism using MEASUREMENT, Goals, and Constraints to direct ACTION.

- **ACTION**

The response of CONTROL to influence the target PROCESS in desirable directions.

- **CYCLE**

Dynamic characteristics of the model such as cycle time.

PROCESS behavior depends on ACTION

The PROCESS transforms needs into tangible deliverables. In considering the process performance, several factors are of importance:

- A process must exist at the organizational level indicated by CONTROL.

In the absence of such a process, one can only control people directly, a poor basis for quality improvement.

- The process must exhibit measurable behavioral effects in response to the available actions.

Otherwise effective control of the process is

MEASUREMENT depends on Process Internals, Entry and Exit Criteria, and Value

Measurement can involve key indicators inside the process, the Entry and Exit Criteria of the products required or produced by PROCESS, or the value or satisfaction derived from the product customer. In considering appropriate measurement, several factors are of critical importance:

- The measurements selected must, at minimum, reflect the degree to which the goals of the process have been achieved.

Otherwise “improvement” can’t reliably be

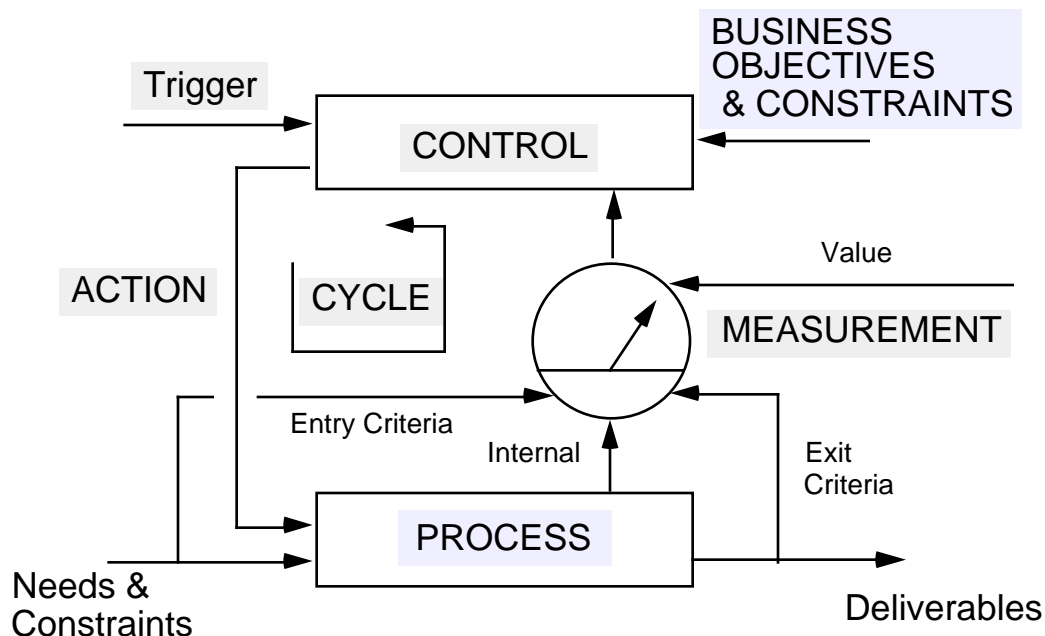


Figure 3, Management with data framework, the Single Engine Model

impossible by the available actions and it is impossible to associate cause and effect .

- It must be known - or at least determinable from experience - what process performance trends are likely to be associated with available actions.

Otherwise, the available actions lack predictability. This would suggest the available actions are not strongly correlated with process performance.

The main point to remember is that **process is fundamental**. Without a clear understanding of what PROCESS is being defined, we can't define Deliverables. If we can't define Deliverables we can't rationally discuss Value with the customer or any other part of the model.

demonstrated by measurements.

- The measurements selected should, if at all possible, include some which quickly respond to actions taken.

Otherwise the lag times may make it difficult to associate cause and effect. Rapid confirmation of improvement is important to motivation.

- The measurements selected must be sufficiently reliable that action can reasonably be based on them.

Otherwise “managing with data” is an illusion. Guessing and vagueness in definitions are often the cause of unreliable metric data showing excessive variation.

CONTROL depends on timely and reliable data.

The decision making mechanism of the model is found in CONTROL. It is needed to explain management's role. The role of control is to receive the goals and measurements, analyze the data and available actions, choose the appropriate action, and finally to select the appropriate time to execute this ACTION. The Trigger input is the event which causes the process to begin and sometimes defines when a process ends.

In considering appropriate CONTROL, several factors are of importance:

- Explicit goals and constraints must be known to the control agency.

Otherwise improvement is undefined and ACTION is without direction.

- A forum must be specified in which measurement is presented to the CONTROL agency.

Otherwise the communication of the state of the process to the CONTROL agency is not reliable and the model degenerates to Royal Dictates.

- Conditions must be stated under which specified measurements are presented to the control agency.

Otherwise the data may not be timely or may not be relevant.

ACTION depends on GOALS and MEASUREMENT

Effective actions depend on measurement of system performance as well as the goals we're trying to achieve. In considering appropriate action, several factors are of critical importance:

- A range of actions must be available.

If there is only one possible control action, there is no decision to be made. It is our position that measurement programs can only be justified by the decisions they support.

- A forum must be presented in which ACTION is presented to PROCESS.

Otherwise the communication of the ACTION to the PROCESS is not reliable and the model degenerates to Measurement for the Curious.

The actions taken must truly be based on the stated goals and measurements.

If not, we are not really managing with data, but are driven by hidden agendas and opinion.

CYCLE Attributes

This is an important component of the model and is often ignored when trying to change or improve things. CYCLE describes how the engine works over time. In considering the cycle characteristics, several factors are of critical importance:

- The time required to initiate an action in response to changes in goals or measurements must be small compared to the response time of the target process.

Otherwise the available actions aren't timely in controlling the process.

- The time required to complete an engine cycle should be small in comparison to other changes in the environment.

Otherwise changes in needs, constraints, business objectives and even process technology will render measurement ineffective.

Shared Memory

The Single-Engine model of Figure 3 implicitly assumes the existence of a "memory" of previous cycle experience as a basis for improvement. 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.

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. This documented description can be shared between workers and rationally analyzed. The use of Professional Practices can be an instance of "organizational professionalism".

We next turn to a model of process improvement. We assume the processes are documented in a group or organizational statement of Professional Practice. These practices are to be improved and improvement is to be based on the evidence of our measurements.

Two Engine Model

The Two Engine Model, shown in Figure 4, 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.

All processes (Professional Practices) should have parameters which enable tailoring to satisfy local needs. These parameters are controlled in the bottom engine.

We comment on and illustrate some especially important points of the Two Engine Model in the following sections.

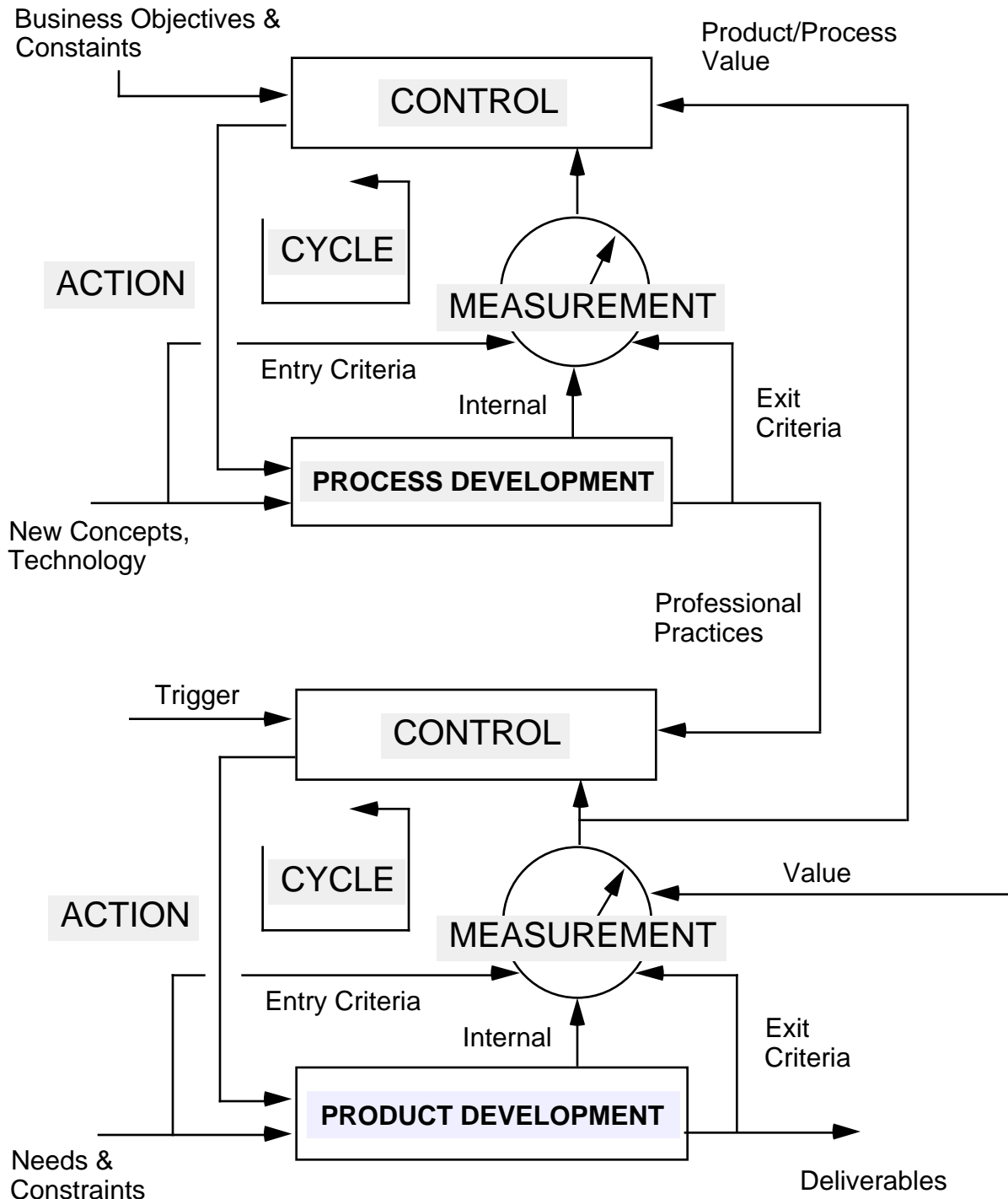


Figure 4. The Two Engine Model

Customer

The bottom engine produces deliverables for some customer who responds with a level of satisfaction or Value. It is essential to note that the model does not imply that the customer is the ultimate end-user of the product or service being created. Here are two example possibilities:

- The bottom engine produces a computer program. The customer is the end-user for that program.
- The bottom engine produces a requirements document. The customer is a software designer.

Process Improvement

Process improvement occurs in the top engine. It results in modifications to Professional Practice which are then used by the bottom engine. Process improvement uses the reported Value as seen by the customer as well as product development experiences. The concept of value is customer dependent. It typically involves technical quality, cost and schedule considerations. Therefore, process improvement is driven by that evidence, together with new concepts and higher-level business guidance.

In general, a given bottom engine can utilize professional practices from several top engines, each specialized to processes specific to a certain domain. For example, a software development process could utilize coding standards, a project planning process standard and a configuration management process standard.

Similarly, one top engine may produce professional practices which are used by many bottom engines. For example, there could be many software development processes using the same project tracking and oversight standard. The same top engine could be collecting data on the results of the professional practice use and be continuously improving the standard.

Even where both the bottom engine (product development) and the top engine (process development and improvement) are being done by the same individuals and controlled by the same manager, the two engine model still is of value. By understanding the difference between using a process and improving it, the team will be able to differentiate between process changes for improvement and change for the sake of change.

Internal and External Process Measurement

According to the model, there are two general types measurements which can be taken. Those which need knowledge of the inner workings of the PROCESS and those that don't. Some examples of external measures include:

- **Entry Criteria:** A definition of needed input to the PROCESS and their suitability to the task. For instance, a requirements document is needed by PROCESS and it must be in writing.
- **Exit Criteria:** A definition of external deliverables produced by PROCESS and measurable attributes of cost, schedule and quality. For instance, source code is produced by PROCESS and the functionality is traceable to the requirements.
- **Value:** The results of the use of the products of PROCESS as reported by customers or end-users. For instance, customer satisfaction surveys are used to assess the way users view the relationships of cost, schedule and quality. Do they think that they are overpaying for the quality they are receiving?

On the other hand, internal measurements are taken within the development process and relate to intermediate products and process characteristics. In Figure 4, the measurement line marked "Internal" represents such measures. If present, these measures imply that we know something of the internals of PROCESS. For example, if PROCESS is using Professional Practices, we can measure the penetration and compliance of the Professional Practice. If the work flow internal to PROCESS is known, we can measure the difference between actuals and estimates for cost and schedule.

Cycle Time

The top and bottom engines need not have the same cycle times and these need not be synchronized. The top engine produces Professional Practices and may not respond immediately to input data changes. There are reasons why excessive responsiveness might not even be desirable. For instance, it may be necessary to know if the bottom engine

experience embodies statistically valid trends or results from special causes.

Limits of a Model

The single and two engine model are useful in communicating and remembering the important components and attributes of improvement or change program. However, these models do not necessarily represent the physical organization of a software development and improvement group. For instance, in a real development organization, there might be many Product engines producing deliverables and each having many customers. Also, there may be a number of Process engines (top) controlling any given Product engine (bottom).

The model does not detail other important characteristics of a successful software organization, for example:

- The incentive process used to motivate the various players
- The processes used for deployment and training.
- The process of the customer in reaction to the products and deliverables of the supplier. Model extensions to this area have been created and are interesting, but are too lengthy to address here.

Conclusion

We have found that once the concept of the Single Engine model is understood, the ideas of the Two Engine model are quickly internalized by any audience. We have seen the models used to describe many different types of processes all with similar results. By identifying what customers and deliverables are of interest in the bottom engine and by understanding who and what will control change, effective measurement programs can be put in place. Without this infrastructure, a measurement program is incomplete and will surely fail; worse, they can actually degrade current practices by sapping organizational morale, energy, and resources.

END OF ARTICLE