



Mauri Laitinen and Mohamed E. Fayad

Surviving a Process Performance Crash

Implementing a software development process will probably result in an initial performance reduction for the obvious reason that people will be adapting to new procedures. (Why this perennially shocks everyone is one of the mysteries of organizational life.) But while we expect things to slow down as we adapt to new processes, we must not let process paralysis creep in [4]. If we detect it, we must get the project back on track by emphasizing that our goal is a product rather than a process. Consultants and outside process improvement teams may help mired groups, but outsiders should not take over: removing process ownership hinders adaptation.

Remember that processes, once defined, will undergo incremental evolutionary change, and that changes in the volume of work to be handled, or in the technology underlying the process, may cause fundamental changes in the process. For example, a process for defect tracking on a small project might be wholly inadequate for a large

project. As processes evolve and environments change, the whole process hierarchy should be examined for improvement possibilities. If there is a project process group, it should coordinate with similar groups in other parts of the organization.

When working on process improvement, remember that reactive change will make things worse, not better. In most cases, people are impatient with the time it takes to improve a process, and they identify every variation in process operation as a problem to be fixed. The result is chaos. Worse yet, this also inhibits learning. A cycle of change, measurement, and evaluation must instead be followed. Deming makes this point forcefully [3]. Again, the system must get worse to get better.

Process Metrics

People tend to attribute mystical properties to statistics sent to managers. “We just collect it and let management decide,” is often the attitude. Managers, on the other hand, are compelled to ask for statistical data, even if they

do not use it or methodically review it. They somehow believe they are not doing a proper job—or think they seem uninterested—if they do not ask for and save piles of data. This only accelerates the collection and generation of useless data.

In micro-level processes, statistics should definitely be oriented towards informing the people doing the process so they can control and improve it. Moreover, at the micro-level, some data measurement may come and go as needed. Giving up information will initially feel like a deterioration in organizational control. But if done properly, it actually affects improvement.

Measure Processes Rather Than People

Process orientation has developed a somewhat tarnished reputation because of the way it has been implemented. The goals of process orientation are to improve reliability and efficiency, thereby increasing quality. Too often, organizations try to “install” quality by the use of such techniques as processes.

EXHORTATIONS TO EXCEL AND TO MAKE NO MISTAKES

have little power to actually make people behave
with perfect reliability.

With the increased emphasis on process as technique, problems may arise. Process paralysis and losing sight of the goal of creating products are common traps. Attempting to use off-the-shelf processes, devaluing skill and experience in favor of processes, and putting experts in the position of defining and imposing processes all contribute to the failures that have damaged the reputation of quality management. Having installed processes, some “clever” management types often decide to speed things up by setting goals above the current statistical average for processes. This is destructive. The only way to improve goals is to change the process. Processes are tools that, if used with a sincere organization-wide approach, can help improve work quality and increase productivity.

If management succeeds in creating a process-oriented approach, the next logical step is to instruct management on the use, improvement, and measurement of processes. This is a radical step, and we cannot do full justice to the idea in this short article. We refer the reader to sources that expand upon this [3, 5]. We present the basic idea by reiterating that effective development is a team effort. Just as the misuse of process data will cause people to subvert error reporting, a system of processes that allows one part of the system to succeed

at the expense of another will also be destructive. Processes help organizational systems run effectively, and therefore cannot be viewed in isolation.

When we suggest a different approach to managing people and processes, we do not mean that traditional issues such as absenteeism and personal responsibility should be ignored. Rather, by focusing on the process, fairer evaluations of people can be made. And since coordinating people to reach a goal unreachable by individual action can be considered one of the prime tasks of management, measuring processes is an excellent measure of management itself. Again, these recommendations appear to reduce management’s repertoire, and to be fair, they do. Managers will be nervous, but if they focus on a process, they will trade limited tools for systematic improvement.

Latzko points out with painful clarity how so many attempts at individual performance evaluation are nothing more than the elevation of random chance in personnel management [5]. If we do not have a good idea of how a process varies normally, we cannot accurately tell if people are working well or not. One person’s outstanding performance for a month might only be the result of normal statistical variation. Changes in environment and in volume of work can also affect a process. If a group is set to do work in a cer-

tain way and cannot compensate for the changes, it is counterproductive to discipline people rather than change the process. The only way to tell if a process is working with normal variation is to measure the process with the intent of understanding it.

We also note that traditional improvement strategies are notorious for their lack of lasting results. Trying to improve work by changing attitudes is difficult at best. Often, such efforts become nothing more than exhortations and slogans to which employees properly pay little attention. In contrast, changing the way people work, changing their roles and responsibilities, as well as the method by which they are evaluated, can result in lasting improvement [1].

A Process Evaluation Example

At the highest level, the development process is a system or group of systems. Outputs of one process flow downstream, and if the upstream product causes trouble farther along, the system as a whole suffers. For example, if a programming group sends low-quality output to a testing group, a shallow view of the situation may suggest that the development group looks effective while the testing group appears unable to meet its commitments. In fact, experience shows that the overall time it takes to finish the product will increase. In evaluating

the situation, if we look at people, we might improperly view the testers as the culprits and give them poor performance evaluations. By looking at the system as a whole, we have a better chance of identifying and correcting problems. As Senge points out, "Dividing an elephant in half does not produce two small elephants" [6]. Correcting various groups in isolation will not create a working system.

Endless Change Means

More Control

It is easier to evaluate a process than to evaluate the people using it. However, both process and person are interdependent and processes, in the long run, are created by people. Thus, there will always be the likelihood of something going wrong.

Another reason to focus on processes and systems is the inherent weakness of human judgment. Human judgment is by nature subjective, and people have limited ability to identify changes, especially over time, without some consistent way of measuring. A process gives a relatively repeatable way to see change. Variations over a long period cannot be accurately correlated to events without measurement and recording. When cause and effect are separated for any significant period humans have very limited abilities to understand systems. The common notion of effective management—to react swiftly to change—leads to making constant changes without analyzing a system and without taking account of variation. Deming calls this tampering rather than adjustment. It is destructive [5].

More Processes Means

Greater Creativity

It should be remembered that people are not automatons. Exhortations to excel and to make no mistakes have little power to actually make people behave with perfect reliability. Using the programmer/tester example again, the current state of the industry assures us that programmers will produce programming errors and that testers will miss some of them. The testing staff often have little control over their input: they get the code after it has been produced. If circumstances such as late schedules or inadequate specifications induce the programming staff to create code with more errors than usual, the testing group in most cases will have to take what they get. Software with many errors tends to be more difficult to test adequately, both because there are more errors to find and because the existence of many errors makes subtle problems harder to find or understand. The testing group cannot be expected to compensate fully for the extra errors. Programmers and testers are human: they will not work with perfect efficiency; they cannot compensate entirely for design problems upstream, and psychological factors will cause even greater variables in their work. To expect them to be super-human is not worthwhile.

This column does not suggest that people should behave like machines. Quite the opposite is true. People create processes, people monitor processes, and people analyze and improve processes. People have unique abilities to compensate and adapt that machines do not possess. But just

as people are not expected to turn screws with their fingers but with screwdrivers, it is unreasonable to expect people to do detailed, complex work without tools such as processes.

How do these ideas relate to high-level processes? Such processes are less detailed, and the roles people play are more complicated and diverse. High-level processes concentrate on analysis of systems, not only the system development models, such as the spiral or iterative approaches, but also the way lower-level processes interact.

Looking at the spiral model used by Boehm [2], the order in which prototyping and risk management are done is less important than the way they interact. Understanding the interactions between activities must be the focus. For example, in most iterative processes, each work package ends with a review that guides the actions of the impending work. Having a review and not using the results is wasted effort. This checklist approach to process management will not be very successful. Without a systems approach to high-level processes, the flow of a project is problematic. The most important task in a project is to understand and differentiate between normal variations and uncertainties and those that indicate real problems. The idea is not to watch while the project drifts slowly out of control. Any project of consequence will have variances from the expected. An estimation process, for example, is used to see if initial estimates were correct, and to re-estimate if the estimates were incorrect. But since product is the goal, it is important that

evaluation and reward criteria do not pit different teams or team members against one another.

Summary

We have given a pitch for implementing processes as a way to improve an organization's capabilities and give more management leverage. The difficulty in changing to a process orientation, as with any change, is that things get worse before they get better. From some traditional management perspectives, such as reduced statistics gathering, less-direct intervention, and less emphasis on individual performance, the process approach seems to worsen problems. However, given the potential results, especially when compared to the limited success traditional improvement approaches offer, getting worse with process improvement will make things better. ■

Mohamed E. Fayad (fayad@cs.unr.edu) is an associate professor at the University of Nevada, Reno.

Mauri Laitinen (mdl@sierra.net) is a principal of Laitinen Consulting at Lake Tahoe, Calif.

References

1. Beer, M, Eisenstat, R. A., and Spector, B. Why change programs do not produce change. *Harvard Business Rev.* 68, 6 (Nov.-Dec. 1990).
2. Boehm, B.W. Verifying and validating software requirements and design specifications. *IEEE Softw.* 1, 1 (Jan. 1984), pp. 75-88.
3. Deming, W.E. *Out of the Crisis*. MIT CAES, Cambridge, Mass., 1986.
4. Fayad, M.E., Laitinen, M. *Transition to Object-Oriented Software Development*. Wiley, New York, 1997, to appear.
5. Latzko, W.J. and Saunders, D.M. *Four days with Dr. Deming*. Addison Wesley, Reading, Penn., 1995.
6. Senge, P.M. *The Fifth Discipline*. Doubleday Currency, New York, 1990.