Applying Systems Thinking to Process Improvement

Michael West
Natural Systems Process Improvement

In our day-to-day work in process improvement using the Capability Maturity Model® or the Capability Maturity Model® Integration, it is easy to lose sight of the big picture. Applying systems thinking can generate breakthrough approaches to effectively improving systems development, integration, and maintenance.

The concept of *systems thinking*, which was definitively described by Peter Senge in his seminal work "The Fifth Discipline" [1], has been used by many people to investigate and resolve deep organizational problems and to achieve higher states of operational excellence. You can use systems thinking to effectively resolve many of the barriers and problems that commonly plague process improvement initiatives based on the Capability Maturity Model® (CMM®) or the CMM IntegrationSM (CMMI®).

Losing Sight of the Forest

Working day to day in a process improvement role, it is very easy to lose sight of the *big picture* quickly. We tend to see individuals around us making independent decisions and taking seemingly unrelated actions. We get caught up in trying to deal with every separate event using a different approach and frame of reference than the last seemingly dissimilar event. It is also quite easy to lose sight of the relationships between your process improvement work and system or service delivery. Sometimes, process improvement – or worse, the model we are using, i.e., the CMM or CMMI – takes on a life of its own, and we end up doing process improvement for its own sake.

In systems thinking, Senge described two systems archetypes that provide a way of gaining a big-picture view of commonly occurring systemic problems in organizations: *fixes that backfire* and *shifting the burden*. These two archetypes are particularly useful in understanding and resolving problems that frequently plague CMM- and CMMI-based process improvement efforts.

Fixes That Backfire

In the fixes-that-backfire systems archetype, the obvious solutions are applied to problems. However, because the perceived or obvious solution is frequently applied hastily and without a thorough understanding of the problem, the result is often unintended consequences, including a worsening of the problem. One of the most pronounced examples of a fix that backfires is corporate downsizing to improve profits. In one 1991 study of 850 companies that had cut staff drastically, only 41 percent had achieved the savings they hoped for [2].

The diagram in Figure 1, known as a *causal loop diagram*, illustrates the dynamics of the fixes-that-backfire archetype as it relates to software and systems process improvement. Because the net, long-term, negative effects of the *fix* are greater than the short-term, positive effects, the reinforcing loop is the prevailing influence in the system.

> "One of the most pronounced examples of a fix that backfires is corporate downsizing to improve profits. In one 1991 study of 850 companies that had cut staff drastically, only 41 percent had achieved the savings they hoped for."

In the figure, the inner loop (or the core loop) represents the organization attempting to address the problem of poor software or systems delivery by implementing model-based process improvement. The outer loop – also known in systems thinking as an *addiction loop* – represents the long-term effects of the *fixes*. When it turns out that the CMMI is not the panacea for all the organization's problems, there can be unintended negative consequences.

The fixes-that-backfire archetype has three primary manifestations in CMM and CMMI based process improvement:

1. The race to achieve a maturity level (the perceived fix) causes widespread cynicism, which in turn leads to a grass-roots resistance – a backlash – to the process improvement initiative (the unintended consequence).
2. Process discipline and improvement (the perceived fix) is done for its own sake, and the consequence is that the organization's software/systems development performance gets worse (the unintended consequence).
3. The unintended consequences of business and model disintegration occur when the cost and value of process improvement activities are not integrated with the price of the things such as products or services that are sold to a customer.

These three fixes that backfire in model-based process improvement are described in the following subsections.

The Race to Maturity Levels

Frequently, executive and senior level managers are sold on the idea that the CMM or CMMI are *the* vehicles for improving software or systems development and delivery. However, they get fixated on achieving maturity levels under the belief that maturity levels are concrete evidence that processes have improved. The maturity level becomes the evidence that the organization has improved its efficiency, effectiveness, and quality. Such beliefs are as faulty as assuming a student has actually learned something because he received an *A* in class.

Acting on such beliefs, executives will sometimes construct incentives such as bonus programs for senior and mid-level managers to achieve maturity levels in their respective sub-organizations. Now, both level-envy, and the ensuing race to achieve levels, is on! The focus on maturity levels drives people to look for quick fixes that will enable them to *pass* an assessment. The symptoms – observable behaviors and artifacts – most commonly associated with a race to maturity levels are the following:

- Deadlines are established for achieving a maturity level without any estimating or project planning that supports the deadline (which, by definition, is a low maturity behavior).
- Slogans appear such as "Level 5 in '05."
- People read the CMM or CMMI, and learn to recite the model's terms and phraseology.
- Policies and procedures are rapidly created, and frequently mimic the practices in the CMM or CMMI instead of defining the organization's processes.
- Project managers, under pressure from upper level management, create elaborate project documentation that satisfies the letter but not the intent of the model. However, they promptly shelve the project documentation and do not use it to manage their projects.

The problem worsens in a large organization in which various sub-organizations are trying to achieve maturity levels independent of each other. The sub-organizations' respective managers too often let their egos and competitive natures get the better of them, and try to outdo each other to be the first to achieve the targeted level.

Once organizations have become entrenched in the level race (or perhaps more accurately, *level wars*), they are in the addiction loop, and you can count on rationale and reason often being abandoned. The organization takes a short trajectory to the maturity level, which it more often than not achieves – one way or another.

What is the result of the unintended consequences? Those incentives for achieving maturity levels usually stop at the mid-level manager, and almost never make it down to anyone doing the work, including software engineering process group members and project managers who have done the bulk of the work in the death-march process improvement project. For a brief time following *passing* the assessment, everyone in the victorious organization is exuberant. But once the assessment *high* wears off, people start looking around for some lasting and meaningful results of their work. Yet for reasons they cannot always comprehend, everything looks and feels the same as it did before the maturity level race.

Even in extreme command-and-control work environments where the primary motivation is fear (usually of losing your job or being marginalized), fixes that backfire eventually take their toll on the morale and momentum of process improvement initiatives. Smart, skilled, process-oriented people start to look outside the organization for more meaningful, rewarding work. Project managers and engineers become jaded on the whole idea of model-based process improvement. They will continue giving a gratuitous *salute* or lip service for fear of not doing so, but they will be burned out on process. Worse yet, they may perceive – perhaps accurately – that their organization's CMM/CMMI effort is a waste of time and money.

The organization that chases maturity levels for their own sake, and fails to set business goals for process improvement, will spend hundreds of thousands or even millions of dollars on model-based process improvement, and can end up with nothing more to show for their efforts than a few gratified egos.

Process for Its Own Sake

Sometimes, organizations do not get too wrapped up in CMM/CMMI maturity levels, yet process still becomes the be-all and end-all to fixing every issue plaguing the organization. For many of my years in Xerox, *it's the process*, or *fix the process* became the only politically acceptable approach to any problem. The primary fallacy of this approach is that it ignores the observable, measurable fact that there really are people and accountability problems, or technology problems that cannot be resolved by addressing only the process.

A classic example of a backfire from applying a fix occurs when organizations attempt to apply the entire CMMI to traditional information technology, systems maintenance, or engineering services shops. With intelligent interpretation and tailoring, many of the CMMI practices can be applied to improve work in these environments. But the implementation of processes and procedures that are nothing but a regurgitation of the CMMI in these environments results in burdening the organization with process overhead that does not add value, thus making the organization less effective and less efficient than they were without the CMMI.

Unintended Consequences of Business and Model Disintegration

Businesses incur unintended and negative consequences when process improvement (and probably other internal *improvement* or quality initiatives, a.k.a., *solutions*) are treated as infrastructure or overhead. As these internal initiatives grow, so does the percentage of the organization's employees whose work does not directly produce something that is sold to or adds value to what is sold to a customer (the consequence). Thus, as William Bergquist noted in his book "The Post Modern Organization" [3], it becomes increasingly difficult for the expanding organization to achieve and maintain its profit goals because operating and overhead costs grow at a faster rate than that of realistically achievable revenue.

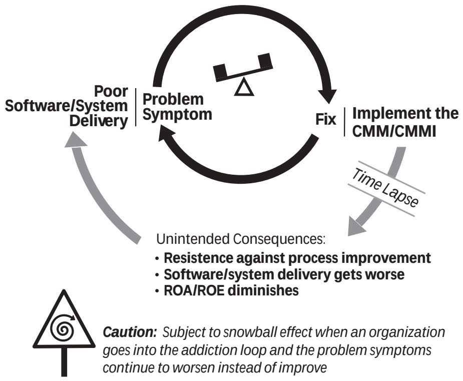
Poor
Software/System
Delivery
Problem
Symptom
Fix
Implement the
CMM/CMMI
Time Lapse
Unintended Consequences:
• Resistence against process improvement
• Software/system delivery gets worse
• ROA/ROE diminishes
Caution: Subject to snowball effect when an organization goes into the addiction loop and the problem symptoms continue to worsen instead of improve

Figure 1: *Fixes-That-Backfire Archetype in Process Improvement*

The approach some organizations take to keep the cost of process improvement down is to hide it. People in charge of process improvement or CMMI efforts in large organizations have actually presented at conferences in which they proudly announced that people worked unpaid overtime on nights and weekends to achieve a maturity level. While this may seem heroic and laudable on the surface, even a layperson can easily see that the maturity level was not truly deserved because of the obvious Level 1 behaviors exhibited in Organization Process Focus.

Strategies for Fixes That Backfire

Here are some strategies you can employ to prevent or mitigate the effects of implementing fixes that backfire:

- In planning the process improvement effort, ensure the plans include achieving measurable or observable business goals in addition to achieving maturity levels. Make sure that reporting progress or success includes status against all the process improvement goals and not just the number or percentage of practices satisfied.
- Increase awareness, especially among senior and executive managers, of the unintended consequences of chasing maturity levels. In 2001, I was involved

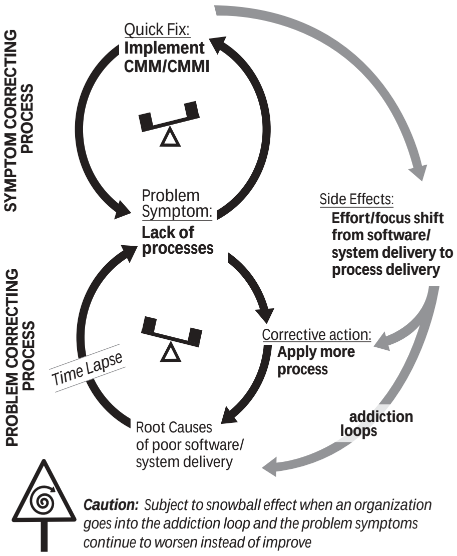
SYMPTOM CORRECTING PROCESS
PROBLEM CORRECTING PROCESS
Quick Fix: Implement CMM/CMMI
Problem Symptom: Lack of processes
Side Effects: Effort/focus shift from software/ system delivery to process delivery
Corrective action: Apply more process
Time Lapse
Root Causes of poor software/ system delivery
addiction loops
Caution: Subject to snowball effect when an organization goes into the addiction loop and the problem symptoms continue to worsen instead of improve

Figure 2: *Shifting the Burden Archetype in Process Improvement*

in an email conversation with senior level managers and marketing people in Computer Sciences Corporation who were trying to come up with strategies for countering the *CMM level race* approach of one of our biggest competitors. I sent out a note thinking that it would be career limiting because it went directly against the prevailing beliefs. Much to my surprise, a senior marketer read and understood my note, and invited me into further conversations on how we could market real process improvement benefits without getting into level wars with our competition.

- Spend the time understanding the problem. You do not necessarily have to conduct time-consuming formal root-cause analysis, which can often lead to *analysis paralysis*, but you cannot afford to continue applying solutions to symptoms, only to have the root problem perpetuate or worsen.
- Establish alliances or relationships with people in your marketing and sales organizations. Find ways to defray some of the cost of process improvement activities by selling (and charging for and collecting) the benefits of process improvement. With the right approach, a customer or client will be willing to pay a higher price for the goods or services so long as you have convinced them that there is greater value.

Table 1: *Common Problems Attributed to Lack of Process That May Have Other Causes*

Software/Systems Delivery Problems	Perceived as a Process Problem	Potential Real or Root Problems Not Addressed due to Shift
Projects experience scope-creep.	• Inadequate requirements management process.	• Poor business relationship with customer. • Lack of discipline in engineering staff. • There is no release strategy. • No strategy/process for insertion of new technology. • Lack of standards for acceptance or decline of requirements. • The Organization does not understand the market it is in.
Projects overrun cost and schedule.	• No estimating or planning process. • Project plans are not adequately documented.	• Culture encourages *low bid*; accurate bidders don't get work. • *Schedule* is synonymous with *plan.* • Poor business relationship with customer. • There is no release strategy. • Staff has inadequate skills to do the work. • The term *Project* is not defined. • Management does not perceive planning activities as work/progress.
Product quality (e.g., defect density) is poor.	• No quality process. • No people assigned to inspect/audit the quality.	• The organization has no defined standards or criteria that define *quality*. • Management and the culture rewards fast and cheap; good is not encouraged or rewarded. • Staff has inadequate skills and resources to produce quality work.
No amount of process improvement activity seems to improve the *bottom line.*	• People will not buy into the PI initiative. • Not enough in-house CMM/CMMI expertise. • Clients do not value the process improvement efforts.	• Misalignment between the chosen model and the organization's core business and business goals. • There are no baseline performance or capability measurements with which improvement could be ascertained. Improvement is anectdotal.

Shifting the Burden

According to Senge's "Systems Thinking,"

> Shifting the burden … usually begins with a problem (symptom) that prompts someone to intervene and solve it. The solution (or solutions) are obvious and immediate; they relieve the problem symptom quickly. But they divert attention away from the real or fundamental source of the problem, which becomes worse as less attention is paid to it. This forces the perception that there is no other way out except the symptomatic solution. [1]

Shifting the burden to process improvement is illustrated in Figure 2.

In this systems archetype, the perceived problem is a lack of process or process discipline in the organization, or that the organization does not have a CMM or CMMI maturity level. So the obvious, or solution, is to implement the CMM or CMMI. In the short term, the fix does appear to address the perceived problem: lack of process is replaced with process.

The unobserved yet insidious effect is that effort and focus shifts from product or service delivery (or integration) to the symptom of inadequate processes that in turn, either does nothing to improve product/service delivery, or hurts it by burdening the existing delivery processes with overly bureaucratic standards and procedures.

Table 1 identifies some common systems development and delivery problems (in the left column). The center column identifies how these problems often are perceived as process (or lack of process) problems. The third column identifies other possible root causes of the problem, which may have little to do with process discipline or maturity levels.

What are the unintended consequences of shifting the burden to process? In many cases, doing so can have a compounding, double negative impact on the organization. With resources diverted from the real problems, the real problems get obscured, are given a lower priority, or are ignored, which diminishes the chance of them being resolved. Worse yet, the process improvement efforts – which can be quite expensive – may not only have no discernable effect on the symptoms, they may exacerbate the root cause.

Take the situation in which the outsourced contractor has a poor relationship with the customer or prime contractor. Shifting the burden to process by throwing the CMMI, ISO, Six Sigma, Theory of Constraints, or some other currently popular initiative at the perceived problem can irritate the customer even more, further worsening the relationship.

Strategies for Dealing With Shifting the Burden to Process

If you are fortunate enough to get involved in a process improvement initiative at its inception, make every effort to get everyone involved in discussing the business problems they want to address, and how the perceived process solutions will address those problems. As with fixes that backfire, spend time first understanding the problem to be resolved or the business goals to be achieved. If the correlation (or better yet, the causation) between business problems and process solutions cannot be clearly established, encourage people to consider pursuing alternative solutions such as organization-wide skill improvement, new technology, or fundamental shifts in business strategy.

If you get involved in a process improvement initiative after it is already underway, be persistent in questioning people around you about their proposed process solutions. Constantly ask questions such as, "Why are you doing this?" "What problem will this solve?" You may become quite annoying to some people, but after a while, you will have them asking the same questions at least quietly to themselves if not overtly.

Again, do not presume that your organization is the first ever to try to apply process solutions to its problems or goals. Conduct benchmarking activities with other organizations to find out what has worked and what has not. Process people are just as susceptible to the pitfalls of the *not-invented-here* syndrome as are engineers and other technical people.

Why Systems Thinking?

Modern software or systems organizations are themselves a system of systems. There are people (social systems), tools and technology (environmental systems), and policies and processes (process systems). The three systems – people, tools, and processes – are inextricably interwoven, and changing one without considering the interrelationships can cause fixes that backfire, do not resolve the original problems, or inadvertently make the problems worse.

The greatest unintended consequence of applying a CMMI or process solution to a non-process problem is too often the vast waste of resources used for the *faux fix*. If you really want to improve things in your organization, start by improving the process of process improvement. You can save your organization money and aggravation by using this systemic approach.◆

References

1. Senge, et al. The Fifth Discipline Fieldbook: Strategies and Tools for Building a Learning Organization. Currency-Doubleday, 1994.
2. Boroughs, Don. L. "Amputating Assets: Companies That Slash Jobs Often End Up with More Problems Than Profits." US News and World Report 4 May 1992.
3. Bergquist, William,. The Post Modern Organization: Mastering the Art of Irreversible Change. Jossey-Bass, Inc., 1993.

About the Author

Michael West is co-founder of the consulting firm Natural Systems Process Improvement (Natural SPI). He has more than 22 years in software and systems engineering management and model-based process improvement. Natural SPI has employed non-traditional, highly effective approaches to helping clients use the Capability Maturity Model® or the Capability Maturity Model® Integration (CMMI®) to achieve measurable business results. This article is excerpted from West's book "Real Process Improvement Using the CMMI."

Natural Systems
Process Improvement
Phone: (866) 648-5508
E-mail: michael@naturalspi.com