Nobody Ever Gets Credit for Fixing Problems that Never Happened:

Creating and Sustaining Process Improvement

Nelson P. Repenning
John D. Sterman

How much would your organization pay to develop manufacturing capability equal to Toyota's? How much would a world-class, six-sigma quality program be worth to your company? How about Harley-Davidson's ability to tap into the hearts and minds of its customers or Dell's ability to manage its supply chain? Most firms are working aggressively to develop these and similar capabilities through process improvement. The combined expenditure of U.S. companies on management consultants and training in 1997 was over $100 billion, and a sizeable fraction went towards efforts to develop operational capabilities matching those of the best firms in business. Whether it's an advanced manufacturing system or the ability to respond quickly to changing customer needs, the drive toward improvement has become a way of life in corporations today. There is only one problem. Despite these vast expenditures, and notwithstanding dramatic successes in a few companies, few efforts to implement such programs actually produce significant results.

Consider, for example, Total Quality Management (TQM). In the 1980s, spurred by the success of many Japanese firms, TQM was all the rage among U.S. firms. Consultants and business school faculty preached its virtues and managers made pilgrimages to companies with award-winning quality programs. By the mid-1990s, however, TQM was considered passé. Academics had moved on to other issues, TQM received rare mention in the popular business press, and articles that did mention it usually did so in a negative context. TQM

Work reported here was supported by the MIT Center for Innovation in Product Development under NSF Cooperative Agreement Number EEC-9529140. For more information on the research program that generated this article, visit <http://web.mit.edu/nelsonr/www/>.

had all the earmarks of a management fad: An initial burst of enthusiasm, a flurry of activity, and then a steady decline as it was replaced by newer innovations such as re-engineering. It would be easy to conclude that TQM's underlying value was minimal.

However, when one looks at the experience a little more carefully, a different picture emerges. A number of careful studies have now demonstrated that companies making a serious commitment to the disciplines and methods associated with TQM outperform their competitors.[1] There is now little doubt that when used properly, TQM produces significant value to both organizations and their customers. Yet paradoxically, it remains little used. A recent study found that fewer than 10% of the *Fortune 1000* had well-developed TQM programs; and, in another study, TQM fell from the third most commonly used business tool in 1993 to 14th in 1999.[2] The situation is similar for a wide range of other administrative and technological innovations.[3] Techniques touted as today's "core competencies" all too often become tomorrow's failed programs. Once an effort has failed, there is an almost irresistible temptation to label it a fad or "flavor of the month." However, digging a little deeper shows that many such techniques have useful content. It should come as little surprise then that many currently popular innovations are little more than old ideas with new acronyms. The core disciplines associated with statistical process control and variance reduction become six-sigma; what was once called a quality circle is now a high-performance work team.

Thus, today's managers face a paradox. On the one hand, the number of tools and techniques available to improve performance is growing rapidly. Further, with advances in information technology and the ever-growing legions of management consultants, it is easier than ever to learn about these techniques and to learn who else is using them. On the other hand, there has been little improvement in the ability of organizations to incorporate these innovations in their everyday activities. The ability to *identify* and *learn about* new improvement methods no longer presents a significant barrier to most managers. Instead, successfully *implementing* these innovations presents the biggest challenge. Put more simply, you can't *buy* a turnkey six-sigma quality program. It must be developed from within.

To learn how firms can overcome this "improvement paradox," we have, over the past decade, studied process improvement and learning programs, focusing on the dynamics of implementation and organizational change. We conducted over a dozen in-depth case studies in industries including telecommunications, semiconductors, chemicals, oil, automobiles, and recreational products.[4] We gathered data through observations, extensive interviews with participants, archival records, and quantitative metrics. We complemented our field research with the development of a series of models capturing the dynamics of implementation and improvement.[5] Using system dynamics as the basis for understanding implementation has yielded a number of insights into the improvement paradox. In at least some cases, these insights have proven instrumental in helping firms benefit from the potential provided by available improvement tools and techniques.

Most importantly, our research suggests that the inability of most organizations to reap the full benefit of these innovations has little to do with the specific improvement tool they select. Instead, the problem has its roots in how the introduction of a new improvement program interacts with the physical, economic, social, and psychological structures in which implementation takes place. In other words, it's not just a tool problem, any more than it's a human resources problem or a leadership problem. Instead it is a systemic problem, one that is created by the interaction of tools, equipment, workers, and managers.

The Structure of Improvement

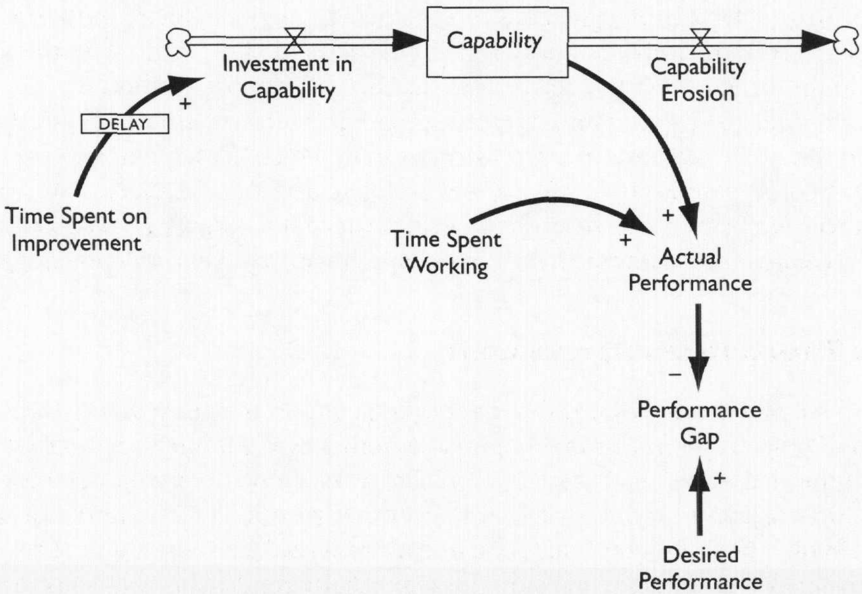
We present the lessons that have emerged from our study in the form of a causal loop diagram. Our model provides both a useful framework for thinking about the challenges associated with implementing improvement programs and practical suggestions to increase the chances that your next such effort will succeed. While the theory reported here initially emerged from the study of two improvement initiatives in a major automaker,[6] the resulting model is quite general and can be applied to a range of situations. We have observed these dynamics in almost every organization we have studied.

Figure 1 begins with the basic "physics" underlying process improvement. The actual performance of any process depends on two factors: the amount of *Time Spent Working* and the *Capability* of the process used to do that work. For example, in manufacturing, net usable output is given by the product of labor hours per day and productivity (usable units per labor hour).

The performance of any process can be increased by dedicating additional effort to either work or improvement. However, the two activities do not produce equivalent results. Time spent on improving the capability of a process typically yields the more enduring change. For example, boosting the workweek 20% might increase output 20%, but only for the duration of the overtime. Gains in process capability, however, boost the output generated by every subsequent hour of effort. Similarly, overtime devoted to reworking defective products can boost net usable output, but only as long as the overtime is continued, while eliminating the root causes of those defects permanently reduces the need for rework. We capture this persistence by representing *Capability* as a stock (denoted by a rectangle), that is, as an asset that accumulates improvements over time. Specifically, *Time Spent on Improvement* increases the flow of *Investments in Capability* that augments process capability.

While it often yields the more permanent gain, time spent on improvement does not immediately improve performance. It takes time to uncover the root causes of process problems and then to discover, test, and implement solutions, shown in the diagram as a delay between improvement activities and the

FIGURE 1. The "Physics" of Improvement

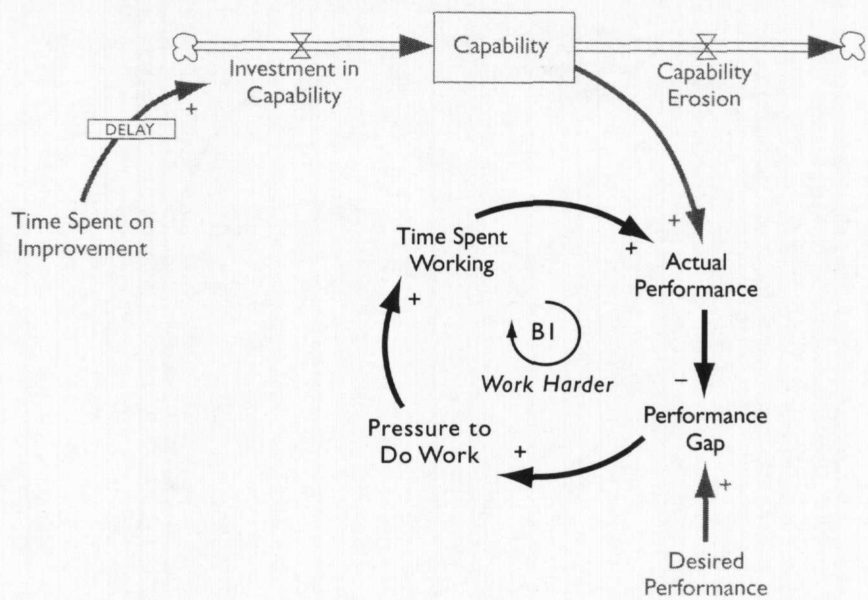


Note: Arrows indicate the direction of causality. Signs ('+' or '–') at arrowheads indicate the polarity of relationships: a '+' means that an increase in the independent variable causes the dependent variable to increase, all else being equal (a decrease causes a decrease); similarly, a '–' indicates that an increase in the independent variable causes the dependent variable to decrease (a decrease causes an increase). For more details, see J. Sterman, *Business Dynamics: Systems Thinking and Modeling for a Complex World* (New York, NY: Irwin/McGraw-Hill, 2000).

resulting change in process capability. Moreover, no improvement in capability lasts forever. Machines wear, processes go out of control without regular attention, designs become obsolete, and procedures become outdated. Thus, we also show an outflow from the stock capturing the inevitable decline of any capability that is not regularly maintained. The lag in enhancing capability depends on the technical and organizational complexity of the process. Studies show that the delay in improving relatively simple processes such as the yield of machines in a job shop is on the order of a few months, while the delay in improving highly complex processes such as product development can be several years or more.[7] Similarly, the lifetime of improvements in capability will be shorter in organizations with high rates of change in products and people.

Besides the physical and institutional structures that determine performance, Figure 1 also shows the goal for process throughput set by senior managers (labeled *Desired Performance)*. The goal could be the number of products demanded by customers each day, the rate at which claims need to be processed by an insurance company, or the number of new products the firm seeks to launch this quarter. People compare that goal to their actual performance to determine the *Performance Gap*. Not surprisingly, in the organizations we studied

FIGURE 2. The *Work Harder* Balancing Loop



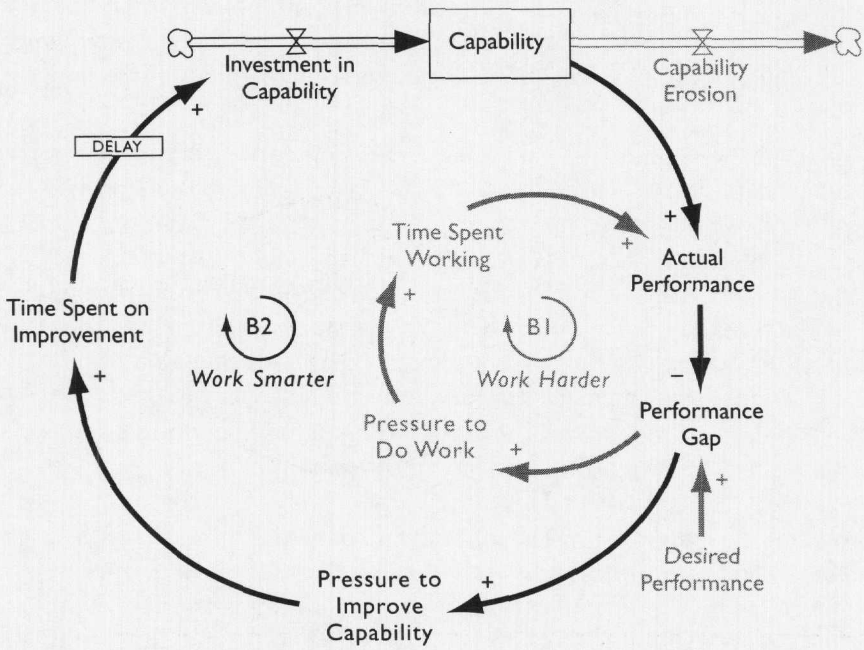
Note: The loop identifier, B1, indicates a negative (balancing) feedback. See J. Sterman, *op. cit.*

it was rare to find a process performing above expectations. Instead, managers, workers, and engineers usually faced high and rising demands, sometimes despite downsizing and cuts in resources. They were constantly searching for ways to improve and close the performance gap. Since most organizations are reluctant to increase plant and equipment or hire more staff, managers hoping to close a performance gap have only two basic options.

First, they can try to increase the amount of time people actually spend working. Figure 2 shows this option, which forms a *balancing* feedback, the *Work Harder* loop B1. The process represented by this loop works as follows: Managers facing a performance gap are under pressure to increase performance. They pressure people to spend more time and energy doing work. An increase in the time spent working increases the performance of the process and closes the performance gap. This structure is called a balancing feedback loop because it constantly works to balance desired and actual performance.

Pressure to do Work includes, most obviously, direct measures such as telling people to work faster or put in overtime, setting more aggressive targets for throughput, and imposing more severe penalties for missing those targets. Pressure also includes more subtle actions designed to extract greater effort from employees. These include the frequency with which performance is reviewed, the detail with which the reviews are conducted, and the seniority of those doing the reviewing. At one company we studied, it was not unusual for senior

FIGURE 3. The *Work Smarter* Balancing Loop



vice-presidents to review the performance of individual machines on the factory floor. Not surprisingly, such attention sent a strong message to all involved: keep the machines busy at all costs. Similarly, a project manager we interviewed recalled that when a subsystem for which he was responsible fell behind schedule, his boss required him to call in every *hour* with a status report until the prototype met its specifications.

A second option to close a performance gap is to improve the capability of the process. In Figure 3 we represent this option as another balancing feedback process, the *Work Smarter* loop B2. Here, managers respond to a performance shortfall by increasing the pressure on people to improve capability. They may launch improvement programs, encourage people to experiment with new ideas, and invest in training. If successful, these investments will, with time, yield improvements in process capability, boost throughput, and close the performance gap. Of course, everyone knows that it is better to work smarter than to work harder: An hour spent working produces an extra hour's worth of output, while an hour spent on improvement may improve the productivity of every subsequent hour dedicated to production. Yet, despite its obvious and documented benefits, working smarter does have limitations. First, as shown in the diagram, there is often a substantial delay between investing in improvement activities and reaping the benefits. Further, the greater the complexity of the process, the longer it takes to improve.[8] Second, investments in capability can be risky. Improvement efforts don't always find the root cause of defects, new tools sometimes don't produce the desired gains, and experiments often fail. While investments in capability might eventually yield large and enduring improvements in productivity, they do little to solve the problems managers face *right now.*

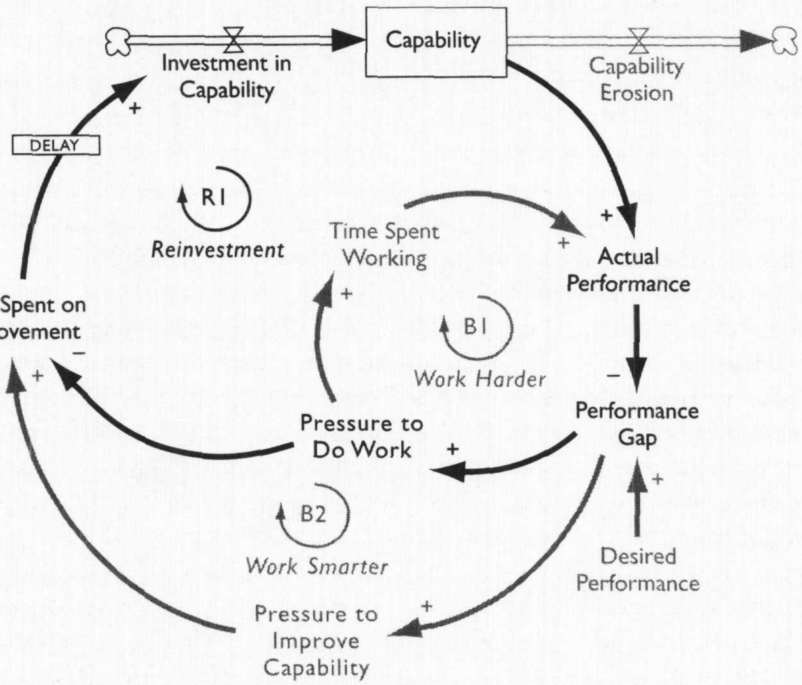
Thus, it is not surprising that managers frequently use the *Work Harder* loop to both accommodate variations in daily workload and solve pressing problems created by unexpected breakdowns or defects. When a manufacturing line serving an important customer goes down, a manager is unlikely to react by sending the work team to training in reliability improvement. Instead, that manager is going to get the line running and push for overtime until the shipment is out the door. Of course when the line is back running and the product has been shipped, the manager should return attention to the improvement activities that will prevent future breakdowns, and make up for the improvement time that was lost during the crunch. However, it doesn't usually happen. Instead, what we repeatedly observe, and what is more difficult to understand, are organizations in which working harder is not merely a means to deal with isolated incidents, but is instead standard operating procedure. Rather than using the work harder loop to occasionally offset daily variations in workload, managers, supervisors, and workers all come to rely constantly on working harder to hit their targets and, consequently, never find the time to invest in improvement activities. What starts as a temporary emphasis on working harder quickly becomes routine.

The Reinvestment Loop

To understand why, it is helpful to consider how working smarter and working harder are connected. The most important interconnection arises because organizations rarely have excess resources. Increasing the pressure to do work leads people to spend less time on non-work related activities like breaks and to put in overtime (that is, they use the *Work Harder* loop). For knowledge workers such overtime is often unpaid and spills into nights and weekends, stealing time from family and community activities. There are, however, obvious limits to long hours. After a while there is simply no more time. If the performance gap continues to rise, workers have no choice but to reduce the time they spend on improvement as they strive to meet their ever-increasing objectives. Figure 4 adds the connection between pressure to do work and the amount of time spent on improvement.

The additional link creates the *Reinvestment* loop. Unlike those described so far, the *Reinvestment* loop is a positive feedback that tends to reinforce whichever behavior currently dominates. An organization that successfully improves its process capability will experience rising performance. As the performance gap falls, workers have even more time to devote to improvement, creating a virtuous cycle of improved capability and increasing attention to improvement.

FIGURE 4. The *Reinvestment* Reinforcing Loop



Note: The loop identifier, R1, indicates a positive (reinforcing) feedback. See J. Sterman, *op. cit.*

Conversely, if managers respond to a throughput gap by increasing work pressure, employees increase the amount of time spent working and cut the time spent on improvement. Capability begins to decay. As capability erodes, the performance gap grows still more, forcing a further shift towards working harder and away from improvement. Here the reinvestment loop operates as a vicious cycle, driving the organization to ever-higher degrees of work pressure and minimal levels of process capability. Not surprisingly, such a vicious cycle quickly drives out meaningful improvement activity. Here, for example, is the way a manager in an electronics assembly plant explained the persistent failure of the organization to engage in process improvement:

> "Supervisors never had time to make improvements or do preventative maintenance on their lines . . . they had to spend all their time just trying to keep the line going, but this meant it was always in a state of flux, which, in turn, caused them to want to hold lots of protective inventory, because everything was so unpredictable. A quality problem might not be discovered until we had produced a pile of defective parts. This of course meant we didn't have time to figure out why the problem happened in the first place, since we were now really behind our production schedule. It was a kind of snowball effect that just kept getting worse."

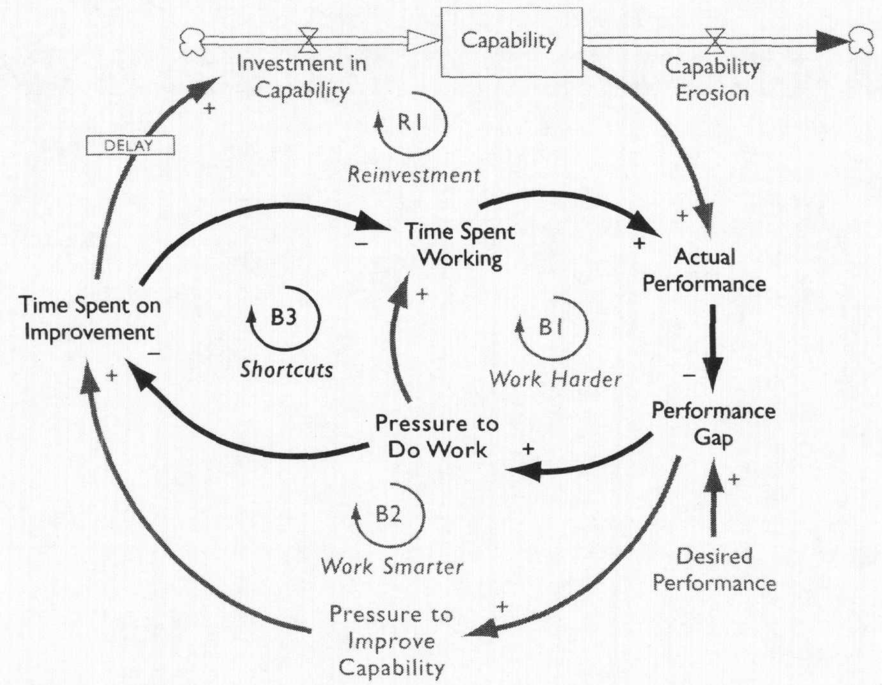
Shortcuts and the Capability Trap

The *Reinvestment* loop means a temporary emphasis on one option at the expense of the other is likely to be reinforced and eventually become permanent. Organizations that invest in improvement will experience increasing capability and find that they have more time to allocate to working smarter and less need for heroic efforts to solve problems by working harder. In the successful initiatives we studied, leadership often worked to strengthen the reinvestment process by explicitly allocating the resources freed up by productivity gains to further improvement. Unfortunately, however, these initiatives were the exception rather than the rule. In most of the organizations in our study the reinvestment loop worked as a vicious cycle and prevented improvement programs from getting off the ground. Even when improvement programs yielded initial results, cost and schedule pressures soon tempted many organizations into downsizing or higher performance goals that drained resources away from improvement, weakening the reinvestment loop and causing capability to stall or even fall.[9]

Understanding why the reinvestment loop typically worked in the downward, vicious direction rather than the upward, virtuous direction requires that we add a final link to the model (see Figure 5). As discussed above, cutting investments in maintenance and improvement in favor of working harder erodes process capability and hurts performance. However, capability does not drop right away. It takes time for process integrity to depreciate. In the meantime, the decision to skimp on improvement—skipping improvement team meetings, neglecting to take machines down for scheduled maintenance, or ignoring documentation requirements—boosts the time available to get work done right now. We capture this interconnection by adding a negative link between *Time Spent on Improvement* and *Time Spent Working*. When the performance gap rises and managers resort to increased work pressure, overworked people cut back improvement activity to free still more time for production. The performance gap falls, closing a third feedback that works to balance desired and actual performance. We label this the *Shortcuts* loop (B3) to capture the idea that increased throughput comes at the cost of departing from standard routines and processes, cutting corners, and reducing the time spent on learning and improvement.

Shortcuts are tempting because there is often a substantial delay between cutting corners and the consequent decline in capability. For example, supervisors who defer preventive maintenance often experience a "grace period" in which they reap the benefits of increased output (by avoiding scheduled downtime) and save on maintenance costs. Only later, as equipment ages and wears do they begin to experience lower yields and lower uptimes (see section 4). Similarly, a software engineer who forgoes documentation in favor of completing a project on time incurs few immediate costs; only later, when she returns to fix bugs discovered in testing does she feel the full impact of a decision made weeks or months earlier.[10] Thus, the *Shortcuts* loop is effective in closing the throughput

FIGURE 5. The *Shortcuts* Balancing Loop

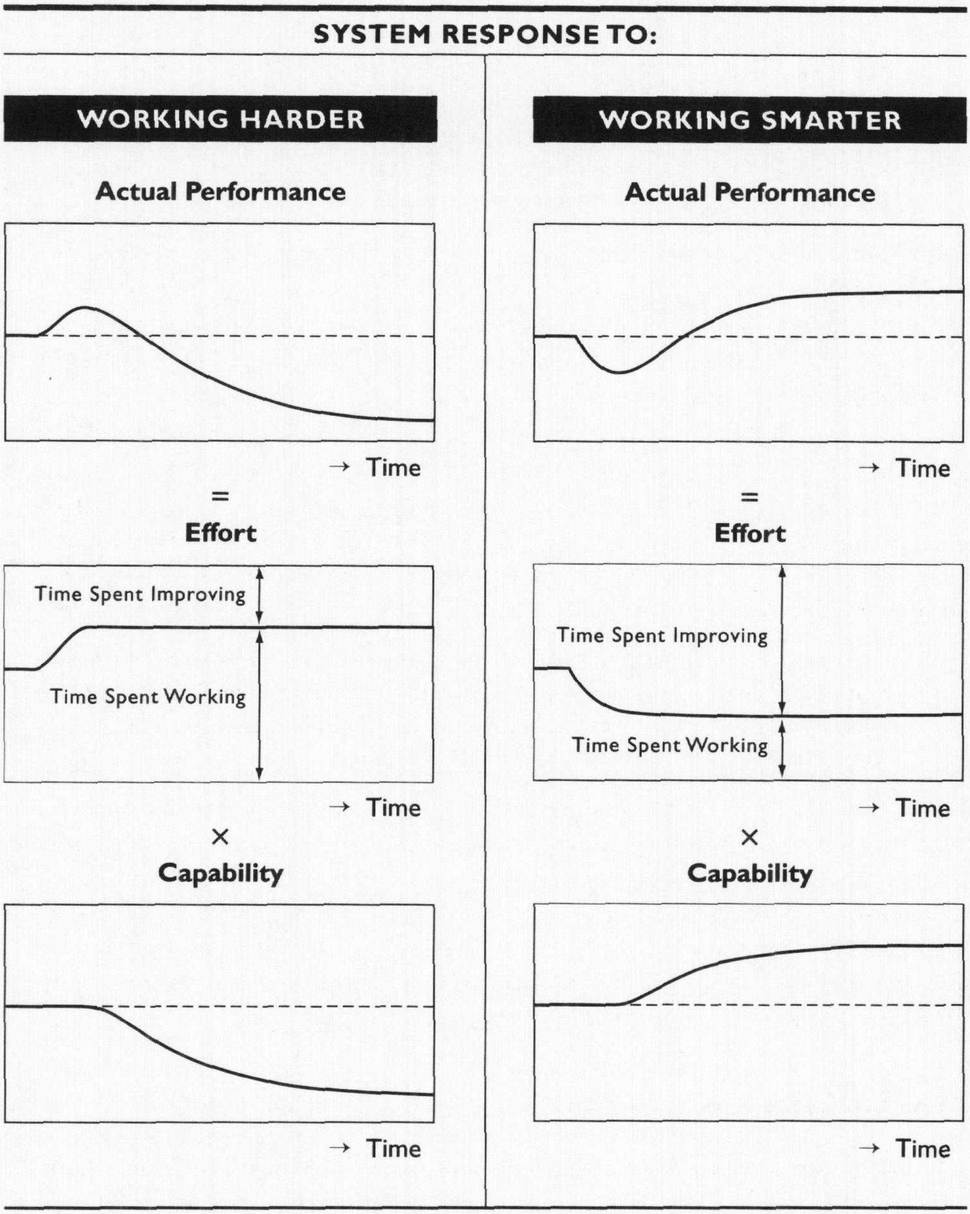


gap only because capability does not change immediately when the time dedicated to learning and improvement declines.

To illustrate these dynamics, Figure 6 shows two simulations of the model in which we show how a hypothetical process reacts to working harder versus working smarter. Both simulations begin in the same equilibrium state. The first simulation shows the response to an increased emphasis on working harder. As more effort is dedicated to work, gross throughput immediately rises. Time spent improving falls immediately, but capability does not. Performance therefore rises. The benefit of working harder is, however, short-lived. With less time devoted to improvement, capability gradually erodes, eventually more than offsetting the increased time spent working. Working harder creates a "better-before-worse" situation. Conversely, as seen in the second simulation, increasing the time spent on improvement reduces output in the short run. Eventually, however, capability rises more than enough to offset the drop in work effort and performance is permanently higher, a "worse-before-better" dynamic.

The interaction between the balancing *Shortcuts* loop and the reinforcing reinvestment loop creates a phenomenon we call the *Capability Trap* and helps explain why organizations often find themselves stuck in a vicious cycle of declining capability. Managers and workers in need of an immediate performance

FIGURE 6. Simulations of the Working Harder and Working Smarter Strategies



boost can get it by skimping on improvement and maintenance. However, capability eventually declines, causing the *Reinvestment* loop to work as a vicious cycle. Managers who rely on working harder and shortcuts to meet immediate throughput needs soon find the process falling short of its objectives, requiring a further shift towards working harder and away from improvement. To see the capability trap in action, consider how a manufacturing supervisor in an auto company explained the inability of her organization to make a commitment to regular improvement activities:

> "In the minds of the [operations team leaders] they had to hit their pack counts. This meant if you were having a bad day and your yield had fallen . . . you had to run like crazy to hit your target. You could say, 'you are making 20% garbage, stop the line and fix the problem,' and they would say, 'I can't hit my pack count without running like crazy.' They could never get ahead of the game."

By keeping the line going rather than stopping to fix the problem, these team leaders relied on the *Shortcuts* loop to hit their throughput objectives. However, by "running like crazy" they also caused the *Reinvestment* loop to operate as a vicious cycle, driving the line to a minimal level of capability and forcing them to run ever faster.

The capability trap is not limited to manufacturing—we have observed it in firms ranging from financial services to construction. For example, the capability trap prevented a product development organization we studied from developing new processes that would have increased productivity. Like many firms, they sought to create an engineering library or "bookshelf" of reusable designs and software. However, as described by an engineering manager,

> "An engineer might not take the time to document her steps or put the results of a simulation on the bookshelf and because of that she saved engineering time and did her project more efficiently. But in the long run it prevented us from being able to deploy the reusability concepts that we were looking for."

Just as machine operators and supervisors in the first example faced a basic trade-off between producing and improving, development engineers were forced to trade off getting their assigned tasks done against documenting what they learned so that others might benefit. Engineers could make more rapid progress towards their objectives by taking shortcuts and ignoring the bookshelf, but doing so prevented them from initiating the self-reinforcing reinvestment loop that would have led to improved process capability.

The Persistence of the Capability Trap

Because working harder and taking shortcuts produce more immediate gains and help solve today's problems, managers unaware of the inherent "better before worse" trade-off are likely to choose them over working smarter. Unfortunately, these temporary gains come at the expense of the long-run health of the process. By pressuring people to work harder, managers often unwittingly force their organizations into the capability trap where ever-increasing levels of effort are required to maintain performance. Of course, this phenomenon is not limited to large organizations. Many readers will recognize this dynamic in different aspects of their personal lives. In situations ranging from learning how to use a new software package to committing to a new exercise program, we often fail to do the things that will improve our long-run productivity and well-being due to the short-run stresses of other obligations.

A question naturally arising at this point is: "Wouldn't managers eventually figure this out?" While it is understandable that, on occasion, people get caught in the capability trap, wouldn't they eventually realize the true source of their problems and rebalance their efforts between working harder and working smarter? Unfortunately, the data suggest that overcoming the capability trap is rare. Managers often do not realize how deeply they are trapped in it. Instead, the lessons that people learn when caught in the capability trap often lead to actions that make the situation worse.

Faulty Attributions

Suppose you are a manager faced with inadequate performance. Your operation is not meeting its objectives and you have to do something about it. As we have outlined so far, you have two basic choices: get people to work harder or get them to work smarter. To decide, you have to make a judgment about the cause of the low performance. If you believe the system is underperforming due to low capability, then you should focus on working smarter. If, on the other hand, you think that your workers or engineers are a little lazy, undisciplined, or just shirking, you need to get them to work harder.

How do you decide? Research suggests that people generally assume that cause and effect are closely related in time and space: To explain a puzzling event, we look for another recent, nearby event that might have triggered it. People also tend to assume each event has a single cause, underestimate time delays, and fail to account for feedback processes. How do these causal attributions play out in a work setting? Consider a manager observing a machine operator who is producing an unusually high number of defects. The manager is likely to assume that the worker is at fault: The worker is close in space and time to the production of defects, and other operators have lower defect rates. The true cause, however, may be distant in space and time from the defects it creates. Perhaps the defect is actually the result of an inadequate maintenance procedure or the poor quality of the training program. In this case, the delay between the true cause and the defective output is long, variable, and often unobservable. As a result, managers are likely to conclude that the cause of low throughput is inadequate worker effort or insufficient discipline, rather than features of the process. The attribution of a problem to the characteristics—and character flaws—of individuals in a system rather than to the system in which they find themselves is so pervasive that psychologists call it the "fundamental attribution error."[11]

Suppose managers conclude that people, not the process, are the source of low performance. Having made such an attribution it makes sense to increase production pressure. As discussed above, an increase in production pressure has two effects. Worker effort immediately rises, closing the performance gap as the manager intended. However, workers are now less able to achieve their objectives by increasing the time they spend working. To continue to hit their ever-increasing targets, they eventually resort to shortcuts, cutting the time spent on improvement. However, as highlighted above, the *Shortcuts* loop, while having the desired effect in the short run, yields a long-run side effect. With less effort dedicated to improvement, capability begins to decline. Performance falls, offsetting the initial gains. By continually increasing throughput objectives in the pursuit of better performance, managers who mistakenly attribute low performance to the attitudes and dispositions of their workforce inadvertently force the system into the capability trap.

Superstitious Learning

The bias towards blaming people rather than the system in which those people are embedded means managers are prone to push their organizations into the capability trap. As workers spend more and more of their time on throughput and cut back on fundamental improvement, shouldn't managers realize that the true cause of sub-standard performance is low process capability rather than unmotivated workers? Unfortunately, in many situations managers learn the opposite lesson.

Managers cannot observe all the activities of the workers. Hence, after they apply production pressure, they cannot easily determine how much of the resulting rise in throughput is due to increased work effort (the *Work Harder* loop) and how much to cutting back on training, improvement or maintenance (the *Shortcuts* loop). For example, suppose there is a performance gap requiring an additional six hours of productive effort per person per week. Managers, believing employees are simply not working hard enough, increase production pressure. Workers buckle down, cutting back on breaks, web-surfing and other nonproductive time. Suppose these responses yield only two hours per person per week. To close the remaining throughput gap, workers resort to shortcuts and gradually reduce the time they spend on process improvement, training, and experimentation until they free the needed four hours per week. Managers observe that throughput rises by the equivalent of six hours of productive effort.

Because managers do not fully observe the reduction in training, experimentation, and improvement effort (they fail to account for the *Shortcuts* loop), they overestimate the impact of their get-tough policy, in our example by as much as a factor of three. The feedback managers receive does not correct the error. To the contrary, managers quickly learn that boosting production pressure *works*—throughput rose when they turned up the pressure. The gains resulting from production pressure provide powerful evidence confirming their suspicions that workers were not giving their full effort.

We call this syndrome the *Self-Confirming Attribution Error*: Once managers decide that the workforce is the source of their difficulties, they take actions that provide convincing and immediate evidence confirming this erroneous attribution. The cycle of self-confirming attributions drives the organization to higher levels of production pressure and fewer resources dedicated to process improvement. Far more importantly, however, it gradually changes the mental models of the managers by providing them with increasingly compelling evidence that the source of low throughput can be found in the poor attitudes and weak character of the workforce. Recall the project manager discussed above who was required to provide hourly status reports on a balky prototype. Soon afterward the problem was solved, confirming the boss's belief that he had acted appropriately, indeed had decisively taken charge of the situation, even though the team was already working around the clock and his interference drained precious time from their efforts to solve the problem.

More subtly, the long-run effects of production pressure also reinforce managers' belief that workers are the problem. The delay between increased production pressure and increased throughput (via the *Work Harder* and *Shortcuts* loops) is short, and the connection between work effort and output is unambiguous. In contrast, the erosion of process capability caused by production pressure is delayed, gradual, and diffuse. It is distant in time and space from its cause. Managers are unlikely to attribute the cause of a throughput gap to the pressure they placed on workers months or even years before. Instead, they are likely to conclude that the workers have once again slacked off, requiring another increase in production pressure.

Workers often unwittingly conspire in strengthening the managers' attributions. Faced with intense production pressure, people are naturally reluctant to tell supervisors they can't meet all their objectives. The more effectively workers cover up the shortcuts they take to meet their throughput targets, the less aware managers will be of the long-run costs of production pressure. Unaware that improvement activity, maintenance, and problem solving have been cut, throughput appears to rise without requiring any sacrifices, reinforcing management's attribution that the workers really were not working hard enough. When managers eventually discover these shortcuts, their view of workers as untrustworthy is confirmed. Managers are then, as they see it, *forced* to monitor worker effort even more closely (e.g., more frequent status reports, stiffer penalties for missing targets, software for monitoring key-stroke rates of data entry operators). What starts as an erroneous attribution about the skills, effort, and character of the workers becomes true. Managers' worst fears are realized as a consequence of their own actions.

Consistent with our theory, we are not attributing these dynamics to unskilled, inexperienced, or ill-intentioned managers. Rather, the structure of the system inadvertently leads even many talented and dedicated managers into the capability trap, while at the same time providing compelling evidence that the sources of their difficulties lie in factors beyond their control, such as lazy workers, a "difficult" union, faulty machinery, or fickle customers. Managers are unlikely to escape the capability trap because they rarely realize they are in it. Instead, as capability stagnates despite repeated attempts at improvement, they slowly, perhaps reluctantly, but with increasing conviction, come to believe that their problems lie in the attitudes and character of the people that work for them. Having made such an attribution, the actions they take, while rational from their perspective, make the situation worse.

How Superstitious Learning Thwarts Improvement Programs

What happens when an organization stuck in the capability trap attempts to implement an improvement program? Performance is low and work pressure intense. In such environments, improvement programs add to the workload—the organization is so far behind that it cannot afford to cut back throughput. Indeed, in many organizations, management imposes aggressive stretch objectives for both throughput and improvement in the belief that aggressive goals are needed to shake things up and motivate people. In one firm we studied, the general manager laid out his goals for improving the product development process by saying:

> "We need a development process that is fast, is the best in the industry, and it needs to increase throughput by 50% in two years. And everyone must adhere to the same process."

At the same time, they launched many new development projects in anticipation of the expected productivity gains. Viewed through the lens of management's mental model these decisions were entirely rational. However, that mental model, conditioned by the self-confirming attribution error dynamics discussed above, led them to the erroneous belief that the delay between improvement effort and results was short and that their engineers were underutilized, undisciplined, unmotivated, and unwilling to adhere to the specified process.

The company spent millions and invested countless person-hours to create a new product development process. The new process included better technical tools, such as improved CAD/CAE/CAM systems, but also increased monitoring, including a structured stage-gate review process and mandated use of project management software. While there were some pockets of success, in most cases the effort had little impact. The leaders of the change effort often attributed its failure to the engineers' lack of discipline:

> "Engineers—by trade, definition, and training—want to forever tweak things. It's a Wild West culture."—Manager A

> "We went through a period where we had so little discipline that we really had the 'process du jour.' Get the job done and how you did it was up to you." —Manager B

> "A lot of the engineers felt that [the new process] was no value-add and that they should have spent all their time doing engineering and not filling out project worksheets. It's brushed off as bureaucratic."—Manager A

> "It was fair to say that a lot of engineers viewed this as a neat way to get some fancy tools and to hell with process."—Manager C

Yet, when we asked engineers why the effort failed, we got a different story:

> "We never had time to take the courses and get the equipment we needed to really make this stuff work. . . . It was really exhausting trying to learn how to use the tools and do the design at the same time. "—Engineer A

> "People had to do their normal work as well as [use the new project management system]. There just weren't enough hours in the day, and the work wasn't going to wait."—Engineer B

> "Under this system . . . the new workload was all increase. . . . In some cases your workload could have doubled."—Engineer C

> "How did we catch up? We stayed late. Most of the team was working from 7:00 a.m. to 8:00 p.m. and on weekends. A lot of people worked right through the Christmas vacation."—Engineer D

> "The new process is a good one. Someday I'd like to work on a project that actually uses it."—Engineer E

While managers felt the engineers had little interest in following the process, engineers became increasingly frustrated with leaders they felt had no understanding of what was really required to develop new products. Faced with the double bind of hitting aggressive performance targets and equally aggressive improvement targets, they were forced to cut corners while still appearing to follow the process. As one engineer remarked,

> "In many ways we worked around the [new] system. Good, bad, or indifferent that's what happened. We had a due date and we did whatever it took to hit it."

As management discovers the engineers' shortcuts and workarounds, their view that the engineers can't be trusted is confirmed, and they are forced to step up their monitoring. Faced with similar difficulties in its effort to implement a new product development process, a different firm even created a cadre of "compliance managers" whose sole job was to enforce adherence to their new development process.

Workers in such organizations quickly learn to hide problems from others. In all of the organizations we studied, engineers routinely neglected to reveal the existence of serious design issues for fear of retribution from managers. In one firm, the motto of the development engineers was "never reveal you have a problem until you also have the solution." In another, engineers called the weekly progress review meetings the "liars' club"—each participant overstated the progress of his subsystem and hid known defects from others in the hope that others would be discovered first, giving them time to catch up.[12] The consequence is long delays in the discovery of needed rework, greatly increasing costs, delaying launch, and, often, compromising quality.

The capability trap goes beyond low capability and high work pressure. Eventually it gets embedded in deeper structures, including incentives and corporate culture. As organizations grow more dependent on firefighting and working harder to solve problems caused by low process capability, they reward and promote those who, through heroic efforts, manage to save troubled projects or keep the line running. Consequently, most organizations reward last-minute problem solving over the learning, training, and improvement activities that prevent such crises in the first place. As an engineer at an auto company told us, "Nobody ever gets credit for fixing problems that never happened." Over time, senior management will increasingly consist of these war heroes, who are likely to groom and favor other can-do people like themselves. As described by a project leader we interviewed,

> "Our [company] culture rewards the heroes. Frankly, that's how I got where I've gotten. I've delivered programs under duress and difficult situations and the reward that comes with that is that you are recognized as someone that can deliver. Those are the opportunities for advancement."

Thus incentives and culture not only reinforce the tendency toward short-run thinking and working harder, but also are themselves shaped by that very short-term focus and work-harder mentality, creating another reinforcing feedback that intensifies the capability trap.[13]

An organization suffering from the self-confirming attribution error is poorly positioned to escape the capability trap. Improvement programs add stress to the organization, triggering greater work pressure that prevents people from investing in improvement, and encourages shortcuts. In such organizations many improvement programs never get off the ground. If, despite the work pressure, people do succeed in allocating more time to improvement, the result is a short-term drop in performance as time spent working falls before the investments in improvement bear fruit. Observing that performance is not improving, managers conclude the particular improvement method is not working and abandon it. Since the need to improve remains, they search for another, more promising tool, only to find it too suffers a similar fate. The result is growing cynicism among employees about "flavor of the month" programs.

More insidiously, these dynamics strengthen stereotypes and conflicts that not only hurt organizational performance but damage society. Consider, for example, how a senior manager explained why the product development improvement effort he ran had failed:

> "Program management and the disciplines associated with it continue to be a problem in my opinion in most western cultures. The people that are particularly rigorous and disciplined, the Japanese and the Germans, tend to be so by cultural norms. I can't tell you if it's hereditary or society or where it is they get it but the best engineers are those that tend to be the most disciplined, not as individual contributors but as team-based engineers. So there's a strong push back from the western type of engineer for much of this."

There is no mention of the structural features of the system or the pressure, felt throughout the organization, to deliver ambitious projects on time.

Instead, this manager blames the failure on the undisciplined character of "Western" engineers. Such attributions, here generalized to entire national groups, and invoking a disturbing racial and ethnic subtext, are typical of the fundamental attribution error. As these attributions are shared and repeated they become institutionalized. They become part of the corporate culture, and, as suggested by the quote above, can strengthen pernicious stereotypes and prejudices in society at large.

Overcoming the Capability Trap

So what can be done? The most important implication of our analysis is that our experiences often teach us exactly the wrong lessons about how to maintain and improve the long-term health of the systems in which we work and live. Successful improvement must include a significant shift in the mental models of those both leading and participating in an improvement effort. This insight was captured succinctly by one manager in a successful improvement effort:

> "There are two theories. One says, 'there's a problem, let's fix it.' The other says 'we have a problem, someone is screwing up, let's go beat them up.' To make improvement, we could no longer embrace the second theory, we had to use the first."

Once the cycle of self-confirming attributions is broken, any number of process improvement tools and methods can help improve capability. Without this shift, new tools and techniques, no matter how great their potential, are unlikely to succeed.

Breaking the cycle of self-confirming attributions is not easy, but it can be done. The following are examples from two organizations that overcame these difficulties and introduced successful improvement efforts.[14]

Du Pont

In 1991, an in-house benchmarking study documented a gap between Du Pont's maintenance record and those of the best performing companies in the chemicals industry. The benchmarking study revealed an apparent paradox: Du Pont spent more on maintenance than industry leaders but got less for it. Du Pont had the highest number of maintenance employees per dollar of plant value, yet its mechanics worked more overtime. Spare parts inventories were excessive, yet they relied heavily on costly expedited procurement of critical components. Overall, Du Pont spent 10-30% more on maintenance per dollar of plant value than the industry leaders, while overall plant uptime was some 10-15% lower.

An experienced manager, Winston Ledet, and a team charged with improving maintenance operations, developed a system dynamics model of these issues. The modeling process involved extensive hands-on workshops in which the team, assisted by an experienced modeler, discussed, tested, and changed the model as they identified areas needing improvement. Using the model as a laboratory to design and test different policies, the team gradually developed an appreciation for the capability trap and the paradox of high maintenance costs and low reliability.

To see how the capability trap arose in the chemicals industry, imagine the effects of cost cuts on maintenance, such as those beginning with the oil crisis of 1973 and subsequent recession. In chemical plants, when critical equipment breaks down, it must be fixed. Hence maintenance managers required to reduce costs must cut preventive maintenance, training, and investments in equipment upgrades. The drop in planned maintenance eventually causes breakdowns to increase, forcing management to reassign more mechanics from planned maintenance to repair work. Breakdowns then rise even more. As uptime falls, operators find it harder to meet demand and become less willing to take equipment down for scheduled maintenance, leading to more breakdowns and still lower uptime. More breakdowns simultaneously constrain revenue (by lowering production) and increase costs (due to overtime, expedited parts procurement, the nonroutine and often hazardous nature of outages, collateral damage, and so forth). More subtly, lower uptime erodes a plant's ability to meet its delivery commitments. As it develops a reputation for poor delivery reliability, business volume and margins fall further. The plant slowly slides into the capability trap, with high breakdowns, low uptime, and high costs.

Policy analysis showed that escaping the capability trap necessarily meant performance would deteriorate before it could improve: While continuing to repair breakdowns, the organization has to invest additional resources in planned maintenance, training and part quality, raising costs. Most importantly, increasing planned maintenance *reduces* uptime in the short run because operable equipment must be taken off-line for the planned maintenance to be done. Only later, as the *Reinvestment* loop begins to work in the virtuous direction, does the breakdown rate drop. Fewer unplanned breakdowns give mechanics more time for planned maintenance. As maintenance expenses drop the savings can be reinvested in training, parts quality, reliability engineering, planning and scheduling systems, and other activities that further reduce breakdowns. For example, upgrading to a more durable pump seal improves reliability, allowing maintenance intervals to be lengthened and inventories of replacement seals to be cut. Higher uptime also yields more revenue and provides additional resources for still more improvement. All the positive feedbacks that once acted as vicious cycles dragging reliability down become virtuous cycles, progressively and cumulatively boosting uptime and cutting costs.

Now the challenge facing the team was implementation. They knew nothing could happen without the willing participation of thousands of people, from the lowest-grade hourly mechanic to regional vice presidents. They also realized that their views had changed because they had participated in the modeling process. Somehow they had to facilitate a similar learning process throughout the plants.

The team converted the maintenance model into an interactive role-playing simulation they called the Manufacturing Game.[15] The game is closely based on the model and realistically captures the time delays, costs, and other parameters characterizing typical plants. They embedded the game in an interactive workshop designed to create an environment for learning that addressed emotional as well as cognitive issues. The process at Du Pont's Washington Works complex in Parkersburg, West Virginia, was typical:

> The team was initiated with a two-day learning lab . . . learning the concepts of defect elimination and experiencing the Manufacturing Game. . . . The material is presented in the form of lectures, skits and participative exercises in an off-site environment. Posters and music are used. The atmosphere is much different than routine plant meetings or training, to open up their thinking. . . . Through interactive exercises, the team develops their personal aspirations for improving the area where they have chosen to work. . . . [Then] they . . . develop an action plan to immediately start working.[16]

Despite its many simplifications, the game quickly becomes in many ways a real plant with real emotions and conflicts among players. Initialized with high breakdowns and low uptime, the people playing the role of operations managers face intense pressure to keep equipment running and often rebuff attempts to increase planned maintenance, just as in the real world. Players who stick with the prevailing cost-minimization, work-harder, reactive maintenance policies can keep costs low for a while. However, as defects accumulate, uptime slowly sinks while costs rise. Teams who follow a planned maintenance strategy first find costs rise while uptime falls. Soon, however, costs begin to fall and uptime rises. The game allows people to experience the worse-before-better dynamic in a few hours instead of a few months. For many, the game was the first time in their careers they experienced the possibility that improvement was actually possible.

The game and learning laboratory proved popular. However, playing it once was not enough. The team found that they had to run several workshops for a given plant before a critical mass emerged to lead action teams and put proactive maintenance policies into practice. Individual plants needed the capability to run the game so their own people, with their site-specific experience and legitimacy, could run it on demand. By the end of 1992, some 1200 people had participated in the workshop, and more than 50 facilitators had been certified.

At plants that implemented the program by the end of 1993, the mean time between failure (MTBF) for pumps (the focus of the program) rose by an average of 12% each time cumulative operating experience doubled. Direct maintenance costs fell an average of 20%. In 23 comparable plants not implementing the program the learning rate averaged just 5% and costs were *up* an average of 7%. Washington Works boosted production capability 20%, improved customer service 90%, and cut delivery lead time by 50%, all with minimal capital investment and a drop in maintenance costs. For the company as a whole, conservative estimates exceed $350 million/year in avoided maintenance costs alone.

However, success creates its own challenges. One issue related to the persistence of the cost-saving mentality. A member of the modeling team commented, "As soon as you get the problems down, people will be taken away from the effort and the problems will go back up." In fact, mandated corporate cost-cutting programs did cause significant downsizing throughout the entire company, weakening the reinvestment feedback and limiting their ability to expand the program. Winston Ledet took early retirement and began working with other companies interested in the game and learning lab. These firms include other chemicals manufacturers along with firms in the energy, automotive, and high-tech sectors.

British Petroleum

One of the organizations Ledet worked with after leaving Du Pont was British Petroleum's refinery in Lima, Ohio. Founded in 1886 by John D. Rockefeller, and once "Queen of the Fleet," the refinery engaged in cost cutting during the 1980s that triggered the vicious cycle of increasing breakdowns, higher maintenance costs, and less planned maintenance, pushing it into the capability trap. By the early 1990s, Lima lagged well behind other U.S. refineries. BP began to think about selling or closing the facility.

In 1994, the Lima facility introduced the maintenance learning lab and other system dynamics tools. It was not a top management intervention: The original champions were an equipment specialist, a maintenance training supervisor, and an engineer. Successful pilot projects led to favorable word of mouth; eventually 80% of all employees participated in the program. Soon dozens of improvement teams were in place. During the first six months, maintenance costs ballooned by 30%. Having experienced it in the game, management was prepared for the worse-before-better dynamic, and focused on the improvements generated by the action teams.

In January 1996, BP announced that it intended to sell the Lima refinery and stepped up its cost cutting and downsizing. A few months later BP stunned the employees by announcing that it could not find a buyer at a satisfactory price and would therefore close the refinery. The announcement was a deep blow to the workers and the community. One of the most important businesses in the community, the refinery employed 450 people and pumped more than $60 million per year into Lima's depressed economy. Some employees became discouraged and questioned the value of the learning lab and improvement program. A few transferred to other BP facilities or left altogether. Winston Ledet described what happened next:

TABLE 1. Improvement at the Lima Refinery

- Pump MTBF up from 12 to 58 months (failures down from more than 640 in 1991 to 131 in 1998). Direct savings: $1.8 million/year.
- Hydrocarbon flare-off down from 1.5% to 0.35%, saving $0.27/barrel and improving environmental quality.
- On-line analyzer uptime improved from 75% and not trusted to 97% and trusted, permitting real-time optimization of product flow. Savings: $0.10-0.12/barrel.
- Safety incidents and lost hours cut by a factor of 4.
- Thirty-four production records set.
- Cash margin improved by $0.77 per barrel of oil processed.
- Total new value created: $43 million/year. Total cost: $320,000/year. Ratio: 143:1.
- Learning initiative under way for other BP facilities around the world.

Source: Paul Monus, "Proactive Manufacturing at BP's Lima Oil Refinery," presented at National Petroleum Refiners Association Maintenance Conference, May 20-23, 1997, New Orleans; J. Griffith, D. Kuenzli, and P. Monus, "Proactive Manufacturing: Accelerating Step Change Breakthroughs in Performance," NPRA Maintenance Conference, MC-98-92, 1998; Paul Monus, personal communication.

> "For those who decided to stay with the ship, a new spirit emerged. They realized that they needed a future in Lima and should take responsibility for creating that future. The first step was to ensure that the exit of many experienced people did not throw them back in the reactive mode. . . . It actually created a clearer focus for the people who remained. They were all there because they had chosen to be there."

Soon the impact of the new maintenance policies and attitudes was clearly visible (Table 1).

These dramatic improvements did not go unnoticed. On July 2, 1998, the banner headline of the *Lima News* announced "Oil Refinery Rescued." Clark USA, a privately held *Fortune 500* company with refining and distribution interests, agreed to buy the Lima refinery for $215 million and keep it operating.

The Du Pont and BP cases illustrate the power of a shift in mental models. The model, game, and workshop do not teach anyone how to maintain equipment. For example, a BP team reduced butane flare-off to zero, saving $1.5 million/year and reducing pollution. The effort took two weeks and cost $5000, a return on investment of 30,000%/year. Members of the team had known about the problem and how to solve it for eight years. They already had all the engineering know-how they needed, and most of the equipment and materials were already on site. What had stopped them from solving the problem long ago? The only barrier was the mental model that there were no resources or time for improvement, that these problems were outside their control, and that they could never make a difference.

The modeling process and the resulting game were effective because they eliminated many of the impediments to learning in the real system. Dynamics such as the progressive slide into the capability trap that normally play out over years or even decades could be experienced in just a few hours. Unlike the real world, people could take different roles: A mechanic playing the role of plant manager might find himself with low uptime and then cut preventive maintenance to avoid equipment takedowns and cut costs. Seeing people from different functions and backgrounds enacting the same behaviors helped break the vicious cycle of self-confirming attribution errors and blame. The systems thinking process enabled people to experience for themselves the long-term, organization-wide consequences of their actions. They discovered how to use initial successes to create resources for further improvement and how to survive the short-run drop in performance. They saw how small actions could snowball into major gains. Most importantly, they learned that they could, after all, make a difference.

Notes

1. See G. Easton and S. Jarrell, "The Effects of Total Quality Management on Corporate Performance: An Empirical Investigation," *Journal of Business*, 2 (1998): 253-307; K. Hendricks and V.R. Singhal, "Quality Awards and the Market Value of the Firm: An Empirical Investigation," *Management Science*, 43/3 (1996): 415-436.
2. See Easton and Jarrell, op. cit.; Darrell Rigby, "Management Tools and Techniques: A Survey," *California Management Review*, 43/2 (Winter 2001): 139-159.
3. See J. Pfeffer and R. Sutton, *The Knowing-Doing Gap* (Boston, MA: Harvard Business School Press, 2000); K. Klein and J. Sorra, "The Challenge of Innovation Implementation," *Academy of Management Review*, 4 (1996): 1055-1080.
4. Published summaries can be found in: J. Sterman, N. Repenning, and F. Kofman "Unanticipated Side Effects of Successful Quality Programs: Exploring a Paradox of Organizational Improvement," *Management Science*, 43/4 (1997): 503-521; N. Repenning and J. Sterman, "Getting Quality the Old Fashion: Self-Confirming Attributions in the Dynamics of Process Improvement," in R.B. Cole and R. Scott, eds., *Improving Theory and Research on Quality Enhancement in Organizations* (Thousand Oaks, CA: Sage, 2000), pp. 201- 235; E. Keating and R. Oliva, "A Dynamic Theory of Sustaining Process Improvement Teams in Product Development," in M. Beyerlein and D. Johnson, eds., *Advances in Interdisciplinary Studies of Teams* (Greenwich, CT: JAI Press, 2000); R. Oliva, S. Rockart, and J. Sterman, "Managing Multiple Improvement Efforts: Lessons from a Semiconductor Manufacturing Site," in D. Fedor and S. Ghosh, eds., *Advances in the Management of Organizational Quality* (Greenwich, CT: JAI Press, 1998), pp. 1-55; J. Carroll, J. Sterman, and A. Markus, "Playing the Maintenance Game: How Mental Models Drive Organization Decisions," in R. Stern and J. Halpern, eds., *Debating Rationality: Nonrational Elements of Organizational Decision Making* (Ithaca, NY: ILR Press, 1997).
5. For formal models of implementation see: Sterman, Repenning, and Kofman op. cit.; N. Repenning, "Drive Out Fear (Unless You Can Drive It In): The Role of Agency and Job Security in Process Improvement Efforts," *Management Science*, 46/11 (2000): 1385-1396; N. Repenning, "A Dynamic Model of Resource Allocation in Multi-Project Research and Development Systems," *System Dynamics Review*, 16/3 (2000): 173-212; N. Repenning, "A Simulation Based-Approach to Understanding the Dynamics of Innovation Implementation," *Organization Science* (forthcoming).
6. Repenning and Sterman (2000), op. cit.

7. See A. Schneiderman, "Setting Quality Goals," *Quality Progress* (April 1988), pp. 55-57; J. Sterman, N. Repenning, and F. Kofman, op. cit.
8. Schneiderman, op. cit.
9. For an example, see Sterman, Repenning, and Kofman, op. cit.
10. The dynamics of project management and software development have been treated extensively in system dynamics. See, for example, T. Abdel-Hamid, "The Economics of Software Quality Assurance: A Simulation-Based Case Study," *MIS Quarterly*, 3 (1997): 395-411; D. Ford and J. D. Sterman "Dynamic Modeling of Product Development Processes," *System Dynamics Review*, 14 (1998): 31-68.
11. For an excellent summary of attribution theory, see S. Plous, *The Psychology of Judgment and Decision Making* (New York, NY: McGraw-Hill, 1993).
12. See also George Roth and Art Kleiner, *Car Launch* (New York, NY: Oxford University Press, 2000).
13. See N. Repenning et al., "Past the Tipping Point: The Persistence of Fire-Fighting in Product Development," published in this issue [*California Management Review*, 43/4 (Summer 2001)].
14. These cases are discussed in depth in J. Sterman, *Business Dynamics: Systems Thinking and Modeling for a Complex World* (New York, NY: Irwin/McGraw-Hill, 2000), chapter 2.
15. See Winston Ledet, "Engaging the Entire Organization: Key to Improving Reliability," *Oil and Gas Journal*, 97/21 (1999): 54-57.
16. R. Tewksbury and R. Steward, "Improved Production Capability Program at Du Pont's Washington Works," Proceedings of the 1997 Society for Maintenance and Reliability annual conference.