The IS Productivity Challenge

Motivating your people to produce breakthroughs

by
Jerry Straus
David Spiwack

One of the greatest challenges facing IS departments today is raising productivity. Research reveals that while traditional programming tools and techniques may have effectively reached their limit in achieving incremental increases in productivity, a huge—and largely untapped—potential for productivity breakthroughs exists in the motivation and management of IS professionals.

In short, *people* are more important than tools when it comes to achieving quantum leaps in productivity in the corporate IS environment. Knowing how to manage and motivate your programming team can mean the difference between catching up with your applications backlog or falling farther behind.

A LOOK AT THE PROBLEM

If you feel overwhelmed by the demands being placed on your department today, you're not alone. According to a recent report from *Business Week* [1], the average length of programs is increasing 25 percent per year. Yet the number of new programmers is up by only 4 percent a year.

In addition, maintenance is now estimated at 60 to 80 percent of the life-cycle costs, so the majority of IS resources are spent just to keep existing software systems up to date.

Users and non-IS management do not always appreciate what is required to create new software systems or to maintain existing systems; hence, resource allocation may not increase in proportion to user demand.

As a result, software professionals are being asked to do more with fewer resources in less time, making project slippages and hefty backlogs a fact of life. (Our informal survey shows that only one out of five IS projects is completed by the originally scheduled deadline.) This forces IS managers to search for ways to get more done with the resources available—in other words, to make every programmer much more productive. But what will it take?

TOOLS ALONE ARE NOT THE ANSWER

Tools and methodologies can only go so far in helping IS solve the productivity problem. For example, ads for CASE tools regularly claim productivity increases of 20 to 30 percent, yet far greater increases are needed to meet the rising demand for software.

If tools and methodologies are not the solution, what is? People—specifically, managing and motivating your IS professionals so they are more productive, individually and as a team.

An article in *Fortune* [3] reports, "For all the technical problems, the biggest obstacles to effective economical software development are managerial: bad planning, organizational rivalries, unrealistic scheduling, or the inability to grasp the business problems they are trying to solve."

In addition, numerous productivity surveys confirm that good people are more critical than tools, techniques, methodologies, and other "silver bullet" solutions to productivity problems.

Assuming you already have good people, the challenge is to enhance their performance. Most corporate IS departments have made significant investments in tools and techniques but have paid relatively little attention to improving "people productivity." Yet growing evidence suggests the latter can have enormous benefit.

Specifically, if tools and methodologies are working, focusing on people and project teams can boost results an order of magnitude. And if tools are not working, focusing on your team members—and the interactions between them—can, in our experience, turn that failure around.

Example: A team at a large manufacturing firm was asked to use an advanced code generator to develop a system. Team members found it awkward to work with and complained that the tool was costing time rather than saving it. Conclusion: Even the most sophisticated tool can be a detriment rather than a boon to productivity without the cooperation of the human element.

Effectively involving people can leverage whatever impact tools are designed to make.

EVIDENCE OF A NEW PERSPECTIVE

As both participants and observers, we have been involved in over 35 IS projects where breakthroughs in productivity were achieved. In all cases, there is evidence that motivation of the team to achieve extraordinary results—and not a new tool or technical methodology—was the primary factor leading to breakthrough. For example,

★ At a major insurance firm, a project team was asked to complete in five months an automated billing system for which the team had allocated nine months. "I didn't know how we were going to do it," said a director in the information systems department. "To reduce the schedule seemed impossible."

Yet the team completed the project in five months, as requested—a 25 percent reduction in elapsed calendar days—with no reduction in quality, no extra overtime, and a return on expenditures of approximately 3 to 1.

★ Another corporation had originally allocated a team of 36 people for two years to develop a new product. But then they made a commitment to achieve a productivity breakthrough by completing the same system in one year with only 18 people, a savings of 54 people-years. And they succeeded.

★ A vice president with a major computer company reports: "To help microcoders consistently achieve peak performance, they were taught specific techniques for solving problems, thinking creatively, and communicating to get clear commitments from their coworkers. The results don't need any elaboration; these people have significantly increased their productivity and expect to improve an important delivery date by six months."

Our observation? When people are "empowered"—that is, when they take ownership and *enjoy* the challenge they are working on—their productivity can increase by an order of magnitude. And as a result, the team can achieve results previously deemed "impossible."

WHAT HAVE WE LEARNED?

After observing productivity breakthroughs in more than 35 IS project teams, we carefully studied them to see if any themes or factors were common to such instances of extraordinary achievement.

Though every organization has had instances of extraordinary performance, our goal was to answer the question: "What is required to *deliberately* instigate that kind of accomplishment?"

What we discovered was that in each instance, the project leaders and team members performed beyond expectations because they were sufficiently motivated to do so. However, the type of motivation needed to lead a project team to this level of performance goes beyond the very legitimate motivations that already exist in the corporate environment (e.g., fear of failure, desire for a raise or promotion, bonuses, praise from superiors, award and photo in company newsletter).

There are additional, more subtle communications between management and the project team required to create an environment that frees seeing, thinking, and action while building a culture for innovation, creativity, ownership, and teamwork.

In an interview in the *Harvard Business Review* [4], Raychem founder and CEO Paul Cook said,

> More important than salaries, bonuses, or promotions, they [employees] want to identify with the success of their organization. . . . And their greatest reward is receiving acknowledgment that they *did* contribute to making something meaningful happen.

But convincing individuals on a project team that the work they're doing is special and meaningful is no small task. Let's take a look at some of the requirements necessary in creating an IS environment where a "breakthrough team" can grow and flourish.

Frame the Effort as a Special or Extraordinary Effort

The first step in deliberately creating an environment for productivity breakthroughs is to single out a particular project as a breakthrough effort and to communicate this to the team leaders and members.

To do this, you must be clear on the definition of a breakthrough project. We define a breakthrough project as one that enables an IS team to meet a deadline or achieve a result previously thought to be unlikely, impractical, or even impossible. It exceeds what's predictable or expected based on existing criteria for measurement. An identifying characteristic of a breakthrough project is the commitment of the team to achieving a breakthrough result without knowing, at the time it commits, how it will be able to do it.

In our observation, a key factor in framing an effort as a breakthrough is the time and effort invested by both senior and project-level management in assembling the team before the start of the project to emphasize the importance of the work.

To achieve a breakthrough result, management must communicate to the team, in a credible fashion, why a breakthrough result on this particular system is needed. Team members need to understand why achieving a breakthrough on the project is important to the corporation, how it will enhance profitability, why users are so eager to have the system, or even how achieving a successful result can enhance the reputation of the IS department within the organization.

Another motivating technique is to discuss frankly the negative consequences of *not* achieving superior productivity results on this particular project. (For example, perhaps department credibility is at stake, and users have threatened to retain outside programmers if IS cannot meet an extraordinarily tight deadline for producing the desired system.)

In most organizations, this kind of communication rarely takes place. Often there is no formal kickoff meeting. And even when IS managers do hold formal meetings, they may not have mastered the communication techniques necessary to empower the team with the required sense of mission and to overcome the team's skepticism and belief that such talk is merely "the company line."

The first step, then, is to assemble the breakthrough team and declare the project a breakthrough effort.

Give Team Members a Choice of Whether to Participate

Turning an ordinary IS project team into a breakthrough team is possible only when team members are given a free choice of whether to treat the effort as a productivity breakthrough project or not. Reason: When a person is *told* to adopt an attitude, goal, or objective, no real choice—and hence no genuine commitment—can take place.

Giving the team a choice of whether to participate and how to proceed results in greater "ownership" of the project by team members. And when a programmer or systems developer invests his time, intelligence, and effort in a project because he *wants* to—not because he is told to—he gains an extraordinary relationship with the work: one of *self*-motivation versus forced motivation.

What happens if the team members say "no" when given this choice of whether to participate? In some instances, there may need to be some negotiation between management and team members on issues of concern to the team, such as how the project will affect careers or what happens to the IS professional if his or her team fails to achieve the breakthrough.

But in our firm's experience working with numerous such efforts, the teams have ultimately always responded "yes" in some fashion to this challenge . . . as they most often will when they believe the choice is up to them.

This doesn't mean that every team member jumps on the bandwagon at once. Far from it. Rather, as in most things, commitment to breakthrough starts with one person (or a small group of individuals) standing up and saying, "Count me in; I say this will be a breakthrough." Over time, a critical mass of team support will develop. (Some team members may remain skeptical throughout yet still make a significant contribution.)

Examine Preconceptions and Cultural Limits

A preconception is a notion about how the job needs to be done based on existing procedures, methodologies, and rules established through a combination of good and bad past experience.

A cultural limit is a perceived limitation on a project imposed by an organizational structure. The cultural limitation may be inherent in the normal functional work flow. Or it may involve how one group in an organization deals with—and feels about—another group.

For instance, in the IS environment, the people who do the testing often report that they are forced into pressure situations in which they must compress their work schedule to meet an original deadline and make up time lost by others who did not meet their own schedules (such as systems developers). This can breed rivalries—even animosity—harmful to team efforts.

And in many corporations, users and IS professionals have an adversarial rather than a team relationship. Users complain that systems developers "want to do their own thing" rather than give the users what they want; developers, after sitting down with users to define requirements, come away shaking their heads and muttering, "Those people have *no idea* of what they want!"

How do you get different professionals within the team to accept their interdependence? "It helps if you can get the individual members of a team to recognize the preconceptions they bring to the table," writes Walter Kiechel [2], paraphrasing Analog Devices CEO Ray Stata. "When they shear through the layers of opinion to get at the facts, they quickly realize how much they all need one another." Hence, including user representatives in these early activities is critical.

Speculation

In the breakthrough environment, teams are managed at the outset in the spirit of *speculation*, instead of mandating performance from the top down.

Speculation is management through a series of "what if" or "what's possible" scenarios. It involves techniques such as questioning and brainstorming. Such interactions challenge people's impatience and counteract their tendency to resort to what's familiar and has been done before.

Making commitments

Another principle that emerges from our study of breakthrough teams is that breakthroughs come when team members and leaders make extraordinary commitments—that is, a commitment inspired by something bigger than the immediate project goals and objectives, by something truly meaningful to them.

One frequently cited example of an "extraordinary commitment" is the famous declaration by President Kennedy in the early 1960s that the United States would put a man on the moon by the end of the decade. Although we had not developed space technology to the point where this was clearly achievable (in fact, the Russians had launched the first successful orbiting satellite and put the first man into orbit), a whole nation took up Kennedy's visionary commitment. And in 1969, an American, Neil Armstrong, became the first man to walk on the moon.

In an extraordinary commitment, the person takes the bold leap and says, "I will achieve so-and-so result, even though it seems impossible right now, and I have *no idea* of how to go about it!"

To obtain such commitment requires an unusually frank dialog among team members and management, which includes the opportunity for them to authentically examine their own relationship to the project, their work, and their career.

There's strength in this type of commitment because it empowers people to think creatively beyond their conventional mind-sets. Instead of looking back to past practices, the technical professional must now *invent for himself* a new way to achieve this unprecedented result. And he does—more often than you'd expect.

Recall our earlier case history of how a team at a large insurance firm completed an important software billing system in five months after originally allocating nine months. A major factor enabling this success was the commitment of the manager and the team members to achieving the goal.

"Once you commit inside," said one team member, tapping his finger to his chest, "you'll find a way to do it. You stop looking at it as something that can't be done, and you come up with ways to make it happen."

Distinctions in commitment

Management must distinguish between two basic types of commitment: commitment to an objective and commitment to a vision. Only the latter seems to empower IS teams to achieve productivity breakthroughs.

For the technical professional, a commitment to a vision may be the commitment to achieving some extraordinary or unprecedented technical feat, for example, to develop a state-of-the-art system or product in his area.

By comparison, an example of commitment to an objective is, "We will complete phase III of system ABC by June of 1992."

The commitment to an objective lacks the power to motivate team members to produce extraordinary results, especially on a pressure project. After all, how many of us become enthusiastic when ordered to do more work in less time?

It is only around a commitment to a vision that the team can be galvanized. What motivates IS professionals and other workers is to be part of a meaningful achievement or significant effort. And that's what commitment to a vision offers them.

We find that once a team commits to a vision, it then creates appropriate objectives consistent with achieving that vision. The problem in implementing this idea is that management, while skilled in setting objectives, has little experience in eliciting team commitment to a vision or broader goal. However, this is a skill that can be gained.

In our work, we like to ask team members, "What about this project will make it worth your jumping out of bed in the morning and will cause you to look forward to coming to work?" You can include each team member's commitment to personal growth, education, and career development—as well as his or her commitment to a breakthrough technical achievement—as part of the project design.

Giving team members a chance to address their relationship to their work in a way that is truly meaningful to them provides the motivation to achieve extraordinary results. Remember, this will require more effort up front in the formulation of the project at a time when most are eager to "get started" and may have little patience for these unconventional activities.

Eliciting the commitment of the team to the extraordinary effort requires that the manager be the first one to "take the plunge." That is, once the manager declares the project will be a breakthrough effort, then team members can consider it for themselves. And, once convinced that the effort is genuine, the team members often make more aggressive commitments than the manager might have been willing to mandate or request from them on his own.

Experience shows that IS professionals are willing, able, and actually eager to operate in this manner versus "business as usual." For instance, one development manager encourages commitment by suggesting to team members that they act as if they're writing programs for their own use. "The results couldn't be better," he reports. "It's made highly motivated people even more effective, increased productivity, and heightened morale."

Breakdowns

Another key concept of the productivity team is that *breakdowns* frequently lead to *breakthroughs.* Let's examine that statement a bit more closely.

A "breakdown" is a seemingly insurmountable obstacle or problem that might lead to missing a deadline, going over budget, making major coding or design errors, or other things we typically view as "mistakes."

The conventional management model is to avoid breakdowns and to design projects so no slippages occur. This is accomplished by setting "realistic" project goals based on past experience.

The breakthrough model, on the other hand, *deliberately* causes breakdowns by having team members commit to goals that are in some way unpredictable. Obviously, the more ambitious your goal, the greater the potential for major breakdowns.

Typically, in the conventional approach to project planning (an approach embodied in most commercially available project management software packages), a step-by-step plan is created. This plan spells out in detail all the specific tasks and activities needed to achieve the overall project goal.

The breakthrough model, in contrast, uses *pathways* instead of *plans* to guide team members toward the objective. Unlike a plan, which has all the answers, a pathway points the way but doesn't specify how to get there. One expert observes, "A pathway is like a plan with holes. The 'holes' are the breakdowns you encounter."

Why design the project so that breakdowns are inevitable? We find that breakthroughs come when breakdowns occur and *the team remains dedicated to sticking with the original goals,* rather than simply moving the deadline, changing the project, or throwing more resources at the problem.

(As an aside, note that most organizations have their own accepted practices for changing the original commitment when things don't work out. For instance, in the corporate IS environment, when a team can't deliver a version with the full functionality, they then deliver a version with limited functionality, saying that the rest of it will appear in "phase II" of the project; phase II becomes the euphemism for "we didn't meet our commitment.")

In our experience working with dozens of IS departments at major corporations, we find that a breakthrough often comes soon after the team chooses to remain with the original commitment in the face of a breakdown. The breakthrough might be a new technical idea or simply an old solution that has never before been applied to the existing area of concern.

Why do breakdowns often precede breakthroughs? Each breakdown that occurs focuses the programmer's or project team's attention on what *missing* know-how, expertise, or experience is required to achieve the desired result. In doing so, the breakdowns point the way to pathways that lead to innovative solutions not previously considered.

To paraphrase Eric Hoffer in a way that speaks to the challenge of IS professionals, "The most gifted members of the human species are at their creative best when they cannot have their way."

RECOGNIZING THE RISK

Recognizing that breakthrough efforts naturally involve a significantly higher degree of risk than conventional efforts, management must be prepared to both support—and deal with—the consequences of this elevated risk. This requires tolerating the discomfort of longer periods of uncertainty throughout the project (something to which most IS professionals are unaccustomed).

The corporate culture in large organizations is in many ways the opposite of the breakthrough mentality. It says, "Avoid risk, don't make any promises you can't keep, follow proven methods, and when in doubt, do what was done last time." The breakthrough model, on the other hand, says that predictable behavior yields predictable results, and the only way to achieve extraordinary productivity is to get your people to take chances, aim higher, and reach farther.

WHAT'S NEXT?

One question senior managers frequently ask us is: "Even if we achieve a breakthrough on project X, how does this process benefit all the other efforts I've got going?"

Being able to achieve a breakthrough on even a single development effort makes a distinct alteration in the IS culture at large because it demonstrates to everyone in the organization that people can do extraordinary things if they commit and are willing to take the risk. Other project teams begin to put breakthrough principles into action merely as a result of witnessing the excitement that occurs when people are working on something they care about and have taken ownership of.

THE NEED FOR PRODUCTIVITY BREAKTHROUGHS IN THE 1990s

Why are productivity breakthroughs necessary? Because in today's fast-paced business environment, there's more pressure than ever before for us to come up with innovative, effective solutions to information handling problems. Yet, as *Fortune* observes, "All over the world, senior executives are wondering why software development is so expensive [and] why it takes so long" [3]. Senior management wants miracles from us but has left it up to the IS manager to discover how to deliver these miraculous results.

As Jerry Wind, a professor at Wharton Business School, observes, managers in the 1990s must learn to deal with constant, rapid change. Conventional management techniques yield only conventional results and small, incremental increases in productivity. The breakthrough model offers a new yet tested management model with a decade of documented success increasing productivity in IS departments or major corporations nationwide.

REFERENCES

1 "Special Report: The Software Trap," *Business Week*, May 9, 1988.

2 Kiechel III, W., "The Organization That Learns," *Fortune*, March 12, 1990, pp. 133–136.

3 Schlender, B. R., "How to Break the Software Logjam," *Fortune*, September 25, 1989, pp. 100–112.

4 Taylor, W., "The Business of Innovation: An Interview with Paul Cook," *Harvard Business Review*, March–April 1990, pp. 97–106.

Jerry Straus and David Spiwack are vice presidents of JMW Consultants, Inc., a New York City–based management consulting firm helping corporations achieve productivity breakthroughs on specific projects and throughout their organizations. For more information, contact JMW Consultants, Inc., Dept. AP, 12 East 41 Street, New York, NY 10017 (212/683-8989). ★