Center for Quality of Management Journal

Reprint No. RP13120

Volume 11, Number 2 **Spring 2003**

ISSN: 1072-5296

Getting the Most Out of Technologists[1]

By David Walden

David Walden was a member of the original CQM design team in 1990, representing founding member Bolt Beranek and Newman Inc. He has remained involved with the CQM in a variety of capacities ever since. In particular, he is co-author, with Shoji Shiba, of the CQM's book *Four Practical Revolutions in Management*.

Introduction — Where my opinions come from and what they're worth

My credentials to talk about technologists are: ten years as a computer programmer (including co-authoring the original packet-switching software that evolved into the Internet), ten years as a technical manager, and ten years as a general manager — all at high tech companies — and a few more years studying and teaching business improvement in high tech companies and at MIT. However, I am not an academic and my views on technologists and getting the most out of them primarily come from my own observations and experience. I'll tell you what I think and what makes sense to me. I make no claim to have "the truth." It's up to you to decide if anything I say is relevant to you and your situation.

Since this paper is about getting the most out of technologists, it is aimed primarily at those who are trying to do this — technical managers and leaders. However, I hope that some of my thinking is also useful, in some way, to technologists themselves.

1. Roles

What do technologists want from their leaders or managers? In good times, engineers[2] want nothing — they want to be left alone (after their projects are funded, of course). In bad times, engineers want leaders solve all the problems immediately so times are good again.

Fundamentally, leaders or managers can't do anything on their own except choose people and allocate money. If something is to be accomplished, the employees must be engaged to do it. This means the leader must somehow connect the his or her insight about where the company needs to go to the interests of the employees It also means the leader or manager must motivate or force the employees to actually change their behavior to do what the leader and they have decided to do.

Such motivation or force is all about installing processes, roles, systems, structures, etc., to encourage and support needed behavior. It's almost not at all about cheer leading. In fact, cheer leading ("we're going to do this, we're going to do that, you can do it!") runs the risk of suggesting or promising things that then won't be achieved which will make the technologists even more skeptical about the capabilities of the leader and how in touch he or she is with what is actually going on.

[1] This paper is a refined version of notes for a presentation given at a conference of people interested in micro computers, etc., at Asilomar State Park, California, April 25, 2002.

[2] I use "technologists," "engineers," and "employees" (meaning the technology or engineering employees) interchangably in this paper.

Of course, it is important to explain to employees what is happening — but the talk shouldn't get ahead of the walk. Frank Pipp, a one-time top manager of Xerox, has said, "Employees can smell management hypocrisy at one part in a million." Employees tend to assume management hypocrisy and are so alert for evidence of it that a well-intentioned mistake is assumed to be an example of hypocrisy.

Often easier than trying to change people is understanding people's capabilities and getting them in the roles where they can best contribute (even as you try to help them improve in areas of weakness).

I don't know how it is in your field. In my field of software development there are vast differences in levels of capability from person to person. Look at Figure 1 (I'm not sure what the correct shape of this curve is, but regardless of whether I have the various slopes and intervals right, this figure helps make my point).

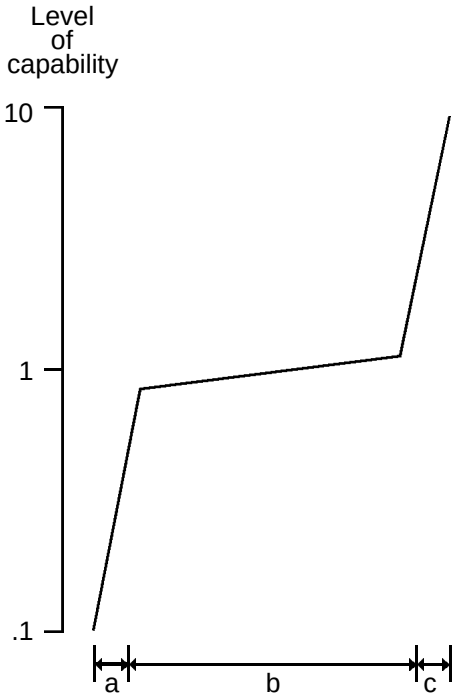

Figure 1: Levels of Capability

The majority of people are average performers, in interval b. There are a few star performers (interval c); their levels of performance can be can be a factor of ten (or more) higher than average performers. Finally, there are a few subpar performers (interval a). Their performance can be a tenth that of the average performers.[3]

The key management job is to find the best characteristics and capabilities of members of the team and to make the best use of them. In all fields of endeavor, maximum success requires taking best advantage of the capabilities of the star performers (those in interval a in the figure). These are the technical leaders. These are the clutch performers. These are the people whose success can enable the success of the whole team. (Unfortunately, there aren't many of these.) You must support these people so you can get their best from them. This means that you have to array the average performers (interval b) around the star performers to enable the star performers to give their best and to expand the impact of both the star performers and the average performers.

[3] Although the log scale of the vertical axis of the figure doesn't allow it to be shown, in fact the subpar performers can produce negative results.

Of course, you also need to help remove some of the most counter-productive characteristics of some members of the team, and you need to completely remove detrimentally weak people (interval c) from the team, after you are confident you have judged them correctly.[4]

[4] Make this decision sooner rather than later — it's especially easy to make in the first weeks after a new employee joins your group when you can still credibly give the excuse that it was a hiring misunderstanding rather than the person being a bad person.

In addition to assigning people to roles based on level of capability, you also need to make best use of different areas of capability. For instance, some people are more technically innovative (coming up with new ideas for products or new ways of implementing them). Other people are naturals at what some people call gatekeeping (knowing what's going on throughout the organization and how to connect to relevant information throughout the world). Some people are extra good at debugging, maintaining, or customer support. Also, many people can do a project that takes a few weeks or a few months. Few people can successfully carry out a project that takes many months: by the time such a large project has been underway for many months, it is surely in trouble regardless of how much planning went into it. As the project gets more confusing and thus more overwhelming, most people have a hard time continuing to concentrate on the project; the people who have the capability to regroup their thinking as necessary and to continue slogging to completion are few and far between and worth their weight in gold.

Remember, also, that the curve of Figure 1 applies in different areas; there will be superstar maintainers, debuggers and customer support people, just as there can be superstar innovators.

It's also useful to consider the roles the various management roles. We often use the words manager and leader interchangably, and both leaders and managers are often told (these days) that they need to be coaches to their employees.

While I don't think we should be fussy about what words we use in ordinary conversation, I do think that it is useful to explicitly think about three different roles that might be represented by the words manager, leader, and coach. I think of a manager a person who is responsibility for allocation of resources and responsibility. A leader is someone who pushes others in the direction the company needs to go. A coach helps people see what they can't see for themselves.

It is difficult for same person to play all three roles:

- Of course, a manager should be concerned about the well-being of his or her employees, however, I believe being a manager and a coach are largely incompatible — a coach needs to be fundamentally concerned with the development of the employee; manager is a role in the formal hierarchy of the company and a manager can never fully forget his or her responsibility to evaluate and optimize utilization of employees in ways that may not be compatible with the employee's needs.
- The roles of leader and coach may be compatible in some cases, but they often require different skills — skills that frequently don't reside in a single person.
- Managers and coaches tend to have (need) longer term stability. Leaders frequently need to change over the course of a project depending on the situation. For instance, early a project, vision may be the key leadership trait needed. Later, a knack for engineering leadership may be needed. Still later during installation, skill for external focus on specific customer situations may be most important. And so on.
- Managers can be effective because they have been assigned authority

and responsibility. Leaders and coaches can be be assigned, but in practice these assignments are only suggestions — team members will decide who they follow and listen to and it may not be the person who has been assigned.

2. The power and problems of process

Effective use of process is the path to high performance in all fields of human endeavor. This process needs to be made explicit — it's impossibly difficult to improve the intangible. Improving a process over time (thus improving skill) requires regular reflection on the process — try process improvements, evaluate how they work, and update the process accordingly.

Thus, technologists must be motivated to develop their processes for creating and delivering innovation. Don't impose a preconceived process — rather, let (and require) the engineers to develop their own process. In particular, I have seen the popular phase gate review system (wherein three or four times over the course of a project a high level management review process must be passed through before the project can continue) cause trouble so often that I think it probably should be avoided. In my experience, phase gate reviews:

- typically waste lots of time on preparation of presentation ma terials
- typically delay the project because it's hard to find a date when all of the mandated (but busy) top managers can attend
- have great potential for doing damage to a project by burdening it with new demands and constraints
- seldom benefit the project because the top managers don't have the necessary insight into the innovation and development process to provide cogent, relevant wisdom

I have few preconceived notions of appropriate process except (a) two principles Kyoshi Uchimaru emphasized of making the process visible and eliminate errors earlier (to do this make processes have many short stages), and (b) reflective practice. I also have no illusion that you can have a fully step-by-step process for doing creative work. However, a good bit of creative work is step-by-step, and you should be able to find step-by-step processes for developing various capabilities that enhance abilities for creative work.

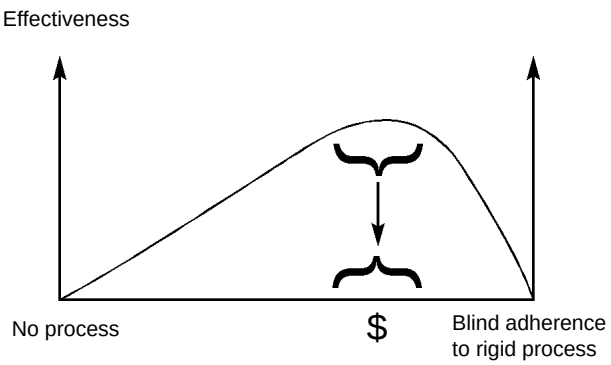

Figure 2: Right Amount of Process

As soon we talk about process, lots of our technologists begin to talk about artistry, inapplicability of process to what they do, what they do is not step-by-step, and so on. I find Figure 2 useful in addressing such comments.

The vertical axis is efficiency and effectiveness. The left end of the horizontal axis is "no process," and the right end is "blind adherence to rigid process." Everyone will agree that the left end of the horizontal axis is not the place to be — without any process we never do the same thing twice in a row, we don't have ways of working together, and so on. The right end of the horizontal axis is where many of our technologists purport to fear we will operate, but this obviously is also not an efficient and effective place to be in many cases. In fact, when they talk about process not applying to them, our technologists are asserting a false dichotomy — that the only alternatives are no process or blind adherence to rigid process. In fact, there the optimum point of operation must be somewhere in between the extremes of the horizontal axis — where we have enough process to become efficient and effective but not so much that we cease to be efficient and effective.

More generally, most demands for no process are not from the best engineers — the people in interval a of Figure 1. The best results tend to come from the people who have developed (perhaps their own unique) processes for doing creating work and who have the greatest ability to focus and keep pushing ahead. Also, most demands for no process are not from the people in interval a of Figure 1. These people are keeping their heads down and trying not to be noticed, lest they be found out.

Most demands for no process — talking about the inapplicability of process to what they do — come from people in the middle group (interval b) of Figure 1. The average employees tend to assert that they are too creative to have process.

On a related point, with some notable exceptions the professional process people don't have the respect of the technologists, and these professionals often don't see the use of process in pragmatic enough terms (someone said, "the disciple is always more zealous than the master"). Rather, I recommend rotating best technologists through 12- to 18-month assignments as process facilitators — it's good for them to take some time off from the creative process to reflect on their processes, they understand the real-life trade-offs and will have to live with what they invent when they rotate back to doing innovation full time, they are credible to the other technologists, and the hypocrisy of management assigning weak performers to such a supposedly key role is avoided.

These technologists temporarily in the process facilitator role are in the best position to teach the relevant theory and tools (and how to put them together in flexible combinations) and to demonstrate the self-analysis of failure (nearly daily reflection on practice) that is critical for learning and improvement.

In particular, a lessons-learned session at the end of a long project is relatively useless. It happens too late to help the project as it is going along, it will surely be at too abstract a level, and — since the project is ending or has ended — team members' thoughts will be moving on to other topics. You need weekly or daily after-action reviews (highly developed self analysis of failure or reflective practice) that address specific issues in ways that can be immediately tested and provide immediate benefit. Kyioshi Uchimaru in his landmark book *TQM for Technical Groups* suggests that the main role of the technical manager should be to teach self-analysis of failure, not unlike how a chess master might teach a student to see how the prior move left a weak position on the chess board.

3. Controlling queues and interactions

In parallel with getting the technologists to develop their creative processes, leaders and managers must support the technologists in avoiding situations where performance is nearly impossible. Thus, there are several *don't* we should keep in mind:

- Don't simultaneously try to maximize function, cost, and time. Give one of these highest priority, do the best you can on another, and let the last one float. To do otherwise is to increase the probability of project failure.
- Don't allow each component of the project to have its own slack time. As shown in Figure 3A, the project will get done some time after the total of the scheduled time for each component of the project and its slack time, since each subproject inevitably will use up all its scheduled time. Better is to allow no slack time for the subproject for any component and to keep all slack time in a central inventory, as shown in Figure 3B1. Then, each subproject will get done no earlier than its scheduled time, but the total may not use all the slack time, as shown in Figure 3B2.

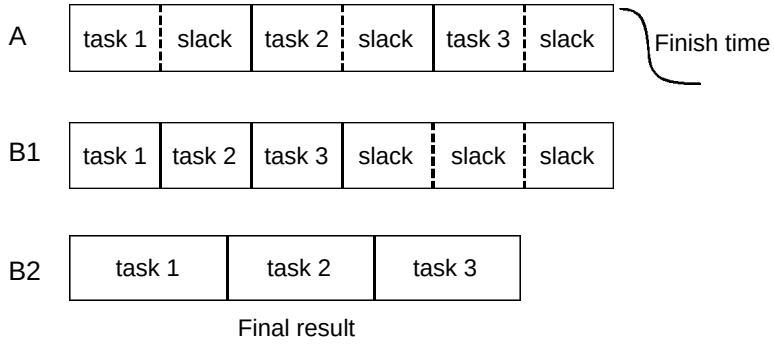

Figure 3: Scheduling Slack Time

- Don't put people in a position where they feel they have to say, "yes." People have to be able to say "no" — to tell you that they can't achieve what you are asking. If people think they must say "yes," then you will not know whether they really mean "yes," mean "maybe," or mean "I can't (or won't) do that and I wish you'd leave my office and stop demonstrating to me that you don't have a clue."
- Don't try to utilize people and capabilities over about 80 percent. Of course, it is not uncommon to try to utilize people in development groups at 150 percent or even more; the basic results of queuing theory guarantee that this will result in projects taking "forever" to get done. When everyone is over scheduled already, trying to get critical resources in a timely manner is impossible when the resources are shared among projects. This leads to the next point.
- Don't time share people: use dedicated teams.

When things haven't been going well and a new manager with a track record of success is brought in, inevitably that manager begins by making the necessary priority and allocation decisions. The choice is made regarding which project will have first priority, which second, and so on. People are dedicated to projects and not over scheduled. Priorities are determined among function, cost, and time. Of course, doing this is hard. Top management has a terrible time making necessary choices, preferring to insist that everything has to be done soon. Yet, time and time again, we see

demonstrated that failing to make the necessary choices almost guarantees little gets done, and making the choices to do first things first and delay others results in things getting done and getting done in a timely way.

In my experience, it is impossible to accurately forecast the total development cost for a number of projects in a given year. Each project takes longer than you think it will (and thus costs more), especially when they are competing for the same resources, as when people are over scheduled or time-shared. A method I have found that does allow accurate forecasting of total development cost in a year is to decide the total development cost that can be afforded and only staff to that level. Then, with that fixed resource, first fully staff the highest priority project. If there is staff left unallocated, fully staff the second highest priority project, and so on. When the first project is done, that staff is available to work on the next highest priority project. In this way, you know the total cost of development, and the projects will get done as fast as they possibly can since they are not competing for resources (and this, in fact, will enable better estimation of project finish times).

I have also found it beneficial to promise only what can be delivered (using the system described in the previous paragraph), and to turn away additional business, regardless of how painful it seems to be to do this. Because customers can better depend on what we promised, in the long run business was better. Surprisingly, I had a hard time convincing my people that this is how I wanted to run things. They were hard to convince, I believe, because it's such an untypical way to run things and because they then had no more excuses for not getting things done on time (they couldn't claim any longer to be over scheduled).

4. Sources of innovation

Technologists are always claiming that customers and users don't know what they need and, therefore, it's a waste of time to ask them: "we need to tell them what we know they need." In fact, customers and users often don't know (or can't say) what they need; and, when we ask them what they want, too often they tell us something which they then don't buy after we have developed it.[5]

However, our technologists are wrong in thinking this means we don't need to ask customers and users what they need:

- If we don't ask customers and users what they need, then they may assume we aren't delivering what they need.
- We may be better able to sell to customers and users what we conclude they need if we can describe it in terms of what they told us.
- If we don't ask customers and users what they need, then we won't have this additional information available to feed into our analysis to help us discover what they really need. If we just deliver what we think customers and users need, we increase our probability of missing the mark.
- To maximize our probability of delivering a successful product, we need to integrate what customers and users say they seem to want with what we think is possible to do that is relevant to their situation.

Thus, we must enable and force technologists to draw on relevant sources of innovation and to make the appropriate tradeoffs. To this end, we need explicit methods.[6]

[5] I've been involved in the wasting of millions of dollars of development building products that meet a supposed customer specification, and the potential customers then don't buy the product. It is too easy for potential customers to say what they need; unless they are paying up front, they may also easily later say (in the immortal words of Emily Litella), "Never mind."

[6] Although I have a favorite method, I won't bias things by citing it here. It is sufficient that you choose an appropriate specific method.

There are a couple of specific points I will make.

First, it is necessary to both listen to what users and customers say and to observe what they do. The latter provides context that helps us understand what the former might mean. To successfully hear what customers and users says and see what they do, MIT professor Eric von Hippel says that it is necessary to have technology people and marketing people jointly visit customers and users. Von Hippel argues that in general marketing people are good at hearing what customers and users have to say, while in general engineers are not good at hearing what they say. On the other hand, von Hippel continues, marketing people tend not to be so good at seeing what customers and users are doing while engineers are good at seeing what they are doing.

Second, it is important to find what von Hippel calls "lead users." Everyone knows the illustration of the life cycle of a product as a bell-shaped curve divided into four regions: early adoptors, early majority, late majority, and laggards. Lead users are a subset of the early adoptors who have such a great need for something the current product technology does not provide, that they have created some sort of work-around of the present technology's limitation. This work-around may be a manual adjunct to the existing technology or a technical modification of the existing technology. In any case, the work-around provides, in effect, a prototype of a modification of the technology. Von Hippel's research shows that in over eighty percent of the cases when a new technology is developed, the innovation does not come from the company that produces the product (even though the company may think it made the innovation); rather, the innovation typically comes from a user (or possibly a distributor) in the form of a work-around, and the company then productizes the innovation.

Because of his focus on lead user's (these users who have created innovative work-arounds), von Hippel is always on the lookout for instances where a company can learn from innovative users. In a video tape by von Hippel, he tells approximately the following story:

> *He was consulting to a company that produced food processing equipment and was visiting a customer of the company with the manager of customer support. This customer was using a bread bagging machine the company produced, i.e., a machine that put loaves of bread in plastic bags and sealed the bags. The machine was rated at 60 loaves a minute, but von Hippel and the customer support manager discovered that the customer had modified the machine so it was processing 180 loaves a minute, although running the machine faster introduced some reliability problems. Von Hippel was excited by what he saw and suggested that the customer support manager should pass this information back to the company.*
>
> *As few months later, von Hippel was again visiting the company, saw the customer support manager again, and asked, "Did you tell the other people in the company about the bread bagging machine that the customer had modified to run three times faster than its rated capacity?" "Yes," the customer support manager answered. "What happened?" asked von Hippel. The support manager answered, "We cancelled their warranty."*

The point of this story is that the company did not have a system in place to gather valuable user data. We must make sure such valuable customer data is gathered and is made available to our technologists.

I'll conclude by noting that more breakthrough comes from long, hard digging than from inspiration, or maybe a better way to put it is that inspiration typically comes as part of long hard digging.

Thank you very much.

Acknowledgments

Of course, in the oral presentation of this material, giving detailed citations was not really appropriate. Therefore, I haven't included them in this text version of my oral presentation. However, I can provide numerous references to various points I mention in this presentation. Contact me via the CQM to request any available citation for a point made in this presentation.