

IT METRICS



HELPING MANAGEMENT MEASURE SOFTWARE & PROCESSES AND THEIR BUSINESS VALUE

The Multiple Dimensions of Metrics: Metrics and the Learning Organization

One might say that organizations learn and grow, stay stagnant, or wither depending on their decisionmaking process and their ability to adapt. In the words of Chris Argyris and Donald Schon of Harvard University and the Massachusetts Institute of Technology, respectively, our organizations “need to adapt to changing environments, draw lessons from past successes and failures, detect and correct errors of the past, anticipate and respond to impending threats, conduct experiments, engage in continuing innovation, and build and realize images of a desirable future.” And, of course, companies must do all this at Internet speed.

The way to achieve a highly functional organization and to determine our responsibility as individuals within the organization is further articulated by researchers Anne Wilson Schaefer and Diane Fassel, who write:

Everyone in this society lives and works in some kind of organization, group, or system. Beginning with the family, moving into school, the workplace, clubs, and civic organizations, most of us spend the majority of our lives within organizations or relating to organizations. In fact, in this society, the person who cannot function organizationally is handicapped.

Thankfully, this is an area in which metrics serve a definite purpose, yielding insight into the inner workings of an organization such as IT. Of course, metrics can be accurate or inaccurate, useful or useless, depending

Continued on page 2.

Meditations on Which Metrics Matter

The Kripalu Center for Yoga and Health in Lenox, Massachusetts, USA, is the largest center for yoga and holistic health in the US, serving people of all backgrounds for more than 20 years. It happens to be right near my home in the Berkshire Mountains, and I was recently there with my wife, Connie, and our two small children for dinner the day before New Year’s Eve. I was trying not to think about IT.

“Who are we looking for?” I asked my wife.

“My friend from New York,” she replied. “She’s an attorney who specializes in contract issues, and she’s here for the weekend. We might find her in the dining hall.”

While we were getting our meal, I came across the executive director of the center, whom I’ve known for years. A former executive with

Continued on page 8.

executive summary

Welcome to *ITMS*. It’s an honor and a pleasure to have been named the successor to Howard Rubin as *ITMS* editor. First, I want to express a sincere appreciation and acknowledgment of Howard’s work. His contributions are innumerable, and they provide a platform we will build on in the future.

To help you maximize the effectiveness of IT in this fast-changing world, I’ll bring you perspectives from the many worlds I’ve “lived” in: first, my views as a practitioner who researched the real-world application of metrics in the early and mid-1980s at *Fortune* 500 companies; second, ideas from my 12 years as a consultant, advising countless teams, both at the project level and (more recently) with senior executives at the organizational level. *ITMS* will tackle these domains and more.

This issue was designed to provide a cohesive picture of upcoming directions for *ITMS*, such as management metrics issues, case studies, benchmark data, metrics theory, and practical guidance on implementing and using measures. This issue sets the stage for those future topics, where we will focus on these issues in a quantitative way — plenty of numbers and charts — with a qualitative approach. In this *ITMS* is a collection of practical “mini-essays” that we will expand on in later issues.

This issue also includes a close look at the tough negotiations and difficult conversations we in IT face every day. Starting on page 11 is Part One of a discussion with Doug Stone and Sheila Heen, researchers at the Harvard Program on Negotiation. Their new book, *Difficult Conversations* (with Bruce Patton), has made a big splash. Stone and Heen offer some ways to deal with those difficult conversations that often result from our measurement efforts. Have you had any difficult conversations about IT in your company lately?

Michael Mah, Editor

february 2000 vol. VI, no. 2

The Multiple Dimensions of Metrics: Metrics and the Learning Organization	1
Meditations on Which Metrics Matter	1
Difficult IT Conversations	11

Continued from page 1.

on the proficiency of the information gathering and its associated analysis. As practitioners, we must constantly assess which measures work and which ones don't.

We must also be aware that there will be times when data tells us things that we do not want to hear. In that regard, it will be hard for organizations to not engage in antilearning defense mechanisms and cover-ups. Like people, organizations can go into denial. As managers, we must lead by our actions, recognizing that the global economy will ensure a competitive pace that will not suffer slow-learning organizations gladly. Industrial Darwinism and the free market will create conditions for a "natural selection," weeding out those organizations that do not learn and grow — both intellectually and emotionally.

At the same time, this is an incredible opportunity for organizations that gather, analyze, and use information to raise their performance to the next level. The proficiency of their acquisition of measures and their ability to learn from these measures will elevate them to higher levels of organizational performance. It will not always be easy. Organizations with no measurement frameworks might even justify their position by saying that the measures are too costly. But I remind you of the familiar saying, "If you think education is expensive, try ignorance."

Living in a Deadline-Driven World: Metrics to Manage Time

During a speech on deadline-driven management at a recent software quality conference, I posed a simple question to the audience of 100 or so attendees: "How many of you feel you have enough time for your families?" Not one hand went up.

Yet in October of 1999, a headline in *Computerworld* claimed "85% of IT Departments Fail to Meet Business Needs."

I wondered how that statistic feels to the legions of people working such long hours that their children feel orphaned. Is it possible that 85% of organizations have overly aggressive demands on IT delivery?

What are these two disparate "metrics" saying? Perhaps the answer to this dilemma is to source more IT work offshore. That way, we can get workaholics in other countries to meet our business needs, while paying a fraction of our own domestic wages. Let *their* children feel like orphans! Better yet, how about only hiring people who don't have children? In an article written some time ago in *Fortune* magazine, one major executive was quoted as saying, "We have to either meet our deadlines or kill our kids." Ugh!

I followed my question to conference members with two additional inquiries. First I asked, "How many of you, when faced with an upcoming project, are given the deadline first?" Nearly every hand went up. Next I asked, "How many of you have at least half of the system requirements well defined by the user, the client, the marketing department, or whomever, at the time you need to start building it?" No hands went up. I said, "Okay, how about a third of the requirements? One-fourth?" Three hands went up.

Thus are spawned Ed Yourdon's so-called death-march projects: impossible deadlines, impossible stress levels, impossible working conditions, and ever-growing or changing scope. In some cases, the scope is unknown: "We don't know what we'd like you to build, but whatever it turns out to be, we'd like you to build it fast." Which brings me to the fact that, for most organizations, metrics must serve to help managers manage their deadlines. This may mean using metrics to estimate backwards. That is, given our IT capacity (capability, productivity, etc.), a given deadline, and a team of this many people, how much functionality can we build? (More on this in "Metrics for Software Estimation," below.)

Editorial Office: Clocktower Business Park, 75 South Church Street, Suite 600, Pittsfield, MA 01201. Tel: +1 413 499 0988; Fax: +1 413 447 7322; E-mail: michaelm@qsm.com.

Circulation Office: *IT Metrics Strategies* is published 12 times a year by Cutter Information Corp., 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA. For information, contact Megan Niels, Tel: +1 781 641 5118 or, within North America, +1 800 964 5118, Fax: +1 781 648 1950 or, within North America, +1 800 888 1816, E-mail: info@cutter.com, Web site: www.cutter.com/consortium/.

Editor: Michael Mah. Publisher: Karen Fine Coburn. Group Publisher: Anne Farbman, Tel: +1 781 641 5101, E-mail: afarbman@cutter.com. Production Editor: Lori Goldstein, +1 781 641 5112. Subscriptions: \$485 per year; \$545 outside North America. ©2000 by Cutter Information Corp. ISSN 1080-8647. All rights reserved. Unauthorized reproduction in any form, including photocopying, faxing, and image scanning, is against the law. Reprints, bulk purchases, past issues, and multiple subscription and site license rates are available on request.

This is not to say that none of you out there, upon looking at the project you're about to start, know what you are tasked to do. Some teams really do know their requirements. The requirements are stable, well-defined, documented, and within the realm of the technology. Teams are able to quantify the size of what needs to be built or delivered. They can map this information into estimation tools to forecast the required schedule, effort, and staffing.

By the way, there is no single answer to the tricky problem of deadlines, requirements, and staffing. There are many possible outcomes, depending on the environmental conditions and makeup of the team, as we discuss in "Difficult IT Conversations" on page 11. But we certainly should consider ways to employ metrics and estimation methods to quantify the potential expectation gaps we need to negotiate.

Project Management for Adults: Metrics to Manage Risk

A discussion about deadlines brings us directly to the subject of risk management. The IT field handles risk management more poorly than any other field I can think of. We seem to have a learning barrier rooted in messages about risk that sounds like this (thanks to Tom DeMarco and Tim Lister for the following snippet):

"We are a proud, cutting-edge, aggressive, risk-taking organization. We have an IT cowboy/cowgirl culture. We are a can-do company.... We encourage all of you fearless people to be bold and to take risks! But you'd better not screw up!"

We do ourselves a serious disservice when we deliver these kinds of double messages. One result is that we take bad risks. The oft-quoted Standish Group Chaos Report states: "Out of the US \$250 billion spent on IT projects in a given year, 31% are canceled, representing \$81 billion in losses, with 52.7% overrunning by more than 189%. Only 16% are on time and under budget, but they got there by delivering only 42% of the originally desired functionality."

Grim news. Not a lot of good risk management going on here. If this is true, or close to true, it says that we are a pretty dysfunctional bunch. We chronically overpromise.

Worse yet, there are a slew of management types, clients, or end users who are more than happy to suck a hapless IT project team into an overpromise condition. Both promoters and promisees participate in a "systems dynamic" of mutual contribution that perpetuates a self-sealing cycle and guarantees a wave of future disputes. The we-needed-it-yesterday culture drives this behavior and locks us into this dynamic.

More insights about root cause come from such things as Chris Pickering's *1998 Survey of Advanced Technology* (www.cutter.com/itgroup/reports/sat98.html), in which we discover that 75% of organizations surveyed have backlogs running from seven months to more than two years. We are addicted to our high technology and can't get enough of it. Something has to break the cycle of poorly managed projects. I suggest that one angle might be ditching traditional software estimation (see discussion below).

Metrics for Software Estimation

When I wrote an article entitled "Just Say No to Software Estimation" (Cutter Consortium *Business-IT Alignment E-Mail Advisor*, 20 October 1999), at least one software IT manager asked if I might be available to marry his daughter (actually, I've been happily taken for more than 15 years). Apparently, this article struck a nerve. It spoke to the fact that metrics and estimation have failed for the majority of IT organizations. No one really estimates; they just fudge the numbers to get an answer that miraculously aligns with the dictated deadline. What a startling coincidence! Here's what I said to cause this wave of gratitude:

Forget about "traditional" software estimation. Don't bother. After all, how could anyone possibly estimate the schedule to build something they're not sure of? Estimate the schedule? Who cares? I know what the estimate is. The estimate is the deadline, and it's on Internet time!

Abandon the notion of first defining the requirements, then estimating the size of new and changed modules, objects, function points, or code, and finally translating that into forecasts for time, effort, and the required staff. To do that, we need to know the requirements — but we

don't. And whatever the requirements are initially, they're going to grow in scope anyway. Estimate the staff required? Forget it; it won't matter. Here's the answer on the estimate of the staff size: it's the 20 people we have available. Now, get to work and start coding. Code what? We're not sure yet.

So, what's an overworked project team leader to do? (Run, I say. Run for your life, as fast as you can!) No, seriously, here's what we know: we know the deadline and we know how many people we have on hand. We have no choice but to work the problem backward. "How much of this kind of software can we build, with this many people, in this little time?" When we come up with this number, we grab onto it, hold on tight, and refuse to give in when we are asked to deliver one line of code more than this limit. If the size translates to, say, 28 modules, do whatever you have to do to hold your ground and not agree to one module more. It's likely that at least one-third of those will change after you've built the first version anyway, only to be thrown out and reworked. So you'll need the overtime to redo that first version of the code.

There are three major implications to this excerpt: First, in order to backwards-estimate the amount of possible functionality, you need some knowledge of software productivity. Second, the term "amount of functionality" begs for a definition. It can be, but doesn't have to be, function points. (Did I just shoot a sacred cow for some of you out there?) Third, negotiation. Tom DeMarco often says that software people need to learn assertiveness training. You betcha — but, Tom, they run the risk of asserting themselves right into the unemployment line!

These ideas come back to the earlier section on generating risk-management scenarios with reliable numbers to manage the deadline. If we accept this as a practical purpose for metrics and project estimation, it begs the question, "What is our productivity?"

Many IT managers cringe at this question, and all kinds of political issues are involved with this subject. But we need to do it better than we have in the past. We need to not only "reverse estimate" and have a

reasonable idea of what we can build in a given time frame, but also better understand and improve the IT world in which we live. Our appetite and demand for technology in the current economy exceeds our capacity. We can't build the stuff that people want fast enough, and, moreover, we have a shortage of people. So if we don't want our CIO to finally get fed up and outsource everybody, we better find a way, as Emeril Lagasse might say, to kick our productivity up a notch.

IT Productivity and Darwin's Theory of Evolution

What are the purposes of productivity metrics? That is, what do I do with the information that my IT department can produce x amount of code or function points per person-month of effort? Or that on a J.D. Power, IT Consumer Reports, or Putnam Process Productivity Index scale, my average productivity is a 17.5, and a normal range is 15 to 19. How do I use numbers like these?

Let's start with Charles Darwin's theory of evolution. No, he didn't include IT productivity as part of his theory on natural selection. The idea comes from reading a philosophical musing by another writer, one in the audio engineering field, which I'm adapting to IT productivity.

Argyris and Schon, in their preface to *Organizational Learning II, Theory, Method and Practice*, have echoes of Darwinian theory when they describe the global economy as a place where "the pace of competition is savage and swift, where the most apparently solid companies and, indeed, whole industries, can disappear or suffer decline because they fail to detect and respond to early warning signals for rapid change."

In this regard, the capabilities provided by the enabling effect of IT can make or break a company. How might one go about achieving the highest possible levels of IT productivity and effectiveness? In more than a decade of helping companies benchmark their IT productivity to establish productivity baselines, my colleagues and I have observed that every productivity benchmark we have conducted has a distribution of values, from low to high. Within this spectrum, if you dig into the underlying factors, are characteristics

both good and bad that explain the “whys.” This is known as causality. These factors provide positive and negative impacts to a project. A project reflects good performance when the positives outweigh the negatives.

I also find that, although we often refer to the term “productivity” as a homogeneous value, it’s never one number. For example, organizations experience different values for new application development and major releases, maintenance and bug fixes, package modification, and package integration.

Left on its own, the IT environment (and the marketplace) will tend to encourage the survival of the processes that lead to good outcomes and will weed out the processes that lead to failed projects. Why leave things to follow their own pace? Metrics practitioners and management can combine their talents to accelerate that process to their advantage.

Evolution looks at the example that, in nature, longer-necked giraffes reached higher branches of trees to feed on leaves that shorter-necked giraffes could not. Thus, they were more nourished, and, as a result of this strength, yielded healthier offspring who also had long necks. Over time, herds comprising long-necked giraffes prospered more than herds with short-necked giraffes. The same principle goes for software IT practices.

“Mah’s corollary” on the theory of survival of the fittest says that if we accelerate “IT natural selection” using metrics, we can find the long-necked giraffes in the IT world and help them to multiply. This idea involves uncovering those factors that resulted in projects with successful outcomes and trying to replicate those conditions as much as possible in the future, helping them to proliferate. Think of it as facilitated evolution.

The factors we’re looking for in this evolution are the tools, methods, processes, and teams that combined and resulted in outcomes that companies were proud of — outcomes that at first might seem ordinary. But those very outcomes might have been made possible by the heroic efforts of teams that overcame high adversity. In some cases, it can be a remarkable achievement that a project is completed at all. The role of productivity benchmarking could be to find and nurture your long-necked giraffes so that your herd will grow and prosper.

The Evolution of Metrics Technology

Software Management Metrics: Data Anytime, Anywhere

At an exciting conference in the fall of 1999 in Boston, Massachusetts, USA, by Safeguard Scientifics, one of the rapidly emerging stars in the world of e-commerce, Pete Musser (CEO) and Harry Walleasa (COO) described a world where, thanks to the Internet, we could have access to information anytime, anywhere, on any device. Subsequent presentations by folks, including Dr. Nicholas Negroponte of the MIT Media Lab, Compaq CEO Michael Cappellas, and spokespersons from Safeguard and its member companies, offered different elements of this vision.

During a break, I was chatting with Doug Mellinger, CEO of PRT Group, Inc., partner of Opus 360, and other Internet firms. He pulled out his new toy, a Palm VII. Mellinger proceeded to look up local theaters and movie times for films playing in Boston, recommend restaurants, and check his e-mail — all through a wireless RF connection made possible by flipping up the unit’s antenna. “Isn’t this cool?” he said, with a kid-like grin.

We, of course, talked about some other uses for the gizmo. Turning to Rita Terdiman, a well-respected industry analyst, I said, “Rita, imagine that you’re an IT executive or project team leader, and you want to know the progress of your top 10 projects, all with looming deadlines that keep you awake at night. Wouldn’t it be cool to flip up your Palm VII antenna, use the little pen to pull up the latest data from the ABC ERP Phase 2 project or whatever, and see whether it’s in a red light, yellow light, or green light condition as it enters system testing?”

“Yeah,” Rita said. “That would be really cool.” Do we IT folks live in a strange world, or what?

Visual Explanations: Metrics as Images

“Steve Jobs has two jobs,” a recent *Time* magazine article pointed out: he is both Pixar chairman and the interim Apple Computer CEO for life. The man who created both the Mac and the *Toy Story* duo of Woody and Buzz Lightyear personifies the intersection of art and science.

Remember where we started? My first experience with a computer was in the 1970s, programming a DEC PDP 11 with paper tape in the Tufts University electrical engineering lab, doing circuit analysis simulations.

The next cool thing was the online editor and compiler I could access from my fraternity room. I could take my telephone handset and put it into a primitive modem with rubber cups that held both ends. I could tap away at my character-cell screen and not have to deal with tape or punch cards at all. Thanks to the brilliant people who pioneered the visual interface at Xerox Parc, Steve Jobs, and others who followed, my children will never need to know the limits of a character-cell terminal. But they still get to listen to my vinyl records, which they call “big black CDs.”

Someday, sooner than you think, our current management and measurement frameworks for IT will look like character-cell terminals. (Surprisingly, not too long ago, a well-known telecommunications company showed me a metrics database repository it had built over the last year. It was character-cell based. Talk about nostalgia.)

If we follow the visionary ideas inspired by people like Dr. Edward Tufte of Yale University, we’ll have visual explanations for all that we live and work for in the world of software. This technology will cut through the muck of numeric tables and arcane metrics ratios of productivity borne of a factory and manufacturing mindset. In an information age, replete with new software and IT products that are the result of a wave of research and development thinking, production-type ratios don’t work very well.

Pictures derived from the numbers, revealing multiple dimensions in high-resolution displays, will replace the clunky tables of numbers that obfuscate truth. These images will provide high-bandwidth flows of IT metrics information, answering questions such as “Compared to what?” and “Why?” to show how our efforts stack up against a frame of reference.

The images will be dynamic rather than static. For example, if a project deadline is compressed, what happens to the risk with regard to defects that might occur in the first month a system is deployed? “Squeezing”

something on one end will actually show the “bulge” effect on the other.

Then we will take this information with us wherever we are by flipping up the antenna on a custom-tuned palm computer. Visual information about your IT projects and departments, in motion, anytime, anywhere, on any device. Stay tuned to *ITMS* for more ideas about this exciting realm.

Metrics and Self-Sabotage Inhibitors to Organizational Learning

Sometimes, I admit, I am late for appointments. On a singular level, if I am late for one appointment, I might look at what happened and try to understand the reason why I was late and correct it in the future. That’s a simple example of first-order, or single-loop, learning, as termed by researchers Argyris and Schon.

But what if I were chronically late? Not just for business appointments but also for family obligations. Then I might have to wonder what’s going on — there’s an underlying pattern that’s been revealed. This is a second-order issue, one that might be related to my values, work habits, or scheduling tendencies.

The second-order issues could be that I am work-obsessed or that I take on too much and overpromise within a given time frame. I might be acting like a workaholic, putting out fires one after another, resulting in my being late too often. Part of this could be that I don’t get enough sleep because I’ve been answering e-mail until 2 am, and I might be too tired the next day to be productive.

There is an opportunity to tackle the second-order learning (also called double-loop learning) in the challenge to correct the first-order problem, i.e., the initial, singular lateness.

It’s the same with teams. Suppose your IT department has just experienced the tribulations involved with a late-running project. When you look closer, you realize this happens all the time. The last time a project finished on time was when disco was in. You have some tough second-order issues to face up to. It could be that you don’t estimate projects well, that you don’t keep good records, and that you fatally overpromise projects at Internet-speed deadlines. It could be that you accept a deadline and then never say “no” to the ever-growing wish list of

requirements. Before you know it, the promise to build 42 modules grew to 68, and the deadline is the same. "We'll just work a little more overtime," you say.

An inability to solve chronic second-order issues might be the result of inhibitory loops that make change impossible. There could be denial of the problem, the blaming of others, and an unwillingness to own up to dysfunctional organizational patterns.

In these cases, antilearning and defensive reasoning mechanisms may even suppress the findings produced by good metrics programs. The numbers could be accurate, legitimate, and so valuable that they create opportunity for entirely new outcomes. But if you have an antilearning culture that has an embedded dysfunction, people may deny the numbers or manipulate them to reflect a different conclusion. DeMarco refers to this as "Limbaughing the data." He defines this as choosing selectively from a body of data those items that confirm a desired result, and never mentioning any that might be construed to confirm the opposite.

In many cases where a company has failed at getting measures in place to make decisions, a common underlying theme is, "We're too busy putting out fires around here to do that stuff. Hopefully, metrics and process will turn out to be another management fad. If I resist it long enough, maybe they'll leave me alone, because I have real work to do and am so stressed out from the deadlines they ram down my throat around this place."

The truth is that many of us are caught in this destructive rut. One insightful writer once said, "Since most of us are [too busy] putting out fires, we can't stop. There's one emergency after another... Putting out fires takes an enormous toll on us. Stopping and pondering how the fires started is important. Since we are pyromaniacs — in truth, our behaviors start most of the fires — we need to begin thinking about how to stop ourselves before we self-ignite."

It's imperative that metrics be employed in ways that allow an organization to understand what is behind overrun and error statistics and to better quantify its delivery capability to support strategic planning. Plots, charts, and graphs can visually portray the demands on IT, as well as its capacity.

Closing the gap requires conscious and deliberate use of IT metrics.

This is the challenge for the learning organization in an Internet-speed economy — to get out of the chaos and find out what it takes to operate at a higher level.

Metrics and Outsourcing

If an organization kicks into antilearning defensive routines in response to its own warning signals, it may (in the short term) escape unpleasant feelings. In the long term, what do you suppose will happen to an IT department that is perceived as being slow to change, that lacks the ability to estimate a project, and that has no metrics to demonstrate its capability?

Roger Pressman once said, "If you don't have evidence that you can build software better, faster, and cheaper, then you are vulnerable to someone who does." It is therefore in an IT department's interest to begin acting like an entrepreneurial enterprise. IT's days of not having to compete are over. The competition is out there, with slick marketing brochures and golfing connections to your upper management. They probably have the metrics to make a business case to your executives for outsourcing. They could be the long-necked giraffe who will eat your lunch and leave you chewing on bare twigs.

Organizational learning has become an issue of survival, and outsourcing is here to stay. IDC's *US and Worldwide Outsourcing Markets and Trends* (1998-2003) report states that "worldwide outsourcing services are fast on their way to blowing by the US \$100 billion marker. In 1998, spending topped \$99 billion. By 2003, it will explode to more than \$151 billion. The majority of the spending was in the US, which accounted for 52% of the total at \$51.5 billion, expected to rise to \$81 billion by 2003."

IT organizations will have to learn how to make a business case for IT being a core competence, if that is indeed the case. If you don't, there will be pressure to consider outsourcing. This may or may not be a good thing — there are outsourcing success stories and there are outsourcing nightmares. There are also success stories that keep IT inhouse as a core competency to enable the company to become a market leader.

In some cases, outsourcing is the best thing to bring additional capacity to an IT department experiencing heavy demand. It can provide sorely needed help by furnishing skill mixes and staffing that an organization cannot easily produce internally, enabling internal IT to focus on what it does best: applications that create new markets and opportunities for the company.

In other cases, IT is a core competency, even though it may not be apparent to the consumer. Consider the position of Fred Smith, chairman and CEO of FedEx, which has not outsourced much of its IT thus far. Smith once described the company's computer systems as being more important than its planes. He marveled at the work done by IT designers. "You're talking about people who are worth a fortune," he said.

The workable sourcing solution for your organization will be a function of how well you align IT within your company's culture. One certainty is that metrics will serve a vital role if you want to keep IT inhouse. It's also vital if your company decides to outsource, since contract money is on the table. Any dispute will have the potential to become a legal one.

Ultimately, the solution that fits your company depends on the internal culture and the manner in which you manage IT as a portfolio asset. Portfolio managers make the best decisions about what "trades" to make when

they are well informed and have the numbers on which to base their decisions.

Some Hope for the Future

There is great opportunity in this arena for all of us to have better information at our disposal so that we may work in a more fluid and cohesive manner. Metrics help us think and understand. In the words of Tufte, "Assessments of change, dynamics, and cause and effect are at the heart of thinking and explanation. To understand is to know what cause provokes what effect, by what means, at what rate."

However, even if we had the best, most reliable numbers we could possibly imagine, we still have to relate to each other as individuals, team members, departments, and companies. The technology field is replete with difficult conversations and tough negotiations. The pressures that exist in our field are enormous, and demands are high — some might call it a crisis.

But in the midst of the "crisis" are also incredible opportunities that await those of us who strive to use metrics to help our organizations learn and create better interpersonal, intergroup, interdepartmental, and intercompany relationships. The outcomes that emerge from these endeavors will involve not only how we solve the substantive issues, but also the interpersonal dynamics that are set into motion between people and groups.

Continued from page 1.

Arthur D. Little, Rick now oversees all of the operational and business issues of the facility. When I saw him, he was wearing an apron and loading plates for the guests as they lined up for their vegetarian meal.

I wondered how many CIOs of IT organizations work the trenches to get a feel for the pulse of their organizations? There I was, thinking about IT again.

"All set for Y2K?" I asked Rick. I imagined what the hundreds of people there would do the following night if anything happened to the power systems.

"You bet. We've planned for this since I got here in February. Got backup generators and propane. We're ready for anything."

I'm sure that being ready contributed to the fact that Kripalu was completely booked that weekend. Being year-2000-ready was business smart, not only for the millennium rollover, but for any outage, such as ice storms or the like. I wondered if being ready for year 2000 was what helped sell out the rooms for the weekend: yoga and meditation would be uninterrupted here.

We finally found my wife's friend. She had arrived earlier that day and was unwinding. After a few minutes of getting acquainted, I casually asked what kind of work she did in New York City.

"I work for a major IT consulting firm, and I specialize in contracts for IT outsourcing. Are you familiar with IT?" she asked.

"Uh, a little," I replied.

After discovering the similar professional worlds we lived in, my wife's friend confided that the field of IT measurement was too confusing for many people, even in her firm, which is a major provider of IT applications outsourcing! I agreed. "People make it too complicated for themselves," I added.

Later that evening, I was still thinking about our conversation. Here was a highly intelligent attorney, originally from Europe, now practicing in the US, who speaks multiple languages and is obviously very open minded, yet finds IT measurement arcane and confusing. I wondered what our collective responsibility as IT professionals might be to the situation, and how to best contribute to a solution.

Becoming a Better Listener

Many of us in the IT metrics field have observed a paradox in the industry. There is a great deal of interest in getting information about IT in the form of measures, but there is a high failure rate in IT metrics programs. Like Howard Rubin, I've asked audiences questions such as "How many of you have successful IT measurement frameworks, the results of which are used to help make strategic decisions?"

In almost all cases, very few hands, if any, are raised.

As I dig deeper for answers, I find that a common reason for this is that people, even those who have spent years in IT, don't understand how to structure IT measurement to help them make their jobs easier, not harder. Yet we find issues like project estimation and productivity baselines crucial to our daily decisionmaking. Measurement plays an undeniably important role.

I suspect that one reason many metrics programs fail is a lack of trust. Often, senior management will mandate metrics implementation without framing the problem within the context of a larger purpose that needs to be thoughtfully communicated throughout IT.

Sometimes, the problem is that members of senior management are unschooled in what measures matter to the organization, and they do not ask for the right information, thus eliminating the motivation to gather

necessary data. Metrics programs are initiated and then dismantled from lack of use.

My advice to managers trying to get a metrics program off the ground is to listen. Make inquiries. Walk the halls randomly and feel the pulse of the IT organization. Ask line-level managers, IT client organizations, and developers what matters to them. Imagine yourself in their cubicles, trying to make that "Internet-speed" deadline, and ask them what they think the goals of measures would be for them. If we are going to follow programs like the goal/question/metric approach advocated by many in the field of metrics research, we need to get the goals right first. And we need to uncover the peopleware impediments to achieving them.

To do that, managers might have to grab an apron and load a few dishes.

Measuring Outcome Versus Output

At the Cutter Consortium *Summit '99* conference¹ held in the spring of 1999 in Boston, Massachusetts, one of the panelists in the outsourcing session described metrics as needing to focus on the measurement of *outcomes*, versus just measurement of *output*. This is a worthy point. For example, let's say you have a project that results in a dramatic increase in revenue to the company's balance sheet and pays for itself and all the IT staff on the project 10 times over, what would that make you want to do? Get more IT staff doing projects just like that, of course.

Now suppose that same project team missed its initial deadline or that the project, because of its complexity, exhibited lower productivity than the industry average. Would anyone care, or would these measures be interpreted differently in the context of the larger purpose? A proper interpretation of any metrics should occur within the framework of overall *benefit* and competitive advantage.

Along these lines, a recent Cutter Consortium survey revealed that 60% of the respondents who aligned IT with formal business models and processes expected to achieve a competitive advantage by doing so. Also according to the survey, 58% of companies have management that views IT as a competitive resource.

¹For more information on *Summit 2000*, visit www.cutter.com/summit/.

This perspective might realign the IT measurement goals of many senior managers. Some who may have taken a limited view of IT metrics as a means to drive down costs might gain a new perspective. In nearly all organizations where I observe a sense of mistrust regarding metrics, an underlying component is fear about how the measures might be used to execute a cost-reduction strategy. Although there is nothing evil about reducing costs per se, it must be done within the context of the bigger picture.

If this is an underlying concern within your IT organization, you won't understand the implications of it unless you actively engage in a dialog to understand what makes IT metrics tick (or not tick) in your company. Not doing so may leave unresolved issues that will contribute to your metrics program being undone after working hard to put one in place.

Other Metrics that Matter

In addition to having a framework that measures outcome, it's essential for IT managers to understand the measures of IT output. What is the capacity of your IT organization? How does it compare to others? How productive is it versus others in our specific industry? The answers to these questions will help you appreciate the functional throughput of your IT organization.

Establishing a productivity baseline is the first step. It answers the question "What is our capability?" When you get these measures in place and have a frame of reference, you can answer the question, "Compared to what?" What are the measures that matter here? A few of those measures are described below.

Some time ago, research by people like Larry Putnam yielded what came to be described by the Carnegie Mellon

The Four Core Metrics

Here are useful definitions of metrics described as the "minimum data set" by the Carnegie Mellon Software Engineering Institute:

- **Size:** What has been built, or what must be built, as countable entities.
 - Moderate resolution: number of programs, modules, processes, function points, and lines of code. All represent the building blocks of a system.
 - Higher resolution: functionality, including new code (or programs, function points, modules), changed code, and reused code without changes. In these cases, size should be noncomment, nonblank source statements (as opposed to physical lines, which include comments and blanks). Size profiles of simple, moderate, and complex programs, entities (for example, source lines of code [SLOC] per program), or SLOC per function point by development language.
- **Time:** Elapsed time in months for each of the major development phases.
 - Moderate resolution: elapsed time for the project from the start through deployment.
 - Higher resolution: elapsed time for each major development phase; for example, feasibility study phase, functional design phase, main build phase (detailed design, code, test), and maintenance phase. Relative proportions of these values as a function of the main build time;

for example, five months functional design, ten months main build, two months overlap would lead to functional design time equaling 50% of the time spent in main build, with a 40% overlap. Extra credit: milestones throughout the design, code, build, and test phases as a percentage of the overall main build schedule (for future estimates).

- **Effort:** Full-time equivalent person-months of effort expended during the project.
 - Moderate resolution: person-months of effort expended throughout all phases.
 - Higher resolution: person-months of effort broken down by development phase or by development labor category. Relative proportions of these values as a function of the main build effort. Amount of overtime spent or overtime by development phase or calendar time.
- **Defects:** An error in analysis, design, or coding that affects required performance.
 - Moderate resolution: number of defects found during testing.
 - Higher resolution: number of defects by severity category (major, moderate, and minor) found throughout system integration testing. Rate of discovery over time. Defects reported within first 30, 60, or 90 days of operational service. Extra credit: defects found throughout early phases of unit coding and testing, as well as defects found during design and code walkthroughs.

Software Engineering Institute as the “minimum data set,” otherwise known as the four core metrics: size, time, effort, and defects. (See sidebar, “Four Core Metrics” on previous page.)

You can look at any completed project in various ways: for example, its duration (months), the amount of work effort expended (person-months), or the number of people working on the project. Their hard work results in a system that represents a certain amount of functionality (size), at a certain level of quality (defects). Anyone embarking on a measurement program should start with these four core metrics. They will tell you what happened on past projects and help you gauge the “functional throughput” recently demonstrated by your organization and its management processes.

Why these four? Often, projects are managed by just two metrics: project milestones and effort (proportional to cost). The other two, size and defects, are often neglected. But size and defect metrics are critical, as they represent what has been built (or will be built, for new projects) and the quality of the end result.

It would also help to add one or two more metrics for past projects. One is the amount of rework; another is the degree of software reuse. For additional background on these metrics and their use, see the useful article “The SEI Core Measures” by Anita Carleton, Robert Park, and Wolfhart Goethert (see “Resources and References” on page 16).

Once you acquire these measures on a sample of your IT projects, you will then have the basis for constructing your own productivity baseline, or benchmark. From here, several opportunities emerge that will be of strategic value to your teams. The benchmark will enable an understanding of what is working and what is not. Immediate opportunities for process and productivity improvement will be possible. Better software estimation will result, and opportunities for more productive negotiations between IT and client organizations will be possible.

But getting in the way of that is the fact that most people misuse or misunderstand how to make the resultant information serve these purposes. In short, people make it harder than it has to be. To avoid doing this to

yourself, one of the most important issues is the data you’re looking to collect — start with the minimum data set necessary. In other words, don’t send your teams chasing 400 metrics on 40 sheets of data collection paper; look for the metrics that matter.

In future issues of *ITMS*, we’ll cover how to supplement the core measures with information that links the findings to the issues being faced by IT, and what to do about it. In the next issue, we’ll describe how you can make use of the four core metrics (and other measures) to build your own productivity baselines without needing a consultant.

Difficult IT Conversations

Scenarios played out in research on negotiation theory and practice result in the following: an identical situation, given to 20 or more pairs of negotiators to resolve, often results in no two outcomes being alike. The moral of the story seems to be that any outcome to reach an agreement between two parties has as much to do with the interaction between the parties themselves (and in some cases more so) than the substantive issues that are “on the table.”

Clearly, the field of metrics needs to better understand the field of negotiation.

The IT world is replete with disputes, whether it’s dealing with clients who keep changing their minds, addressing unrealistic deadlines and ever-expanding project scope, firing an employee, responding to an angry phone call about another missed deadline for an already late project, or dealing with a productivity shortfall with an outsource supplier or contractor.

More than a year ago, after a particularly harrowing day helping clients in painful disputes, I called Tim Lister as part of my research into the field of study on dispute and conflict resolution. I thought of Tim because of his work in the field of legal arbitration, and he suggested that I look into the research at Harvard University. This led me to the Program on Negotiation (PON) at Harvard Law School. PON is an interuniversity consortium comprising Harvard, the Massachusetts Institute of Technology (MIT), and Tufts University. The latest research on this subject was being taught at

Harvard Law School by the PON faculty. Their mission is to improve the theory and practice of negotiation and dispute resolution to change the way people, organizations, and nations resolve conflicts. Participants came from around the world to study and put these theories into practice, whether it be in Boston, Atlanta, Seoul, Bosnia, or Rwanda. My objective was to deal with what I see as a tidal wave of disputes as the technology field thunders into the new millennium at its frenetic pace. For others, such as the former United Nations negotiator who was dispatched to conflicts in hot spots around the world, it was about understanding what might be done differently between governments and rebel leaders.

As part of the program, I met researchers Sheila Heen, Doug Stone, and Bruce Patton, the authors of *Difficult Conversations: How to Discuss What Matters Most*. It's the latest in a series that began with the international bestseller that transformed the way we negotiate, *Getting to Yes*. The authors provide a user-friendly framework for understanding what makes difficult conversations so hard, the mistakes we commonly make, and a step-by-step approach to handling them with more confidence and better results.

A great thing, I thought, especially since nearly all IT disputes I've observed and tried to help solve have experienced a virtual breakdown in the relationships between the parties. Unmet expectations and angry disagreements between individuals escalate and create "camps," sometimes scaling up to larger disputes between groups, such as departments and business units and (in outsourcing or subcontracting) between companies.

The book's fundamental premise is that every difficult conversation is actually three conversations happening at once around a dispute. First, and most obviously, the authors say is a conversation about "What Happened," complete with competing versions of events, accusations about others' motivations, and enough blame to go around. Describing why folks have such different stories about what's happened and people's intentions uncovers some simple, but powerful, ways to have a more productive conversation about each side's perceptions of events. The authors also guide us toward a

less blaming and more problem-solving approach to understanding what has gone wrong. Accepting that usually everyone has contributed to the current situation in some way orients the conversation toward fixing things going forward, rather than defending past mistakes. This is a crucial stance to adopt in a multimillion dollar IT project that is already late and over budget or that is performing poorly in the field.

Often, we can't successfully address the substantive problems unless we manage the other two pieces of the puzzle: what the authors call the "Feelings Conversation" and the "Identity Conversation." The authors point out that once things get difficult, feelings are usually at the heart of the problem between people or members of a team.

Feeling unappreciated, taken advantage of, misunderstood, frustrated, confused, and perhaps even betrayed — these are the emotional arenas of today's Internet-paced IT world, along with insufficient time to spend processing such emotions. *Difficult Conversations* offers a smart and efficient way to understand your own feelings, negotiate with them, and use them to get to the heart of the matter, professionally and purposively.

Finally, the anxiety involved in many of these conversations can be traced to the "Identity Conversation," or what the situation seems to be saying about us. Telling the boss or the client "no" may be particularly hard if you think of yourself as someone who gets the job done, even if it takes heroic effort. Firing someone or being straight about critical feedback is difficult if you hate to hurt other people's feelings. We may experience our own personal "identity quake" if our self-image as a competent, caring, ethical, responsible, or superstar self is threatened by the conversation.

The stories and examples throughout the book will likely make you laugh. The authors draw on a mix of negotiation theory, psychology, organizational behavior, and a vast well of people with a wide range of challenges.

The authors also teach an executive crash course at Harvard Law School entitled, "The Difficult Business Conversation," which is often packed with hundreds of executives

who have traveled from around the world to attend this two-day event.

I spoke with Heen and Stone, two of the three authors, about dealing with angry clients, nonperforming employees, the challenge of personal competence, the pursuit of happiness, profit, and lifelong learning. The following are excerpts from what turned out to be a great conversation about *Difficult Conversations*. This is the first of a multi-part series on the subject.

A Conversation with the Authors of *Difficult Conversations: Part One*

Michael Mah: In the realm of performance measurement for technology firms, a common response to metrics, or measurements, is to sugarcoat findings and keep senior management from potentially judging an already overworked department harshly. How can the thinking in *Difficult Conversations* help the team and the boss to uncover things that seem “undiscussable”?

Doug Stone: A common problem throughout corporate America. At the most basic level, what’s happening is that the senior managers are conflating “evaluation” with “problem solving.”

Sheila Heen: Let’s give an example. If I’m cooking dinner, and there are 10 people at a dinner party waiting for the dinner, my spouse might come into the kitchen and say to me, “Hey, this isn’t going well. You’re not a very good cook.” That’s evaluation. It judges my performance — and probably accurately! But it doesn’t help me to get the meal prepared now. There are still 10 people waiting to eat, and so far all my spouse has done is discourage me.

DS: The key here is your purpose. People get into trouble when they imagine their purpose to be evaluation, when in fact what’s called for is hard-nosed problem solving: “Here’s where the project is, here’s where it needs to be, how can we get there?” That’s a very different question than, “Here’s where the project is. Who should be penalized because it’s not where it needs to be?” Evaluation has a time and place, but it’s actually counterproductive if you are trying to understand why the project is in trouble and how to get things straightened out.

MM: So managers sugarcoat the results because they don’t want to hurt morale?

SH: Nobody wants to discourage people who are working hard — even when what they’re doing isn’t working. On the receiving end, nobody wants to be blamed for the problem. Everyone knows that eventually, when things go wrong, someone is going to be blamed.

DS: See, this is the tension. Blame and evaluation inhibit problem solving. I guess that’s the punch line — it’s hard to do both at the same time. If I’m your manager, and I want to have an honest discussion of where the project is and where it needs to be, and you know that a not-so-subtext of this discussion is that if it’s not where it needs to be, you will be punished, then you’re not going to be open and honest with me about where we’re at and why. You’re going to filter everything you say, which means our problem solving isn’t going to be efficient or effective — just the opposite of what you want at that point.

MM: It sounds like organizations must be losing absolutely astronomical amounts of value at the hands of this dynamic. Now that we know it’s out there, what can be done about it?

SH: At each level of the organization, there are things you can do around the margins. These involve communicating skillfully, knowing your purpose, and so forth. But the real leverage is at the level of organizational culture, which to a fairly large extent is going to be determined by the people in charge. Management needs to be exquisitely clear about how people will be judged and evaluated. Employees have to be downgraded or penalized when they truly aren’t pulling their weight or when they make real mistakes.

But — and this is very countercultural — I’d say 90% of the time, when things go “wrong” it’s not because someone wasn’t trying hard enough, or because they didn’t care, or because they’re too dumb. Instead, 90% of the time, it’s because a department is overworked, or the specs were ambiguous or unrealistic, or there wasn’t adequate training, or various parties weren’t communicating well, or there were pressures or incentives in the system that discouraged someone from raising something that would have saved the project.

DS: In other words, 90% of the time, a project runs off the rails because of something larger than what one person or even one department is contributing to the problem. If you are looking for one person or group to blame, you're making a crippling mistake that will inhibit you from finding the real cause of what's gone wrong, and hence, being able to diagnose and fix it.

MM: Well, maybe that clarifies my next question. Many technology projects are faced with a deadly combination: a fixed and aggressive deadline, at Internet speed, combined with requirements that are not completely defined at the outset, and cannot be. Incredible disparities exist because of the expectations that are established. It's sort of like a four-month deadline for building a custom home, where the buyer doesn't know what the final blueprint is going to look like. And all too often, what happens is that after the foundation is poured, the buyer decides they need a bigger house.

DS: That's a brilliant example of what we're talking about. Let's say that happened. So everyone is gathered around, and people are saying, "Well, this foundation is the wrong size. Whose fault is it?" At this stage, there's plenty of responsibility to go around. You could say that the buyer forfeits his right to change his mind once the foundation is poured, or should have been clearer before it was poured that it shouldn't be poured yet. You could say the architect should have known to wait until the buyer signed off on a final plan before moving ahead. You could say the construction folks should have recognized up front that the timetable was unrealistic.

SH: But if you're in the business of doing this — say you build 50 houses a year — then trying to figure out exactly who is to blame isn't helpful. What's helpful is trying to understand why this happened — beyond just, "Well, we had a tight deadline" — and work to fix that dynamic. In order to do that, you have to look at the whole system, all the various choices and incentives that are in place that led up to the outcome you got, and then ask yourself, "What needs to change so this doesn't happen again?" This is called looking at the "joint contribution system."

MM: So you're saying, that if, for example, you fire the foreman of the construction team, that's unlikely to fix the real problem.

SH: Right.

MM: What would fix the real problem then?

SH: Joint contribution systems are complex, and they'll be different in each case. But one way to look at the system is to reverse-engineer a "mistake." So you'd say, "We're aware of this dynamic where we're under ridiculous time pressure, and the specs get changed in the middle, and then we crash and burn. What needs to be done differently so that doesn't happen?"

DS: There's nothing that will make the problem of time pressure or uncertainty go away, obviously. You can't just say, "Instead of needing this project in three months, let's get it to you in a year or two."

SH: In other words, after your competition has completely cornered the market.

DS: Nor can you say, "We don't really know how the final project should look quite yet. So let's just make something up and stick with it." Instead, what you'd want to do is discuss both of these concerns up front.

MM: And what does that get you?

SH: It does two things. First, you set expectations. "You need this in three months. Under the following circumstances, we can deliver on that. Under a different set of circumstances, it will take us four months, and under a different set of circumstances, it will take us five months." And you need to be as clear as possible about what each of those circumstances were, so that when they arise, you're still, in effect, meeting expectations.

MM: But the purchaser is going to come back and say, "We don't care about what the circumstances are, this has to be done in three months!"

SH: Which they might. A couple things: First is, if it's absolutely, rock-bottom true that the project is not doable in three months, then I think you have to say that, and be clear that competitors who say otherwise, assuming they don't have greater capacity, are either not telling the truth or aren't fully informed. Rather than lying to your potential client, it's your job to persuade them that what they are asking for simply can't be

done. The second thing is that by identifying the problem up front, you have the opportunity to problem solve up front.

DS: You're saying, "Here's the problem we're going to run into. Let's brainstorm and see if we can come up with some creative way to prevent that."

MM: It sounds like honesty is a touchstone of your thinking around this.

DS: The ethical virtue of honesty turns out to be pleasantly aligned with the business virtue of getting things done efficiently.

MM: But here's my problem. In the real world, nothing is ever quite so clear cut. For example, you might think you can get a project done, when you can't. Or you might imagine that if you hire more people and work them around the clock, you'll be able to pull it off. And you might be right most of the time.

SH: Well, that comes back to being clear up front. If your calculation is, "If every single thing falls into place and the stars are all aligned between now and when the project is due, we can get this done," that's different from, "We can get this done." You have to be clear with the client what the variables are.

DS: That also gets to a piece of uncertainty we haven't mentioned yet, which is on the client's end: what to do if the specs for the project change once the project is in process. There again, you confront the problem, which is absolutely foreseeable, up front, and problem solve. "What if this happens? And how about this? And what about this?" And so forth. If you think there's a 10% chance something "unexpected" will happen, then it's not really unexpected, is it?

MM: There's a statistic that highlights the importance of all of this. A recent study suggested that 85% of IT organizations failed to meet business users needs. That's a pretty scary figure. If it's true that there are contributions from both sides for this state of affairs, how would one broach this subject without getting into a downward spiral of accusations and defensiveness?

SH: Again, we're simplifying everything here to bring things into higher relief, and we don't want to give the impression that this is all very easy to fix if you just follow three

little steps. But we can all do better than we're doing at managing all of this. When things are going wrong and you need to try to get the project back on track, there are some things you can do. First, you've got to set the purpose of the conversation right up front. You must be clear that you're investigating what we've each contributed to the problem, how we've gotten to where we are, and what needs to happen differently going forward. Use "we" instead of "your people and our people" — not to be friendly, but because it's true. Keep reminding yourself and everyone else that when it comes to getting the project done right and on time, you're all on the same team.

DS: The hardest thing to do is to keep things on a productive track if the other side is talking in terms of blame. They say, "This is your fault." Instead of saying, "Wrong, it's your fault," you have to say, "I suspect there are things I wish we'd done differently, and also things you guys might have done differently. Let's take a look at how we got here and what we need to change going forward."

SH: And let's face it, everything changes once lawsuits become a possibility. Once you set out the specter of a lawsuit, you're basically giving up on the possibility of problem solving. But all of these conversations are intended to avoid that in the first place.

MM: There is another thing I would like your comment on. Once a conflict starts to become traceable to a contractual dispute and it looks like money is going to be on the table, in some cases, lawyers may say, "Okay, gag order time, don't talk, don't admit anything." How do you break that?

DS: That is just about the hardest question.

SH: I agree. Being lawyers ourselves, I think that is one of the hardest issues in conflict "escalation" today. The rules of the litigation game have such different incentives than the rules of a game that would actually be constructive in repairing the relationship — figuring out together how to go forward in a way that minimizes the damage and maximizes the upside.

DS: A great analogy is Truth Commissions, like in South Africa, where they've faced a very difficult choice. They said, "Do we want to have truth and mend our society, or

do we want legal justice?" They didn't feel they could have both, because once you say, "We want legal justice," and they start putting people in jail, everyone shuts up and you never learn the truth. It is the same in organizations where something goes wrong and lawyers get involved. There's this huge penalty involved — maybe hundreds of millions of dollars — that shuts down the process of understanding what went wrong, how we got here, and how we can fix it. It just shifts you into the legal framework of "It's *your* fault" or "No, it's *your* fault," and it becomes a contest.

SH: The irony of the situation is that many — perhaps even most — lawsuits are driven by emotions, emotions that are created when folks shut down, and refuse to be honest with each other and address the problems with an eye toward fixing things. The dollars involved are often a proxy for the frustration people feel when a "partner" lets them down.

I think the question has to come down to the specifics of the situation. Certainly, there are situations where you would decide the risk of being honest about the mistakes you've made is too great. And there may be situations where being open about your contribution to the problem and maybe even apologizing, while somewhat risky, is the better approach. You may say, "Look, we messed this up on our end, and we confused what needed to be in place at this date, but we are getting it fixed now, and here is where we are going from there." Often, that will be enough. You actually increase your partner's confidence in your ability to handle the job and get things done even when the going gets rough.

DS: Lawsuits shift the purpose. To go back to Sheila's dinner party example, instead of asking, "How can we get this casserole in as good shape as possible and make this dinner a great experience for our guests?" you're saying, "To hell with the guests! What's important here is that the fact that this burned casserole is your fault, and you're going to pay." We're not saying lawsuits aren't sometimes necessary, but once you get into that arena, you're into the domain of blame and evaluation and more or less out of the problem-solving domain. As Sheila says, the more clearly and skillfully you handle the problems at the outset, including all the uncertainties that go into determining what will be delivered when, the less likely you are to find yourself having to fend off blame.

About PON

Based at Harvard Law School, PON is a collaborative effort among faculty at Harvard, MIT, and Tufts University to design, implement, and evaluate improved dispute resolution practices. Its ongoing projects include the Dispute Resolution Program, the Harvard Negotiation Project, the Negotiation Roundtable, the Project on Preventing War, the MIT-Harvard Disputes Program, the Program on Negotiations in the Workplace, the Project on International Institutions and Conflict Management, and the Project on the Psychological Processes on Negotiation. For more information, visit www.pon.harvard.edu.

Correction

In the January issue of *ITMS*, Vol. VI, No. 1, the article on page 11 by Michael Wills should have been titled "Functional Reliability." We regret any confusion this error might have caused.

Resources and References

Carleton, Anita, Robert Park, and Wolfhart Goethert. "The SEI Core Measures: Background Information and Recommendations for Use and Implementation." *The Journal of the Quality Assurance Institute*, July 1994.

Heen, Sheila, Doug Stone, and Bruce Patton. *Difficult Conversations: How to Discuss What Matters Most*. Viking/Penguin, 1999.

Patton, Bruce, Roger Fisher, and Bill Ury. *Getting to Yes*. Penguin, 1999 (2nd edition).

Schaeff, Anne Wilson and Diane Fassel. *The Addictive Organization: Why We Overwork, Cover Up, Pick Up the Pieces, Please the Boss, and Perpetuate Sick Organizations*. HarperCollins Publishers, Inc., 1988.

Pickering, Chris. *Survey of Advanced Technology — 1998*. Available from Cutter Information Corp. at www.cutter.com/itgroup/reports/sat98.html.

Argyris, Chris and Donald A. Schon. *Organizational Learning II: Theory, Method, and Practice*. Addison Wesley, 1996.