IT METRICS STRATEGIES

Helping Management Measure Software and Processes and their Business Value

From the QSM Database: Productivity Statistics Buck 15-Year Trend

by Doug Putnam

The QSM database is one of the most comprehensive repositories of modern-day software projects collected worldwide. It contains trends from more than 5,400 completed software projects from North America, Europe, and the Far East, representing more than 200 million lines of code (LOC), 100+ development languages, and 55,000 person-years of effort. During the past 20+ years, QSM has maintained this database, analyzed it, and provided the results of this analysis to companies to serve as their own repository for their software and IT metrics.

The largest segment of data currently in the repository represents IT projects. Over the years, QSM has continuously monitored application-development productivity with respect to cost reduction, speed, and quality improvement. Generally, since the early 1980s, all of these dimensions have steadily improved.

However, a recent productivity study revealed that this trend has undergone a reversal during the most recent three-year time period. IT applications completed between 1982 and 2000 were extracted from the database. The data sample was then sorted into six three-year time periods spanning 18 years. In each time period, trends were plotted for project size, average productivity index (an indexed measure of overall project efficiency),

Continued on page 2.

Secrets of a Benchmarking Consultant

by Michael Mah

Metrics is a people business. Having spent more than 15 years in the metrics field, that concept has reinforced itself with every engagement I've undertaken, first as a project leader within large companies and later as a managing partner in a private consulting and training firm. Measurement may initially seem to be about benchmarks, trends, and data, but what comes first is getting the data. And to do that, you have to be with and talk to people.

I've had the pleasure of knowing some of the best in the business when it comes to this aspect of metrics work. Although their skills and backgrounds are diverse, they all share a common characteristic: good interpersonal and communication skills. Whether you serve as an internal consultant within your firm or are engaged by a company to come in as an outside expert, I've found that *how* you gather metrics is as important (if not more so) as the actual metrics you collect.

Continued on page 7.

august 2001 vol. VII, no. 8

executive summary

IT metrics data — what's happening in the industry and in your company — takes the stage for this month's ***ITMS***.

I'm pleased to include an article on recent IT productivity trends by Doug Putnam of QSM. Over the years, QSM metrics research has been adding new data on an ongoing basis into an industry-wide projects database. Trends have been plotted over time, and, recently, a story emerged about how we transitioned through the millennium. Something major has been happening.

Complexity is clearly at an all-time high. Companies have churned staff, the industry had to deal with Y2000, the Web happened in a big way, and new architectures were thrust on IT organizations at a rapid rate. There was a lot to handle. We've all felt it, and the numbers show it: productivity is down. Given the huge changes, this is not surprising.

These latest developments reversed a 15-year productivity trend, so we believe the downturn is temporary. There is also good news in that quality levels are way up. We hope this article will help put things into perspective; even though the last three years involved quite a dip, the overall picture is positive.

Next is an article I wrote, entitled "Secrets of a Benchmarking Consultant." It deals with the fact that although metrics may seem to be about numbers, it's really a people business. If you need to gather stats within your organization, you'll want to read these pointers on how to work with people to retrieve critical information and then communicate it effectively.

Finally, Jim Mayes talks about developing a balanced approach to benchmarking productivity, using weighted dimensions for speed, cost, quality, and throughput. This is the basis for creating a balanced scorecard, which can be used for either internal purposes or for outsourcing. We hope these articles tie everything together to help you get the numbers you need to be a better IT manager.

Michael Mah, Editor

CUTTER CONSORTIUM

Continued from page 1.

schedule, effort, staff, mean time to defect (MTTD), and reuse.

In the context of significant industry dynamics — Y2000, enterprise resource planning (ERP) solutions, the dot-com explosion/implosion, outsourcing, object-oriented (OO)/client-server development, etc. — the charts reveal significant findings in the context of long-term productivity trends. These are described below.

Developed Software Size Behavior over Time

Figure 1 shows the average project size based on new-plus-modified functionality for IT projects, beginning with a three-year time period, starting in 1982 and continuing through 2000.

The overall trend through the 1980s and early 1990s was a steady reduction in project size. On average, the size of software projects was cut in half during the 15-year period from 1982 to 1997. This was generally the result of more powerful development languages as technology progressed, along with deliberate strategies by IT organizations to manage projects to 12- to 18-month schedules. Generally speaking, there was also implementation of reuse architectures such as object libraries and classes as "buy and modify" IT strategies, versus building applications from scratch.

In the 1997-2000 time frame, a radical change occurred, in which the average size of IT projects virtually doubled. This was the case in size measured by both function points and LOC, and it reversed the 15-year trend in a dramatic way.

One potential cause of this is the explosion of Internet, e-commerce, and Web development architectures during the 1997-2000 time frame. This coincides with many first-generation Web products (both sites and tools) where there was no previous existence of reusable code. Many of these applications had to be built from scratch.

Although no one can say for sure, we speculate that software size may reduce in the near future in a gradual fashion, as the architectures built during the last three years are leveraged in future generations of IT projects.

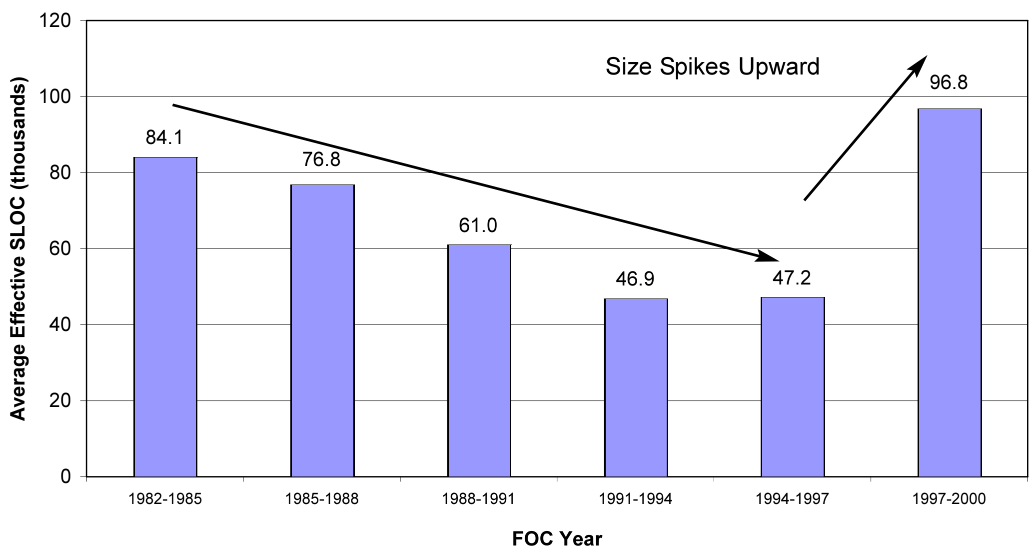

Figure 1 — Average effective size versus three-year time periods over 18 years.

Editorial Office: Clocktower Business Park, 75 South Church Street, Suite 600, Pittsfield, MA 01201, USA. Tel: +1 413 499 0988; Fax: +1 413 447 7322; E-mail: mmah@cutter.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: Tel: +1 781 641 9876 or, within North America, +1 800 492 1650, Fax: +1 781 648 1950 or, within North America, + 1 800 888 1816, E-mail: service@cutter.com, Web site: www.cutter.com/consortium/.
Editor: Michael Mah. Publisher: Karen Fine Coburn. Group Publisher: Kara Letourneau, Tel: +1 781 641 5126, E-mail: kletourneau@cutter.com. Production Editor: Pamela Shalit, Tel: +1 781 641 5116. Subscriptions: $485 per year; $545 outside North America. ISSN 1080-8647.

Average Productivity Index Performance over Time

The QSM Productivity Index is an aggregate measure of process productivity, calculated from metrics for size, time, and effort of completed software projects. These represent three of the four core metrics expressed by the Carnegie Mellon Software Engineering Institute Minimum Data Set — an established industry standard.

The productivity index (PI) is different from traditional measures of applications productivity that emphasize only two dimensions of metrics, such as output size (i.e., function points or LOC), per unit effort (person-months). It incorporates development schedule shortening or lengthening, by including time in its calculation. Therefore, each index rise corresponds to a reduction in effort (about 25%) and/or a shortening in time (about 10%) from the previous value.[1]

The calculated PI increased over the 15-year period from an initial base value of 13.8 to 17.3 by the year 1997. However, during the 1997-2000 time frame, the three-year average dropped to 16.6 (see Figure 2).

We believe several factors may have been at play to drive productivity downward. These include:

[1]Traditional metrics for productivity do not include time. Therefore, if effort improves but schedules lengthen, their "improvement" can potentially be misleading.

- The adverse impact of resources diverted to Y2000 projects
- Labor churn from the rotation of staff to e-commerce and Web initiatives
- Significant learning curve associated with customizing and implementing large-scale applications such as ERP
- Dramatic shift in project complexity from traditional IT applications to those that incorporate more complex elements such as wireless telecommunications, system software, fiber optic, and even real-time elements

Staffing Performance over Time

Figure 3 shows an average staffing profile on a typical project over time. The trend has been reasonably constant, in the range of 6-7 people per project during the 1990s. In 1997-2000, the average project team increased to 9 people. That's about a 50% increase.

There appear to be two factors contributing to this trend: project size growth and an acceleration of project deadlines to complete at Internet speed. In essence, companies are striving to build even more functionality in less time, and they react to these pressures by adding more people to projects.

Schedule Performance over Time

Figure 4 shows the average duration of projects for each three-year time period. In 1982,

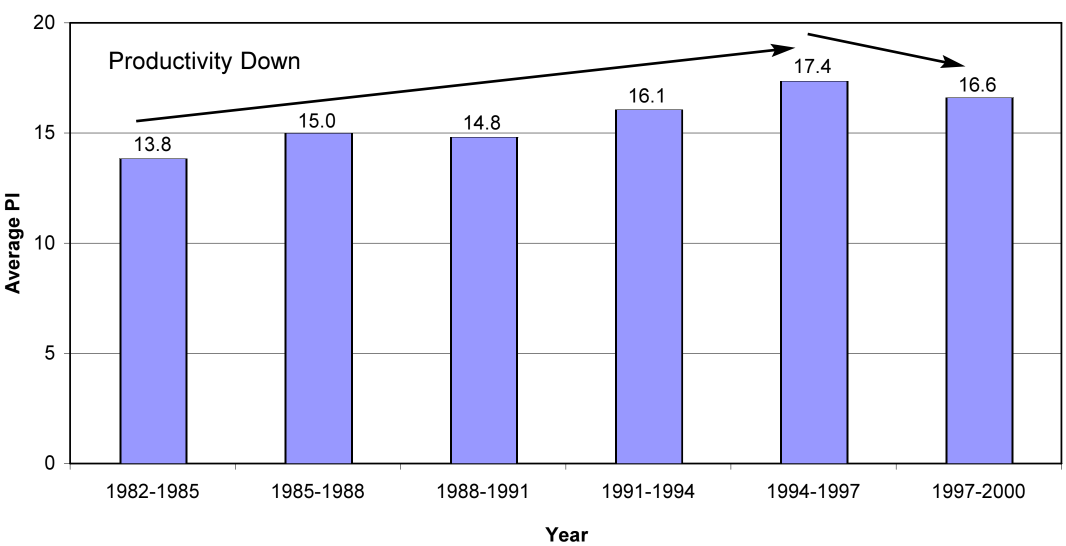

Figure 2 — Average productivity index versus three-year time periods over 18 years.

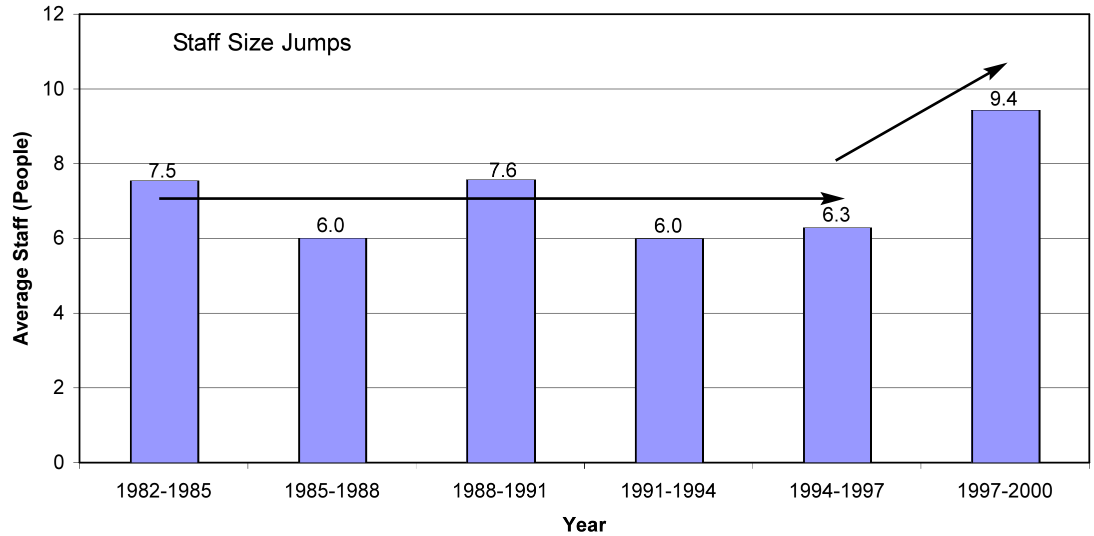

Figure 3 — Average staffing versus three-year time periods over 18 years.

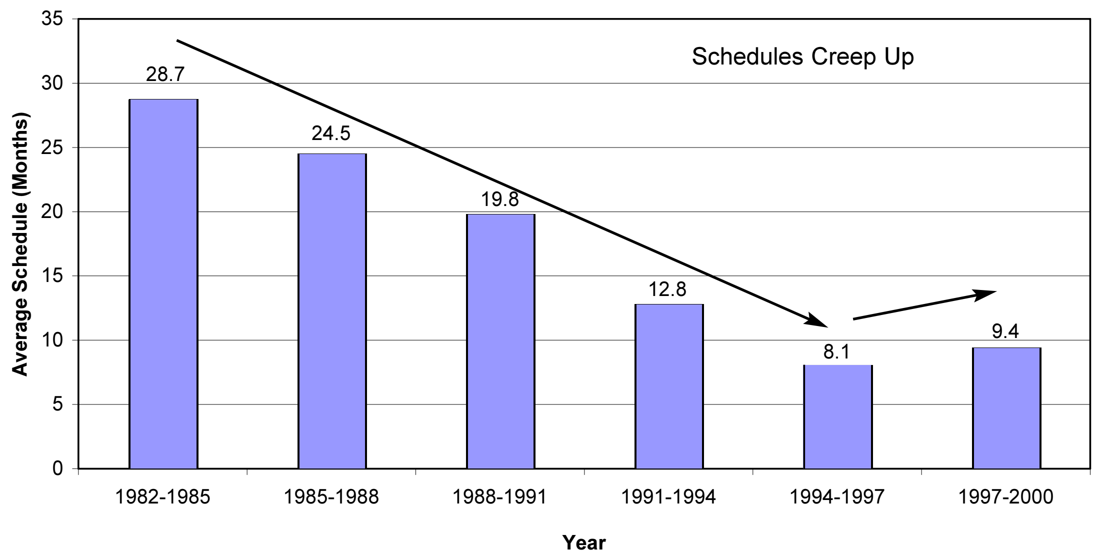

Figure 4 — Average schedule versus three-year time periods over 18 years.

the typical IT project lasted nearly 2.5 years. By the 1994-1997 time frame, it had dropped to 8 months! That's a pretty impressive trend over the 15-year time period. The two most important driving factors were the reduction in project size and the improvements in productivity (see Figures 1 and 2). Both behaviors result in schedule reduction.

It's interesting to note the industry studies on overruns and slippages, beginning with a study by the US General Accounting Office in 1979, up to and including the Standish Group Chaos Report in the late 1990s. The data is irrefutable — projects have completed faster and faster every year. One can only surmise that the industry "overrun and chaos" studies reflect that demands and expectations outstrip even this dramatic rate of improvement. Internet-speed deadlines, overly optimistic estimates, and scope growth/change may be more of the culprit with respect to projects being "late."

However, in the 1997-2000 time frame, this trend sustained a reversal, in which the average project schedule grew to 9.4 months. This is likely due to the combination of project growth and the aforementioned drop in the QSM Productivity Index.

Effort Performance over Time

Figure 5 shows the average effort for each time interval. In the 1982-1985 time frame, the average effort per project was more than 165 person-months; 15 years later, that figure dropped dramatically to less than 60 person-months.

However, in the 1997-2000 time frame, average project effort (and associated cost) nearly doubled to more than 100 person-months, at a cost of about US $1.5 million. This is a dramatic reversal of the 15-year trend. It appears to be a combination of project growth/drop in reuse, the 50% increase in average team size, and the modest drop in productivity.

One might expect that rising project costs might be a factor in companies attempting to reduce costs by outsourcing in the economic climate over the past several years.[2]

Software Reuse Performance over Time

Figure 6 shows the trend in software reuse. The reuse is expressed as the percentage of reuse that was achieved during the three-year time period. With the exception of the initial three-year data segment, the overall trend from 1985 to 1997 was an increase of software reuse. This approached 65% during the early and mid 1990s.

During the 1997-2000 time period, the trend retreated back to approximately 50%. We believe that most of the Internet and first-generation OO projects comprised new development.

[2]It should be noted that the QSM database is not restricted to projects built inhouse. It contains statistics for both inhouse and outsourced projects.

Looking forward, it's likely that 60%-70% reuse is the practical upper limit that can be realistically expected over a broad range of products in an organization in the normal course of business.

Mean Time to Defect over Time

Figure 7 shows the trend for reliability at delivery over time. The reliability is expressed as MTTD. It is the average time between occurrences of runtime errors in a software application.

The data shows MTTD remained relatively constant during the 1980s, at about five days on average, and improved to just under nine days during the 1994-1997 period. Then, during the three-year segment from 1997 to 2000, it improved dramatically to an average of 12.5 days.

Contributing factors are likely to include improved process maturity and more attention to quality issues. Many modern-day software applications require 24-hour-a-day/7-day-a-week operation, with greater emphasis on system availability. All of these advances appear to be producing good results with respect to quality.

Conclusions

In summary, the data reveals significant changes in applications productivity in the 1997-2000 time frame. With the exception of

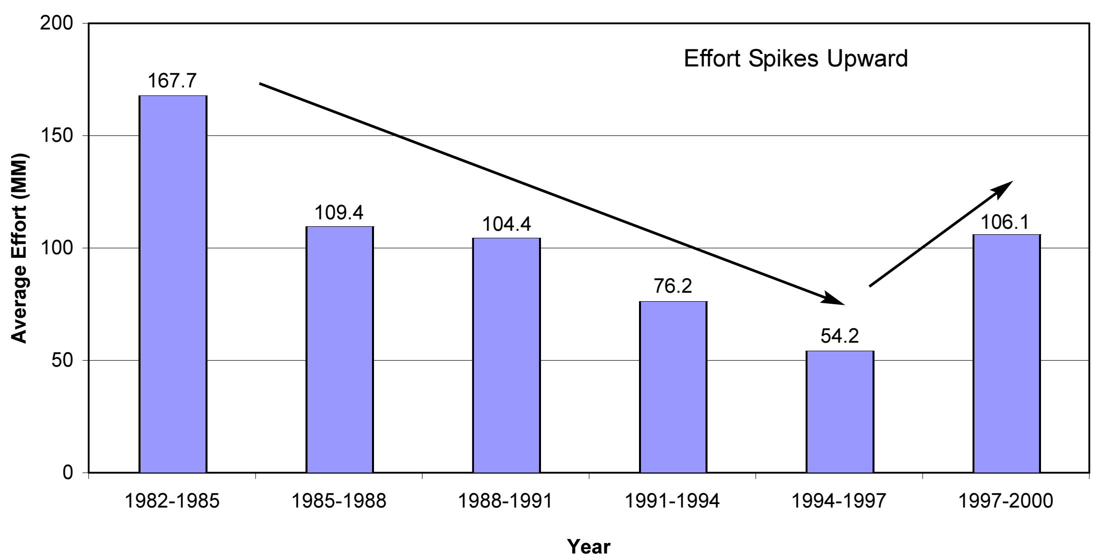

Figure 5 — Average effort versus three-year time periods over 18 years.

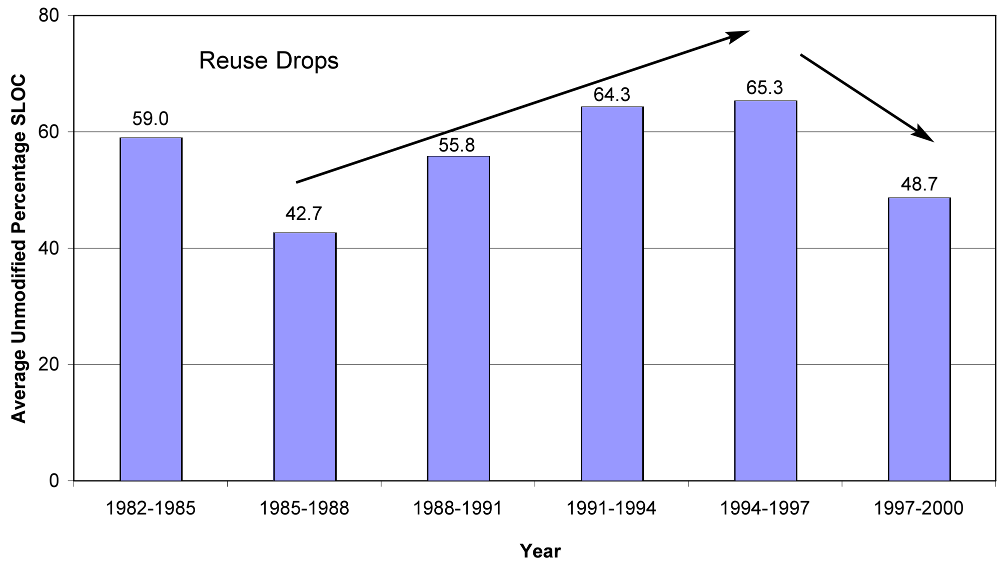

Figure 6 — Average reuse versus three-year time periods over 18 years.

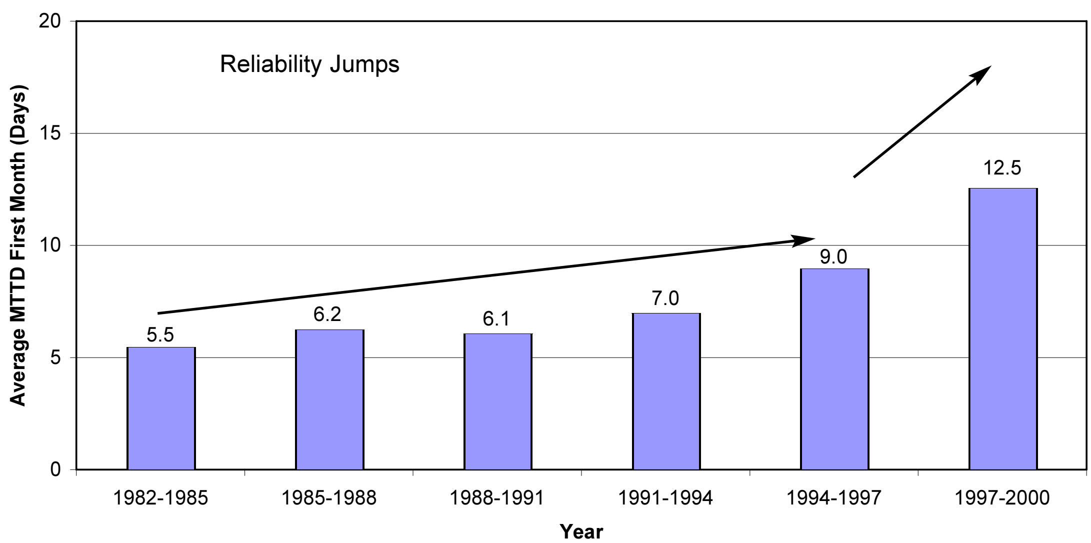

Figure 7 — MTTD versus time period over 18 years.

quality improvements, all other indicators countered a 15-year improvement trend:

- Staffing was higher
- Effort was higher
- Schedules took longer
- Software reuse was lower
- Project size (new + modified functionality) was much larger
- Productivity was down

We speculate that several underlying factors were at play during this time frame, which may have been root causes of these results. It appears they comprise a significant rise in IT project complexity during a time of dramatic change. In this context, temporary drops in productivity make complete sense. These factors include:

- Implementation of packaged ERP solutions were attempted on a large scale in many organizations around the Y2000 time frame. The complexity of these endeavors was widely underestimated during the planning stages in many cases.

- OO development was started in earnest. This meant a new infrastructure had to be built for all the application-specific classes. Many organizations discovered that implementing OO was more difficult than anticipated. It took more time and more effort during its initial adoption.
- Web-based development and the advent of dot-com enterprises pulled many talented engineers away from traditional development, producing turbulence as highly skilled people left for Internet startups and lowered the overall skill level in the *Fortune* 1000 companies.

The data shows that productivity and the other associated management metrics don't always improve linearly year over year. We believe the data is revealing the turbulence that was experienced during the past three years. However, over the long term, there is no question about the productivity improvements our industry is exhibiting. The recent data may simply be reflecting a slowdown that manifested itself during the turbulence of the Y2000 transition and the Internet and e-commerce revolutions.

This analysis demonstrates the industry insights that are possible through the use of metrics. Some companies conduct analyses like this within their own companies, but many others don't know where they are, where they are headed, and have no road map to guide their decisions moving forward.

To avoid this fate, establish your own productivity benchmarks and set a process improvement plan in motion. If you do, it will be possible to see and explain what's going on in your company and take proactive steps on issues that can improve your productivity and set your company ahead of the pack.

About the Author

Douglas Putnam is the vice president of Professional Services at QSM. He has more than 19 years of experience in the software measurement industry. Mr. Putnam has written and lectured extensively throughout the world and has participated in more than 100 estimation and measurement engagements in his career at QSM. He can be reached at doug_putnam@qsm.com.

Secrets of a Benchmarking Consultant

Continued from page 1.

If you find yourself in this role, it's important to understand that people are coming to you or your group looking for advice on gauging what's going on within their work processes — good, bad, or indifferent. And although some may view measuring work as a purely professional endeavor, nothing could be further from the truth. Work is personal. As Julia Roberts exclaimed in the film *Erin Brockovich*, "The time I spend at work is time I'm not with my kids. It doesn't get any more personal than that."

When an organization asks for assistance with a benchmark, people in that organization may feel vulnerable. The organization may feel that its identity is at stake, usually around issues of competence when something like "productivity" is being measured. Thus, gathering metrics often places people's feelings and "perceptions of self" (for both individuals and for the organization) on the table. That's a far cry from benchmarking being purely about just "data."

As we go through what's involved in benchmarking in this article, I'll be keying in on the people aspects. These ideas should ease your way through the process, whether you're the person collecting metrics, the one furnishing them, or the recipient of metrics findings.

Measure Processes, Not People

IT work is never about individual endeavors. In technology, it's about teams. And teams are effective because of (and sometimes in spite of) work processes. The development of information systems is about leveraging technology to manage information. Most of the time, we are in the business of creating new ways to manage information. As such, IT is knowledge work: teams of people coming together to create solutions that solve problems. Knowledge work is also about thinking, sometimes on our own, often more in the context of a group to which we belong.

Most of the time, when companies want metrics, they desire a quantitative understanding of what's going on in these group processes, especially for large-scale work of high importance to the organization. That means we should be going after metrics that give them the most bang for the buck. Most often, this rules out collecting information on IT projects characterized by one person working part-time or projects of short duration. We want information on broad IT processes to acquire understanding about team dynamics and the organization's processes. It's very unusual for a small project to be part of an organization's top-10 list, unless you're working in a very small company.

Organizations also want to understand communication dynamics. In knowledge work, this is usually where the high stakes are. Among teams designing and implementing IT projects, communications are where efficiencies are created or destroyed. We also want to understand the degree of positive impact and business value that's created by the implementation of information technology.

It's a good idea to compile a list of IT projects that meet meaningful selection criteria. Imagine saying to senior executives, "What are the 10 projects that are the most vital to our company?" Those are the projects you should be measuring; creating value for a metrics program means applying metrics to projects with high importance to the organization. The goal is to positively impact projects that make a difference to the company's competitiveness and performance in the marketplace.

Framed in this manner, a metrics benchmark is viewed as a strategic activity, not overhead.

The Metrics Project Launch

Establishing IT productivity baselines, especially in projects like software applications development, requires a successful benchmarking launch. The first step involves getting the stakeholders together and conveying the purpose and mechanisms behind collecting data. Measurement is an area that tends to make people nervous. People need to be assured that the right metrics are being gathered, that they are credible and accurate, and that the results will be used intelligently.

Any tensions in this area should be brought out in the open. If these fears go underground, the data collection and validation could be negatively affected. A number of key questions are likely to arise. Exactly what data are we collecting? What projects will be included? How much effort and time commitment will this take? What should we do if we feel that the data is not accurate? What will the analysis look like, and how will it be used? Will the information be respected, and will parts be treated with confidentiality if necessary?

A metrics team should prepare in advance to answer these types of questions and have information templates, meeting schedules, timetables, and a game plan to communicate to participants. This meeting should take about two hours. At the end of the meeting, you'll want to give people all of the information listed above as a road map for what comes next. This is important. Most people are overwhelmed with work; if you're asking them to spend time away from other projects, you have to sell the idea to them. They need to feel that the task is clear, that they can get help from you if they're unsure of what they need to gather, and that something is in it for them when the results are completed.

Talking with People, Gathering the Data

Gathering data is about conversations. The term data collection sounds clinical and impersonal — it is anything but that. If an organization wants benchmark metrics, it is asking you to serve as an agent on its behalf, to acquire information and derive insights it does not presently have. Sometimes this information — items such as cost data, project schedules, software application failure rates, field outages, project size, and effort expenditures — are available from an organization's public records. But most often, this information resides in the desks of the people responsible for the work. You have to talk to them to get it.

This involves sitting down with people and conducting interviews. At least two weeks prior to the interviews, inform people about the kind of information you'll be looking for to avoid surprises. This helps them prepare and feel more comfortable. They may even start gathering the materials you need in advance. If so, the interview can be more relaxed, serving as a data review. Other times, the interview will involve extracting the information on the spot, requiring more extensive followup.

It's very important that the manager or consultant conducting the interview have good interpersonal, organizational, and communication skills. If project managers are anxious about metrics, it's up to the interviewer to put them at ease, project confidence in the process, and provide support (both personal and technical).

Begin by talking about the "story" of their project, especially any good news that may have been associated with it. Have them explain the work processes and the people who were instrumental in making things happen. This allows the project leader to tell his or her story and share personal experience. The interviewer must be a good listener — often, crucial information is buried in the story, and further layers of data are teased out during the conversation.

It's best to allocate no more than an hour to an hour and a half for interviews; fatigue becomes a factor during longer interviews. Construct a detailed action item list for acquiring followup information later if all the information is not obtainable in one session. The interviewer should allow for about 30 minutes alone after the interview to record summary thoughts and key aspects of the conversation. It's helpful to dictate these notes into a tape recorder while the information is fresh.

Checking and Double-Checking the Information

After the interviews are complete, you'll need to share feedback fairly promptly to give people a sense of participation and progress. Leave about one week (two weeks at the most) for closure of followup action items. Any longer than that, and the memory of the conversations might begin to fade. Keep a running matrix on Microsoft Excel for action items across all the projects; these should be sent out regularly to the project executive sponsoring the initiative and the participants. Keeping everyone in the loop maximizes productive communication and ensures that action items don't get pushed off indefinitely.

Once the data begins to come together, start to graph the metrics for cost, speed, project size, or transactions. You may want to examine what the data tells you from several dimensions. For example, if you're benchmarking transactions over time based on fixed IT staff headcount, then make transactions the metric as a function of fixed time and effort. You may also want to examine time and effort metrics as a function of number of transactions or project size. The key is to first determine your purpose. The questions you want to answer will emerge from that, and the appropriate metric will logically follow.

These types of graphs are simple to construct using Excel. You may also want to try one of the commercial metrics database repository and analysis tools that are available from various software measurement firms. (Many of the charts used in ***ITMS*** were created using the SLIM-Metrics productivity benchmarking database from QSM.)

Usually, data that seems awry makes itself apparent at this stage; for example, major outliers will show up clearly on the charts. Pictures tell a strong story, so it's important to portray the metrics graphically to reveal anything that might require double-checking.

Understanding the Whys Behind the Numbers

If you have projects that deviate significantly from the norm, it's a good idea to review the data to make sure there are no errors and then examine the significant drivers, both positive and negative. You may need to have followup conversations with the people who provided the data. Share what you're seeing with them, and bring them into the process. They will likely be glad to help, since they have a vested interest in seeing what you've found. Chances are, they'll want to know whatever insights are emerging to help them validate their ideas on how to make things better.

It's in this phase that you begin to reveal the root causes of productivity issues in the organization. The results will direct people's attention toward solving the IT productivity challenges they share. This creates the basis for collaboration, pooling of respective talents in the organization, and improving work processes.

The findings need to be presented to at least two constituencies: the management group sponsoring the initiative and the individuals who took the time to contribute data to the process. Communicating the results to those who enabled the data to be gathered addresses short-term interests by providing them with near-immediate results of their contribution.

It also creates positive energy in support of future metrics activities.

Using the Findings Wisely

The goal of generating benchmark findings should be to answer some clear-cut questions (such as the ones below). These questions should provide a road map for future updates to the baseline:

- How productive is the organization on its most critical projects?
- What are the findings across different areas of our business?
- Is IT application development and maintenance productivity increasing or decreasing, and at what rate?
- Are schedules getting shorter?
- Are effort and cost decreasing?
- Is reliability improving?
- How do we compare to others in the industry?

When drawing conclusions from the data, it's very important to be sensitive to careful use of language, avoiding words implying judgment or evaluation. For example, rather than saying effort metrics are getting "better" or "worse," say that effort metrics are moving "higher" or "lower." The truth is, it's difficult to make accurate judgments at first glance. The language of evaluation puts people on the defensive, causing them to worry about their reputation rather than focusing on the data and conducting a thorough analysis to understand the causes of IT project behavior.

Because of the intense pressure IT organizations are under, people are at risk of feeling harshly judged. That's not the point of benchmarking. If people get the wrong idea, it can lead to all sorts of problems, from coverups to quiet sabotage of the metrics program. Difficult conversations arise when facts are disputed because strong emotions come into play, especially if identity issues or feelings of competence are at stake.

Focusing on process and framing the results into lessons learned enables everyone to explore solutions to the problems they share. Everyone is in this thing together, and good results can be achieved with cooperation. Too often, if the numbers are not what people were hoping for, the knee-jerk reaction is to focus on who's at fault. It's important to be sensitive to this fact and present the results wisely.

An organization trying to raise itself to the next level of productivity must master not only the mechanics of acquiring knowledge about its performance, but also the ways management and staff react and respond to the information that's revealed as part of the measurement process. If you're successful at handling these people issues, you can play a vital role in elevating your organization's capabilities and improving how people interact with one another, resulting in a happier work environment.

About the Author

Michael Mah is editor of ***IT Metrics Strategies*** and managing partner with QSM Associates. He can be reached at mmah@cutter.com, or by phone at (413) 499-0988.

A Road Map for Balanced Productivity Metrics: Are We There Yet?

by Jim Mayes

The Balanced Approach

Philosophy

IT outsourcing continues to be a popular course of action for streamlining businesses. These arrangements always include measurements of performance and productivity improvement. The questions that we should be asking are:

- Why isn't it just as important to measure performance and productivity related to inhouse IT?
- Why don't we develop insourcing contracts as an alternative to outsourcing?

Whether IT is inhouse or outsourced, performance and productivity are important.

You know how kids in the backseat of a car are always asking, "Are we there yet?" Most companies make outsourcing decisions without knowing their current IT capability — and are often surprised to find out they were better off prior to outsourcing. Similarly, many companies embark on software process improvement programs without knowing their existing level of performance and without any metrics in place to measure productivity improvement. It's like the old saying "If you don't know where you are, a map won't help." This applies to measuring progress as well. It is difficult to know if we're "there yet," if we don't know where we started or exactly where we need to go.

Definition of BPM

Balanced Productivity Metrics (BPM) incorporates the use of both quantitative and qualitative data for measuring performance and productivity improvement. The primary goals of a Software Engineering Institute (SEI) Capability Maturity Model (CMM)-based process improvement program are to improve productivity, improve quality, and reduce risk via consistent processes. BPM focuses on the SEI CMM core measures (size, time, effort, and defects), as well as other data collected to measure process improvement. A multidimensional approach is needed to measure productivity improvement in a way that also provides an understanding of the environmental factors (and other significant factors) that influence project productivity.

The BPM approach is used to measure productivity with regard to software development and maintenance. BPM is based on the principle that the management of productivity improvement should focus on achieving a balance of time/schedule, cost/effort, and quality/defects (time, cost, and quality [TCQ]) improvement.[1] It is consistent with a balanced scorecard philosophy, since these metrics components cannot be observed independently with regard to providing a valid productivity assessment. Balancing TCQ is much like filling a balloon. If we compress or fill it too much, it pops. If it doesn't pop, and we push in one place, it expands in another. Likewise, if we compress project schedules too much, then project effort, project-related defects, and production defects are increased, which makes our maintenance effort go up. The reverse is also true. Project schedules can be increased too much, which does reduce effort and defects, but may not provide the desired business value. Therefore, a balanced measurement approach is needed.

To quote Karl Wiegers:

> A risk with any metrics activity is dysfunctional measurement, in which participants alter their behavior to optimize something that is being measured, rather than focusing on the real organizational goal. For example, if we are measuring output productivity but not quality, expect some developers to change their coding style to expand the volume of material they produce, or to code quickly without regard for bugs. I can write code very fast if it doesn't actually have to run correctly. The balanced set of measurements helps prevent dysfunctional behavior by monitoring the group's performance in several complementary aspects of their work that lead to project success.[2]

There are two reporting components associated with the BPM approach:

1. Productivity metrics report (PMR)
2. Productivity process report (PPR)

PMR Quantitative Measures

The quantitative measures associated with the PMR are used to assess productivity improvement. The initial assessment period is considered the baseline year. Improvement is normally expected at the end of the second year over the baseline year. Every year thereafter, improvement is expected over the previous year.

The PMR uses a combination of metrics for calculating the BPM score, which measures success in achieving productivity improvement goals. The project metrics are calculated for each software enhancement and new development project, and then averaged for each productivity metrics component. The maintenance metric is calculated with regard to the yearly hours required to provide ongoing support for the total inventory

[1] Mayes, Jim. "Achieving Business Objectives: Balancing Time, Cost, and Quality." *IT Metrics Strategies,* Cutter Information Corp., March 2000.

[2] Wiegers, Karl. "A Software Metrics Primer." *Software Development*, July 1999.

of supported applications. The metrics components to be included in calculating the productivity score are shown in Table 1.

PPR Qualitative Measures

The qualitative measures associated with the PPR, when analyzed along with the quantitative measures, provide insight into the overall process productivity and direction toward achieving process improvement goals. Looking at the BPM score — which is a letter grade or number — only provides the current status of the contractual measure. Unless there is an underlying analysis behind that score, there is little information to go on for planning the next steps for improvement. The PPR provides information that can be used to identify problems, as well as areas that have improved, for managing process productivity improvement activities. This report provides the following analyses, trending, and comparisons to the previous period:

- Demographic data analysis related to organization profiles, project types, size, complexity, and languages
- Statistical analysis of metrics categories against previous performance trends
- Analysis of delivery performance, including requirements volatility
- Significant factors impacting project results

Table 1 — Balanced Productivity Metrics (BPM) Components and Calculations

Metrics Component	Calculation
Project Schedule Duration	= Total project months/total projects' units of product measurement (UPM) = Months per UPM (function points, lines of code, etc.)
Project Output Productivity	= Total project hours/total projects' unit of measure = Hours per UPM
Project Quality	= Mean time to defect (MTTD) for each project is calculated by dividing the number of days of exposure during the first 30 days after deployment by the number of defects found = SUM (each project's MTTD x each project's UPM)/total projects' UPM = Weighted average MTTD
Maintenance Output Productivity	= Total maintenance hours/total maintenance UPM = Hours per UPM

- Analysis of customer satisfaction survey results in relation to project data
- Analysis of environmental factors related to tools/methodology, technical difficulty, and personnel

Process Overview

Administration

The BPM process involves joint responsibilities. The client performance metrics manager (client PMM) manages the process and the database and serves as team lead for a productivity metrics team. The client PMM will prepare the PPR, and the IT metrics group will prepare the PMR, which includes calculating the BPM score. Change management with regard to this process will be the responsibility of the productivity metrics team. (These roles may vary depending on organizational structure. Additional roles are identified in Table 2.)

Frequency

The PMR should be calculated annually for productivity scoring related to the contract agreement at each contract anniversary. This report quantifies the productivity that was achieved during the course of the previous year. For purposes of monitoring progress and enabling the parties to promptly address any service or process deficiencies, the parties will also cause the BPM reports to be calculated and the cumulative results reviewed at the end of each quarter.

If at any time the BPM score is to be calculated, and one or more metrics or component metrics cannot be calculated, either because the relevant data is not available or the applicable scoring scale has not been adopted or is not yet in effect, then the BPM score should be calculated disregarding such metrics. The weights of the other metrics or component metrics within a given metric shall be increased proportionately so that such other metrics or component metrics maintain their respective relative weights.

Data Collection and Dictionary

Data consistency is the key to having a good decision support database; therefore, to ensure consistency, the definitions must be documented. Also, when building the internal historical project repository, the project data points should be categorized. This will

Table 2 — BPM Process Schematic Description

Activity	Performed by
1. Software Enhancement or New Development Project Launched Whenever new projects are launched, it must be determined whether they meet the criteria for being included in the productivity metrics process. There may be some projects that were launched prior to the implementation of this process that should be included as well.	**IT project manager**
2. Track Project Phase Effort, Schedule, and Defects The actual effort hours, start date, and end date for each project phase should be tracked. The project phases include planning, requirements analysis, and main build. Defects are tracked by severity (1-critical, 2-serious, 3-moderate, and 4-tolerable/cosmetic) from the beginning of system test until deployment. Defects must also be tracked by severity for the first 30 days after deployment.	**IT project manager**
3. Schedule and Perform Software (UPM) Sizing Function points, source lines of code, or some other method should be used to size the project and application.	**IT project manager and technician**
4. Enter Project Data into Detailed Data Collection Form On project completion, project data is entered in the detailed data collection form or spreadsheet. This form is sent to the IT program manager prior to the project closeout meeting.	**IT project manager**
5. Review Project Data The project information provided in the detailed data collection form should be reviewed and validated during the project closeout meeting. Agreement should be reached on the ratings and results provided. This can provide an opportunity for joint problem solving related to project impacts and process improvements needed. The validated detailed data collection form should then be forwarded to the client performance metrics manager.	**Client program manager**
6. Enter Detailed Project Data The data contained in the detailed data collection form is entered into the metrics database.	**Client performance metrics manager**
7. Track Ongoing Maintenance Effort Hours Actual effort hours charged to ongoing maintenance support must be tracked.	**IT project manager**
8. Schedule and Perform Application (UPM) Baseline Sizing Function points, source lines of code, or some other method should be used to size the applications being maintained.	**IT project manager**
9. Update Maintenance Effort and Size Spreadsheet A data repository should be maintained containing maintenance productivity data.	**IT metrics group**
10. Prepare Quarterly or Yearly BPM Report The productivity metrics report containing the BPM score is created by the metrics group, and the performance manager creates the productivity process report.	**IT metrics group and client performance metrics manager**

provide stratification such that apples-to-apples project comparisons can be made based on project type, environment, user organization, development organization, language, etc. The data collection requirements should also be defined and documented. The collection of software project data during the project is designed to track the status of projects, enable predictive analysis, and facilitate the collection of project closeout data required for the BPM reports. On completion of the project, the data should be summarized in a detailed data collection form.

Tools/Database

BPM project data should be stored in a metrics database. The project-level information in this database can be used for planning new projects and evaluating risks. At a minimum, this can be done by using data from similar historical projects for providing an analogy to the project being planned. This is associated with the CMM Level 2 Software Project Planning key process area and supports the software project estimation and risk management processes. (For more information on this, see my December 2000 ***ITMS*** article, "Saving the World One Project at a Time: Planning by the Numbers.")

Process Steps

The BPM process-flow schematic is illustrated in Figure 8. The BPM schematic description and responsibilities are provided in Table 2 on page 13. The BPM process begins with the collection of baseline project data and the development of the productivity baseline. The schematic illustrates the next steps after a baseline has been established.

Critical Success Factors

The following factors are critical to the success of a balanced productivity metrics program:

- Client and IT partnering relationship with open communication paths
- Senior management support and a documented policy statement
- Client program management, client performance metrics management, IT project management, and IT metrics group support and synchronized processes
- Accurate data collection and sizing, on a timely basis
- Using project-level quantitative and qualitative metrics data for software project planning and process improvement (i.e., don't just accumulate it for contractual purposes)
- Communication, training, and mentoring related to the BPM process

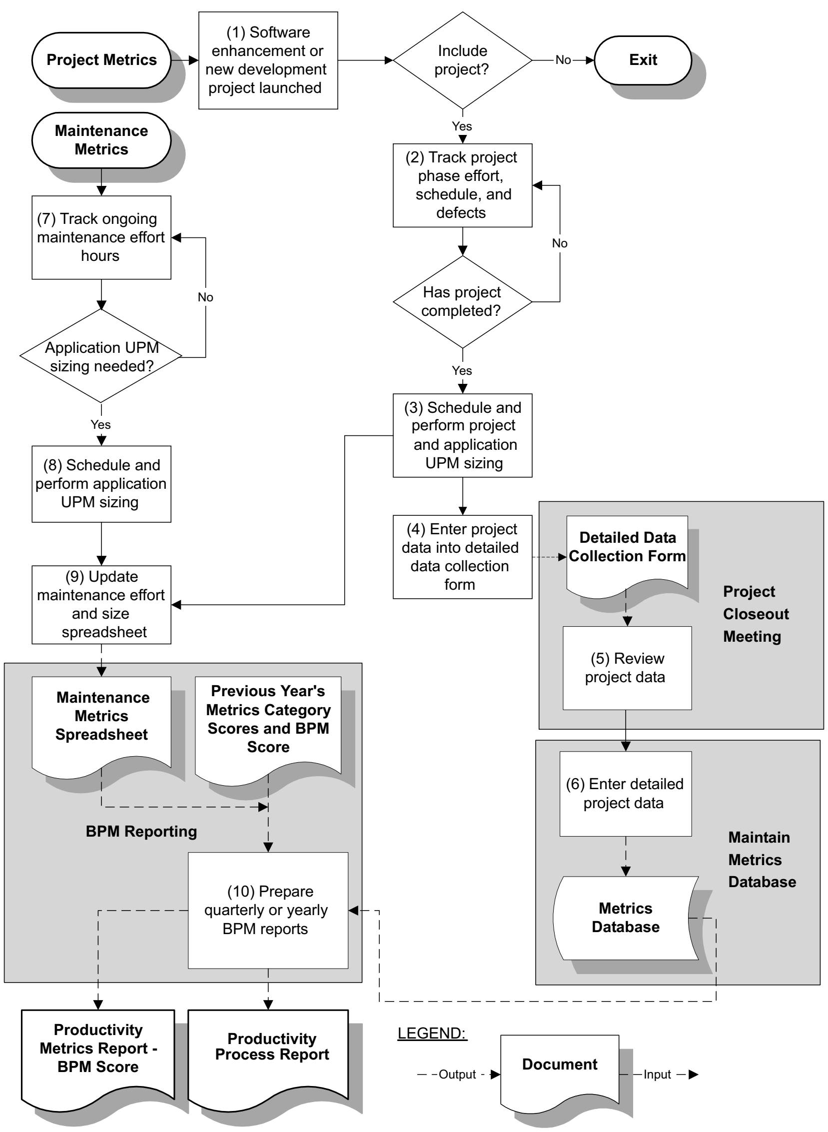

Figure 8 — BPM process-flow schematic.

Scoring

Metrics Component Scores

Each BPM metric component will be given a letter score determined in accordance with the scoring scale shown in Table 3, assuming that a 5% yearly improvement for each component is desired. This letter grade will be converted to a numerical score using the scale shown in Table 3. (The percentages and letter grades can be adjusted based on the yearly improvement desired for each TCQ category.)

Metrics Component Weighting and Balanced Scoring

Business value should determine the weightings with regard to project TCQ and maintenance, since this is a business decision. Of course, this will vary from project to project, depending on the circumstances, so the BPM weightings should reflect the overall or "averaged" philosophy. Component/category weightings should be established once a year, at the beginning of each annual measurement cycle. If everything were equally weighted, the total BPM composite score would be calculated as shown in Table 4, and the composite score would be calculated as shown in Table 5. Examples of BPM score calculations are provided in Tables 6, 7, and 8.

Conclusion

The measurement of productivity and performance is just as important for inhouse IT as it is for outsourced IT. A balanced approach, as provided by the BPM process, is needed to ensure that all aspects of productivity are assessed and that the correct behavior with regard to performance is achieved. Some outsourcing vendors will argue that only one measure should be used, such as effort per unit of output, but to most clients, schedule and quality are equally important. Therefore, measures should be in place such that schedule or quality cannot be sacrificed. It is equally beneficial to IT that all factors be measured, due to clients' ever-changing TCQ priorities. In addition to balancing the TCQ metrics and quantitatively tracking projects, it is also important to understand the relationship between the quantitative and qualitative measures that influence productivity. When we trend the quantitative and qualitative data and compare the results to previous internal benchmarks, our corporate knowledge will be increased so that we can make intelligent business decisions. The BPM process can help us understand the factors that influence our productivity, know where we are on the map, effectively predict where we are headed, and recognize when we get there.

Table 3 — Metrics Category Component Scoring

Grade	Numerical Score	Scoring Scale (calculated for each metrics component)
A	5.0	A for = 5.5% or better (exceeded)
B	4.0	B for = 4.5%-5.4% (expected)
C	3.0	C for = 0%-4.4%
D	2.0	D for = <0%-4.4%
F	0.0	F for = -4.5% or worse

Table 4 — BPM Composite Weighting

Component	Weight	Composite
Project schedule duration	25%	Project schedule duration score x 0.25
Project output productivity	25%	+ Project output productivity score x 0.25
Project quality	25%	+ Project quality score x 0.25
Maintenance output productivity	25%	+ Maintenance output productivity score x 0.25
Total	100%	**= Composite total**

Table 5 — BPM Composite Scoring

Grade	Numerical Score	Assessment
A	4.5-5.0	Exceeds standards
B	4.0-4.4	Meets standards
C	3.0-3.9	Does not meet standards
D	2.0-2.9	Does not meet standards
F	0.0-1.9	Does not meet standards

Table 6 — Example 1: BPM Score Calculation

Metrics Component	Baseline Values	Year 2 Values	Percent Change	Component Grade	Component Score x Weight	Composite Scoring
Project schedule duration	1.1	1.0	5% better	B	4 x 0.25	1
Project output productivity	3.6	3.4	5% better	B	4 x 0.25	1
Project quality	23.0	24.2	5% better	B	4 x 0.25	1
Maintenance output productivity	50.0	47.5	5% better	B	4 x 0.25	1
Total						**4 = B (meets standards)**

Table 7 — Example 2: BPM Score Calculation

Metrics Component	Baseline Values	Year 2 Values	Percent Change	Component Grade	Component Score x Weight	Composite Scoring
Project schedule duration	1.1	1.0	5% better	B	4 x 0.25	1
Project output productivity	3.6	3.5	3% better	C	3 x 0.25	0.75
Project quality	23.0	24.2	5% better	B	4 x 0.25	1
Maintenance output productivity	50.0	47.5	5% better	B	4 x 0.25	1
Total						**3.75 = C (does not meet standards)**

Table 8 — Example 3: BPM Score Calculation

Metrics Component	Baseline Values	Year 2 Values	Percent Change	Component Grade	Component Score x Weight	Composite Scoring
Project schedule duration	1.1	1.0	5% better	B	4 x 0.25	1
Project output productivity	3.6	3.4	6% better	A	5 x 0.25	1.25
Project quality	23.0	24.2	5% better	B	4 x 0.25	1
Maintenance output productivity	50.0	48.5	3% better	C	3 x 0.25	0.75
Total						**4 = B (meets standards)**

About the Author

Jim Mayes is currently an independent IT consultant with QSM Associates, Inc., providing a variety of quantitative software management services. Prior to this, he worked in IT at BellSouth for 27 years. Over the past seven years with BellSouth and as a consultant, he has been directly involved in software project estimation, project tracking, predictive analysis, data analysis, trending, benchmarking, process improvement, function point analysis, outsourcing, and productivity measurement. At BellSouth, he gained experience in software development and lifecycle management as a programmer, systems analyst, applications development and maintenance manager, Software Engineering Process Group member, and Software Engineering Metrics Group lead. Mr. Mayes is a certified function point specialist, a member of the IEEE Computer Society and the International Function Point Users Group, and has written numerous articles on software metrics and process improvement. He can be reached at jimmayes@bellsouth.net.