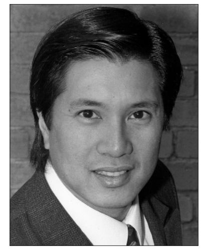
IT METRICS STRATEGIES
Helping Management Measure Software and Processes and their Business Value



IT Organization, Benchmark Thyself (Part 2)

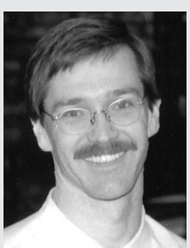
by Michael Mah

When people think of the Internet economy and the role of information technology, the one word that often comes to mind is speed. Today, competitiveness seems to be all about speed. What's perceived to be at stake is corporate survival. And with many companies experiencing IT applications backlogs ranging from seven months to two years, IT capacity needs to be ratcheted up in a hurry.

At the same time, IT expenditures are significant, with some estimates in the neighborhood of US $250 billion per year worldwide. A major telecommunications company for whom my firm recently completed a productivity benchmark study estimated that its annual IT expenditures alone were in the range of $4 billion.

With that amount of money being spent, most people indeed will want to know how productive they are. That begs the question: what does "improving productivity" mean to you? To one company, it might mean a desire to reduce that $4 billion expenditure by 30% over five years while maintaining the same level of functional output. To another, it may mean keeping a lid on spending growth, but raising output by 25% (to satisfy that applications backlog). To a third, it might mean spending more, say 10% more, but cutting time to market by a third (also to reduce the backlog). And perhaps someone else wants to increase system availability by 50%, because with every minute that a system is down, you can count the

Continued on page 2.



The High Technology Detective: Mini-Postmortems and Their Practical Applications

by James T. Heires

The use of historical data to predict future events has been the source of much debate in the software development community over the years. Opponents claim this practice is analogous to driving a car while looking only through the rearview mirror. Proponents protest that it would be foolish to purchase stock in a company without first looking at its past performance. When pressed, however, most agree that appropriate historical data can provide a clue to future events, but that it should be supplemented with additional information for best results. This story begins in 1994, when a mini-postmortem

Continued on page 10.

executive summary

This issue of ***ITMS*** begins an ascent into how to construct a practical metrics framework within your organization. The February and March issues established the goals and purposes of IT metrics; now it is appropriate to discuss the "how."

In "IT Organization, Benchmark Thyself (Part 2)," we take a look at productivity baselines. The examples are from a real company, using real data. We examine the notion of establishing your own benchmark to identify the output capacity of your organization in terms of speed to market, cost, effort, and reliability. Like an image in a photo lab, what an organization sees about itself is sometimes surprising. This is the first step in becoming a learning organization that makes better decisions.

I'm also pleased to introduce Jim Heires, who, in the second article, gives us a look into a proven process of collecting and analyzing metrics from a practitioner's point of view. This timely story from the "real world" is complete with practical advice and lessons learned that will save you time. Heires focuses on what you can do to effectively "mine" the information within your organization.

Last but not least, a familiar face returns to the pages of ***ITMS***. Carol Dekkers offers us six steps that provide good common sense in "Making Software Measurement Really Work: Aligning Measurement Expectations (Part 1)." This is the first in a series of articles designed to help you overcome cultural barriers to establishing successful measurement programs. She talks about what measurement can and cannot do, and what elements make sense to ensure that metrics programs focus on delivering value.

Michael C. Mah, Editor

april 2000 vol. VI, no. 4

CUTTER CONSORTIUM

Continued from page 1.

money lost from the missed transactions (witness the recent eBay or Schwab outages.) If you're really greedy, you might want costs to be reduced by something like 50%, schedule reduced by 30%, and reliability improved by 40%, all over a five-year time frame.

Whatever productivity means to your company, it must be articulated and understood in terms of your priorities. This means looking in the mirror to assess IT performance from a view that is framed within the context of your specific industry. What matters to a bank about IT productivity might have little relationship to what matters to a dot-com organization.

The dilemma is that, even with nothing more urgent than what's happening in IT, the availability and access to IT metrics data within most organizations is incredibly poor. Multimillion- or billion-dollar decisions are made from a misinformed state. During speeches at industry trade conferences, no one raises their hands when people like Howard Rubin, myself, and others ask who in the room has an effective metrics program in place that actively supports strategic decisionmaking. It seems that only in IT is this the case.

Often, IT organizations try to tackle this vulnerability by looking at what other people's data says; they try to make decisions about their own directions based on other organizations' experiences. Although some information is probably better than none, this tends to create a false sense of metrics security. Better to have your own productivity measures from *your* organization to make decisions about *your* fate.

Obstacles to Acquiring IT Metrics Information

There are a number of issues that create obstacles to IT data collection within an organization. This is a huge topic in and of itself, but some of the most prevalent obstacles are:

- Measurement fear
- Perceived lack of time due to deadlines (we have "real" work to do)
- Emotional negotiations being the norm: who needs numbers?
- Lack of management education on practical metrics
- Invalid approaches
- Staff churn and poor record keeping
- No established protocol for project postmortems

In his article on page 1, Jim Heires from Rockwell Collins, Inc. offers sound advice on how to overcome some of these barriers through his candid sharing of what worked within his organization. His article contains numerous prescriptions for solutions from a process standpoint. This information has been distilled from years of experiment and experience that can save you time and effort.

With data in hand, the IT metrics specialist's task is to proceed to meaningful analysis. Once you know what to look for, there are several known pitfalls that can be avoided. Moreover, how you portray your findings will be very important. If the data is misrepresented, it can lead to false conclusions and bad decisionmaking.

In a moment, I'll share with you how some firms approach metrics analysis to give you some guidance on interpreting the data and making thoughtful recommendations.

What Projects to Use

Generally speaking, it is a good idea to compile a list of IT projects that meet meaningful selection criteria. Among these will

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@qsma.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 Nields, 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. ISSN 1080-8647.

be projects with much at stake with regard to their outcome. Imagine saying to your senior executives, "What are the top 20 projects that are the most vital to you and the company?" Those are the projects you're looking for.

For now, you want to look at projects that have recently been completed, preferably within the last year or so. Projects that are in "mid-flight" will be discussed in future ***ITMS*** issues, as there is much to say about this unique kind of analysis. Right now, however, we want to concentrate on answering questions about what the organization has demonstrated on projects that have recently been finished.

The purpose of establishing this initial focus is to go where the money is, so to speak. Creating value for a metrics program means applying metrics to projects with high importance to the organization. In this manner, the "value outcome" of a metrics initiative will be coupled to IT projects with high value to the company. Your ability to positively influence what decisions are made on these projects is intended to make a true difference to your company's competitiveness and performance in the marketplace.

Framed in this manner, a metrics program is viewed as strategic, as opposed to an extra overhead activity that's misperceived as a necessary evil. In the latter scenario, the program might be at risk of being trimmed or cut during lean or problematic times. Yet if an organization "flies into turbulence," that's the most critical time to have the organization's metrics instruments well lit and fully operational. If they are not, then you're "flying" IT by sight only, without the instrument flight rule certification that could prove vital during bad weather or on a dark, foggy night.

Some additional minimum project criteria is also appropriate. Generally, you want to select projects longer than 3 months in duration that expend at least 12 work-months (1 person-year) of effort. The reason is that you want to acquire knowledge about medium to large projects that reflect the effects of team dynamics. Most projects that are smaller than this are usually not part of an organization's "top 20" (unless you're working in a very small company), and their measures usually reveal more about individual people than about process. For example, benchmarking a project that Lee worked on over a two-week period reveals more about Lee and whether he/she was a good fit for that project than about the company's IT productivity.

Old Data Versus New

You can use the same method to benchmark a series of projects finished in the last four months as you would for a singular project that finished last week. However, as time passes, institutional memory begins to fade and reconstructing what happened on a project becomes more challenging. The more time that's passed, the more likely it is that a given team has scattered to various parts of the organization, with its members now in the middle of something else.

Thus, it is important to try to gather metrics when the data is reasonably fresh. It's not impossible to conduct "what happened" conversations for projects that are six months to a year old; I've personally done it for scores of projects over the years. But it is harder, and the margin of error sometimes increases, depending on the reliability of the record keeping.

Analysis Advice

After the data has been collected and validated, the analysis is often concerned with establishing clear answers to questions. These questions should map into goals we want to establish for the organization going forward. Some of them are:

- How productive was our last batch of critical projects?
- What are the findings across different areas of our business?
- Is our IT applications development and maintenance productivity increasing or decreasing?
- Are schedules getting shorter?
- Are effort and cost decreasing?
- Is reliability improving?
- By how much is productivity increasing or decreasing and at what rate?

- How do we compare to others in our industry?
- What were overrun statistics and time/effort profiles like in the face of changing requirements? (For an example, see sidebar, "The Overruns Continue.")

When drawing conclusions from the data, it will be very important to be sensitive to organizational problems associated with language of judgment and/or evaluation. There may even be a natural tendency to compare projects and teams. The risk involved with using measurement for evaluation is that people begin to fear evaluation, which can lead to all sorts of problems, from cover-ups to quiet sabotage of the metrics program.

What companies need to do is focus on process, using the language of organizational learning. Having this body of knowledge provides opportunities for insights, learning, and consciously authoring a course of action in which problems are discovered and rectified in a more focused, efficient, and meaningful way than in an organization with a chaotic approach to problem solving. This doesn't mean an organization without measures will not be able to improve; however, it will likely do so more slowly than an organization that has more complete knowledge of itself.

For an organization to raise itself to the next level, it must master not only the mechanics of acquiring knowledge about its performance, but also the ways in which it reacts and responds to the information that is revealed as part of its measurement process. In other words: what will it choose to *do* with the information it acquires.

The first issue of the day is to be conscious about rewarding and encouraging the people who expended the time and effort to collect the metrics. If what you do with the measures is declare judgments and concentrate on evaluation rather than problem solving, you can be sure the next time you ask for measures, people will say they're too busy, and your metrics program will collapse.

Don't Play the Ratio Game

Now that you have data, you must be careful to use it wisely. Many metrics analysts fall into the trap of using overly simplistic measures — for example, measuring productivity only in functional output per unit cost. The problem with this metric is that it's a simple ratio that omits an all-important value: schedule.

I have said this before, but it is worth repeating until we all get it right: in this Internet-speed economy, time is of paramount importance. Remember that work-months are a unit of effort, not a unit of schedule. For example, 6 people working full-time on a project that is 10 months long expend 60 work-months of effort (6 people times 10 months). If you double the staff on the project, you might (I said might) shrink the schedule to 8 months, a compression of about 9 weeks. The time is now 8 months, but effort is now 96 work-months (12 times 8), an increase of 50%+. Time was compressed by about 20%. But watch out — it's not uncommon that defects in projects compressed in this manner increase sixfold. This is the 200/20/6x effect. You double your staff by 200%, you cut the schedule by a lousy 20% (if you're lucky), but defects rise about sixfold.

In both cases, output in the numerator remains constant, at say, 100,000 lines of code, or 100 function points, or 500 C++ objects, while the denominator increases by 50%. That makes the productivity ratio fall dramatically, when all that happened is more people were put on the team. In the case of 12 people versus 6, it would look like productivity fell by 33%, when in fact the project was delivered 9 weeks faster — the perceptions conflict.

A Graphical Analysis

Instead of doing numeric ratio analysis, I suggest building your own graphical (visual-oriented) trends. Here's where we expand on the idea of the growth charts I used for illustrative purposes in last month's ***ITMS***. Figure 1 shows a trend for an IT organization's schedule performance across projects ranging from small to large. This is a large

The Overruns Continue

QSM Associates recently examined overrun and slippage characteristics from completed IT projects (both new developments and enhancements) collected during several productivity benchmark consulting engagements. The sample consisted of 210 projects from 24 companies across several industries, including utilities, financial, retail, and equipment manufacturers. All of the projects were completed during the three-year period from 1997 to 1999. Here are some of the results:

- Of the 210 projects, 98 (47% of the sample) were not able to report their originally planned schedules and/or budgeted effort because the plan had changed so many times.
- Out of the 210, 83 (39.5%) were delivered late by an average of 66.7%. A small percentage of projects (7.6%) were delivered early by an average of 16.8%. Another small percentage of the projects (6.2%) were on time.
- Out of the 210, 63 (30%) were over budget by an average of 127%, 49 (23.3%) were under budget by 22.1%, and 23 (11%) were on budget.

Below are the average values and general characteristics for schedule, effort, and staffing for this sample.

Schedule

- Requirements: 4.6 calendar months
- Main build: 6.6 calendar months

Effort

- Requirements: 32.1 work-months
- Main build: 91.0 work-months

Staffing

- Requirements: 5.5 full-time equivalent (FTE) staff
- Main build: 11.0 FTE staff
- Staff churn in construction averaged 10%

Size

- 85,065 new and modified instructions
- 56% of sample sized in function points and/or source code counts

Languages

- 49% used Cobol, C++, and C as their primary development language
- 51% of sample was split among 31 languages, including PeopleSoft, Visual Basic, SAP, Oracle, Java, 4GLs, and code generators

Requirements

- Requirements volatility averaged 22.2%
- Scope creep averaged 13.4%

Reliability

On average, 120 errors were found and fixed during testing:

- 40% of errors had a serious or critical severity ranking
- First 30 days postproduction: average "up time" was 13.9 days

(Note: the main build phase encompasses detailed design, construction, and all testing.)

Observations

It is interesting to note that a large number of projects (nearly half) had lost track of their originally planned schedule and budget. This fact speaks of the extreme volatility that IT application and development teams are dealing with in the course of their work.

It is therefore not surprising that overruns and slippages are running at 127% and 67%, respectively. Although these numbers are not as severe as reported in other studies, they are nevertheless quite significant. It should be noted however, that some projects were delivered under budget by 22% and ahead of schedule by 17%.

Finally, performance should also be interpreted in the context of continuously changing requirements while the imposed deadlines were kept fixed. It seems obvious that, in light of scope changes and scope growth, the teams performed admirably. These are circumstances that (sadly) reflect the norm of our volatile business environment. Let's give credit to our IT staffs as they continue to operate "under the gun"!

— Michael Mah

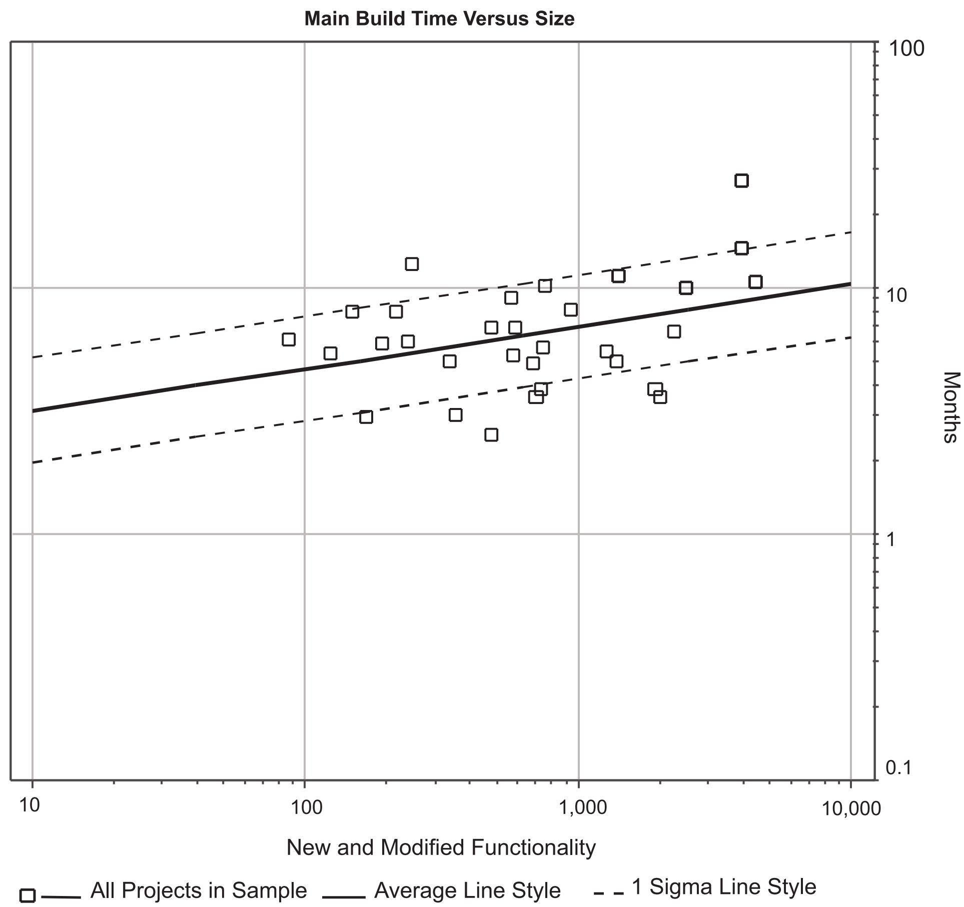


Figure 1 — Speed performance: 31 completed projects.

IT organization from a US-based telecommunications company that employed about 2,000 people in its IT department at the time. The data was part of a benchmark study that was conducted to establish a "Year 1" productivity baseline to support a major process-improvement initiative.

The chart shows speed on the vertical axis (in months) as a function of size in new and modified functionality. Each project from the data sample is represented by a single dot on the plot. It represents the elapsed schedule from detailed design through deployment of the system. If a project that comprised 100,000 lines of new and modified source instructions, 1,000 function points, and/or 500 C++ objects took 8 months, it would plot at 1,000 function points and 8 months.

The dot on the plot (remind you of a Dr. Seuss book?) also has a companion plot position on other trends. These include trends for effort versus size, defects versus size, and number of full-time equivalent staff versus size. Together, each family of values tells a story about that individual project, while the aggregate tells a story about the organization.

The center line is the average schedule trend for this company drawn through its data. Upper and lower bound lines are above and below the center. This delineates the plus and minus 1 standard deviation of the data. That simply means that projects on the lower bound are in the 84th percentile. Projects that hug the upper bound are in the 16th percentile (they take longer).

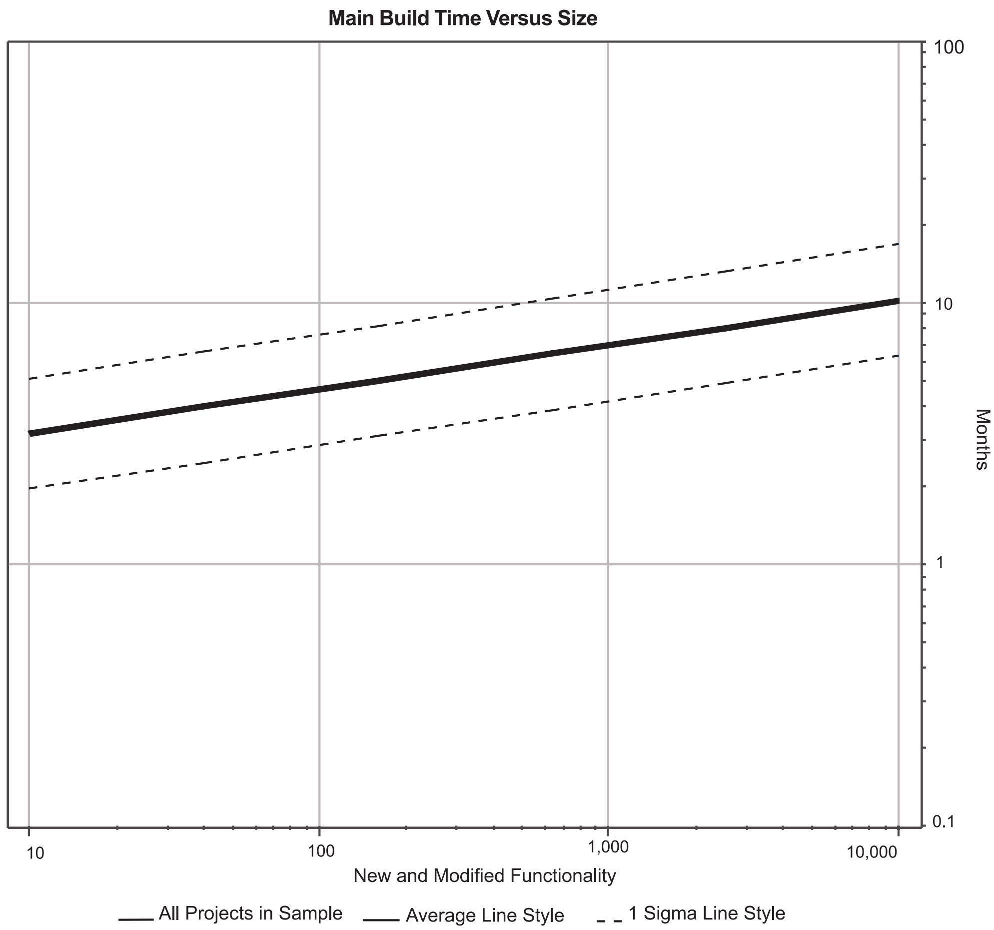


Figure 2 — Company ABC: baseline schedule trend.

The chart in Figure 2 shows this organization's baseline trend *without* the individual dots that represent each project. What we are left with is the organization's Year 1 baseline schedule trend. The slope of the line rises from left to right. This reveals that larger projects take progressively more time and indicates by how much. Although it looks linear, the graph is actually somewhat deceptive. Logarithmic scales mean that, if drawn on linear axes, the trend would show curves — much like the National Center for Health Statistics (NCHS) pediatric growth charts in the March issue of ***ITMS***.

Figure 3 shows four sets of trends placed on one view. They show trends for schedule, effort, staff size, and defects found and fixed during testing, all as a function of small to large projects. Figure 4 shows the same trend again without the individual projects, just the aggregate trend. Taken collectively, these baselines represent the IT "throughput" capacity of the organization.

What It All Means

What I've depicted here is a sample framework that moves away from "traditional" IT metrics analysis (dependent on elementary ratios) in favor of a visual depiction, showing the multiple dimensions of metrics. This type of visual tells a more coherent, holistic-oriented story.

The implications of such a framework are significant. This method allows an organization to "become its own NCHS" and to produce its own productivity baselines. The