What You Don't Know Can Hurt You

Douglas A. Ebert
McKesson Corporation

This article provides senior managers with a methodology to develop a metrics program that will form a basis for management decisions. It presents a series of questions a senior manager should ask to address business needs, rather than just getting informational briefings.

There is an old accounting adage that states, "You can't manage what you can't measure." In the development world we certainly take this to heart. Walk into any boardroom of any information systems development group and you will find remnants of charts, briefings, status reports, and analyses of all types – and yet, in the 2003 version of the Standish Group's "Chaos Chronicles" [1], only 34 percent of all information technology projects are labeled successful!

Maybe we just do not have enough metrics. Some time ago I had a discussion with a project manager working on a large development project, the manager estimated that he and his subordinate team leaders spend around 30 percent of their time preparing reports aimed at higher levels within the company. These reports culminate in monthly senior stakeholder meetings attended by the highest-ranking members of the business, yet 15 months into the project, the manager cannot recall any directions or decisions coming from these meetings. This void is in spite of the fact that the project will come in over a year late and that early deliverables have received unfavorable comments about their quality!

As senior managers, we have gone astray somewhere in our metrics programs. It is not enough just to be fed with data if that data is not able to drive decisions. We could put the old accounting adage another way: "You don't measure because you want to know, you measure because you want to be in control!"

Standard Metrics Programs Quickly Become Stale

Most of us have a standard set of project reporting templates that we expect our development project managers to fill out periodically. These reporting systems evolve over time because senior leaders really do have a need to know what is going on. The problem is that without any more thought than this, managers default to asking for metrics that are readily available. This includes things like work hours or resources spent to date, defects uncovered so far, and the proximity to being on budget. Worse, without any insight into what we think this data shows, our project managers obediently and simply provide that data.

While perhaps interesting, this kind of information does not provide the senior manager with any basis of understanding about whether the project really is going well or poorly. Nor does it indicate if there is anything he or she can (or even needs to be able to) do about it. Thus, project reviews become informational briefings, not periodic checkups. Watts S. Humphrey says it best:

"*As senior managers, we have gone astray somewhere in our metrics programs. It is not enough just to be fed with data if that data is not able to drive decisions.*"

> By concentrating on the schedule, managers overlook the things they can influence. These are to start jobs promptly, provide adequate staffing, and ensure that the work is done in a disciplined and professional way. [2]

What we need is a program that will link the requested metrics to business objectives and provide "objective results that can be used in making informed decisions, and taking appropriate corrective actions" [3]. As an important part of this program, our project managers need to know why we think we need this data. Knowing what we think we are getting, product managers will be able to suggest their own methods and reporting tools. This helps build a consistent approach and provides a basis for mentoring as well. Building a set of metrics consists of asking the following five basic questions.

1. What Do You Need To Know?

This is not a trivial question and is an essential step since it forms the basis for the entire metrics program. Rather than treating every project alike, take the time to sit down with your project manager to define information needs based not only on the specific nuances of the project, but also on the need to be alerted if decisions are required.

For example, it is unwise for a senior leader to fall into the trap of making a simple needs statement like, "I need to know if we are on schedule." A better needs statement would be something like, "I need to know that I'll stay on schedule." The difference in these two statements is not that subtle. The first statement is a simple snapshot that may let the senior manager sleep that night, but does not tell him whether he should cancel his upcoming vacation time. More to the point, if the schedule is slipping, the senior leader has no basis to understand why it is slipping and what can be done about it.

Then again, each project is a proving ground to see if you learned any lessons from the last project. As a senior manager, you always have a need to know that with each project you are getting better at something – whether that something is better efficiency, productivity, or just plain quality. With each project, target metrics that will help evaluate improvement measures, either to establish a baseline or assess improvements. Need statements for this category could look like this: "I need to know that my quality improvements are meeting the objective," or, "I need to know the volatility of requirements during each stage of development."

It is interesting that when I was a project manager, my senior managers would regularly ask me, "What are you doing differently?" But rarely would they demand that I demonstrate my changes had any effect! It seemed just causing the disruption associated by introducing new processes met their expectations.

2. What Questions Do You Need to Ask?

For each of your need-to-know items, you have to design the questions correctly to get useful information – this is not as simple as it might seem. Again, many senior managers ask for simple data points such as the current project master timeline. The project master timeline is data; being able to demonstrate that the project kept to that timeline on purpose rather than by accident requires analysis.

For example, let us take the following case: "I need to know that I'll stay on schedule." One question you might want to ask would be, "What is my burn rate of resources for each phase/task of a project compared to my estimates?" It should be noted that to answer this question, a project manager will still need to present the traditional data points of milestones and resources consumed in a period of time.

Be watchful for events inside the project that might create problems later. This not only includes scope creep, where new requirements are being added, but also scope volatility. A programming effort in which I was involved was right on schedule for the first eight months, although during the last month a change in direction caused changes in almost 15 percent of the requirements. The needed rework did not really show up as delays for another two months and came as a complete surprise to senior managers.

Here is a hint: One good place to look for questions is to challenge planning assumptions. Assumptions made during program planning are excellent informational needs for the measurement process. If the assumption is not realized, then many of the resulting schedule and resource plans may need to be re-examined, or replanned. [4]

Simplistically, if the project plan assumes 70 percent availability of staff, that becomes a key metric to predict the final project outcome.

3. Why Do You Think This Is the Right Question?

This is the introspective part: Challenge yourself about each question you believe you would want to ask. This requires you to think through to the *so what* of whatever metrics you may receive – to think through what you might do with the data and what management actions you might take. This step also prepares you to defend your demand for these metrics.

For example, you may explain your rationale as this: "Knowing the resource burn rate compared to estimates will give me a glimpse into the solidness of the project plan. If our estimates are consistently low, I may have to reset project expectations or assign additional resources to your team."

> *"One good place to look for questions is to challenge planning assumptions. Assumptions made during program planning are excellent informational needs for the measurement process."*

Another example, if your need-to-know is that we have been meeting requirements of our quality program. A need-to-ask question could be: "What is the trend of SQA [software quality assurance] violations, by type?" Why would you want to know this? Humphrey says it best: "When software projects fail, it is usually because a manager did not insist that the work be done in the right way" [2].

Here is another example: "How do the hours expended to address customer-related problems compare to my budget?" Why would you want to know this? Well, unless you have the luxury of a separate, dedicated support group, surges in customer-related efforts often rob your project of essential resources. Remember, if you wanted to have a good feeling that you would stay on target, this is a question you may need to ask.

You may wish to discuss your rationale with your managers. This will give them insight to your needs and concerns and allow them to provide their own input. Do not misunderstand, though; this is not a voting situation. Just because there is no current process to answer your questions does not mean you should not get the answers.

4. What Measures Would Provide Input to Management Actions?

In most cases, you may already have some idea of the questions you need to ask. Stop and refine those questions to ensure you have enough detail to take action. For instance, knowing about SQA violations will give you an overall feeling about how well your process programs are being followed. However, having these violations reported per project step or phase may highlight where your retraining efforts may provide the biggest impact.

This polishing question could also be asked like this: "Before I take action to correct a problem, what other information would I need?" If you knew the distribution of defects per module, reported by severity, that would be more useful in a project review than just hearing something like, "We're sure seeing a lot of problems!"

This is the time to look for collaborative or explanatory evidence. In this last example, it would be interesting to compare SQA violations to defects per module as you would expect to see a correlation between the two. However, if SQA violations are not accompanied by defects, it could mean your SQA processes are a little overzealous! On one project, a rash of SQA violations was reported during inspections of requirements specifications. Upon further investigation, we discovered that the documents associated with the requirements program had recently been automated, yet the SQA checklist still required physical signatures by both author and checker.

5. What Is My Objective for Each Metric?

Put another way, this last question sets an acceptable range of values for each metric. It is important to set up each of these trip points ahead of time. Do not wait to get a report before you have to ask the question, "Should I be worried about this?"

For example, you may wish to call a more in-depth schedule review if the resources allocated are greater than 10 percent of the estimated burn rate. Or, perhaps you have set a goal of decreasing SQA violations in the planning phase by 20 percent. You must choose the acceptable range of values based on specific circumstances of the project, including the broader standards that may be established by your company and the maturity of your project members.

This last question also provides a kind of acid test. If there is no unacceptable value – if no actions are forthcoming regardless of the report – then this is data, and not a metric.

Final Words of Advice

Here are some final suggestions in building a set of metrics:

- Avoid the simple answers – they may be good data points but they are generally not helpful.
- Metrics can have a life of their own so avoid creating a monster! Start small and grow with time. Remember that collecting metrics consumes project resources.
- Make establishing a baseline a priority if you do not have one already.
- Listen after you ask questions. This encourages participation and mutual respect.
- Remember, the primary purpose of a metrics program is to support change! (As a corollary; metrics programs make very poor clubs.)
- Push the comfort zone. You need to know what you need to manage, not what information somebody is comfortable telling you.◆

References

1. The Standish Group International, Inc. CHAOS Chronicles. Vers. 3.0. West Yarmouth, MA.: The Standish Group, 2003 <www.standishgroup.com/chaos/toc/php>.
2. Humphrey, Watts S. Winning With Software. Boston, MA: Pearson Education, 2002.
3. Software Engineering Institute. Capability Maturity Model® Integration (CMMI®), Vers. 1.1. Pittsburgh, PA: Carnegie Mellon University, 2001.
4. Baxter, Peter. "Focusing Measurement on Managers' Informational Needs." CROSSTALK July 2002: 22-25.

About the Author

Douglas A. Ebert completed a U.S. Air Force career as a lieutenant colonel and is currently a vice president for strategic planning and partnerships with McKesson Corporation. In this capacity, he leads the adoption of emerging industry technologies and standardization efforts in developing information solutions for the healthcare industry. A speaker at technology conferences, he is also the chair of the Software Engineering Institute's Capability Maturity Model Integration® Interpretive Guidance Expert Group.

McKesson Corporation
1400 S. Wolf RD STE 200
Wheeling, IL 60090
Phone: (847) 495-1718
Fax: (847) 537-4866
E-mail: doug.ebert@mckesson.com

CALL FOR ARTICLES

If your experience or research has produced information that could be useful to others, CROSSTALK can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for three areas of emphasis we are looking for:

Risk Management
February 2005
Submission Deadline: September 20, 2004

Personal Computing
March 2005
Submission Deadline: October 18, 2004

Personal Software Process/
Team Software Process
April 2005
Submission Deadline: November 15, 2004

Please follow the Author Guidelines for CROSSTALK, available on the Internet at <www.stsc.hill.af.mil/crosstalk>. We accept article submissions on all software-related topics at any time, along with Letters to the Editor and BackTalk.

COMING EVENTS

October 2-9
Internet, Processing, Systems, and Interdisciplinary Research (IPSI) 2004
Sveti Stefan, Montenegro
http://montenegro.internetconferences.net

October 3-6
2004 IEEE Custom Integrated Circuits Conference CICC 2004
Orlando, FL
http://www.ieee-cicc.org

October 4-6
14th International Conference on Software Quality
Orlando, FL
http://software.asqquality.org/events.htm

October 6-7
IT Research Associates' 3rd Annual Six Sigma for Software Development
San Francisco, CA
http://www.frallc.com/infotech.asp

October 9-13
11th International Conference on Architectural Support for Programming Languages and Operating Systems
Boston, MA
http://www.eecg.toronto.edu/asplos2004

October 24-27
17th Annual ACM Symposium on User Interface Software and Technology
Santa Fe, NM
http://www.acm.org/uist

October 31-November 6
Association for Computing Machinery SIGSOFT 2004
Newport Beach, CA
http://www.isr.uci.edu/FSE-12/venue.html

April 18-21, 2005
2005 Systems and Software Technology Conference
Salt Lake City, UT
www.stc-online.org