

IT METRICS STRATEGIES

Helping Management Measure Software and Processes and their Business Value



Sizing Up Your Promises and Expectations

by Michael Mah

In previous issues of *ITMS*, I tackled the subject of managing Internet-speed deadlines (see *ITMS*, May and June 2000). Specifically, what do you do when IT projects are given a deadline first, before you know the full scope of the requirements? Also, how do IT applications development and maintenance projects fundamentally behave — what are the “laws of nature?” How can that knowledge help us estimate projects better?

Well, since then, I have continued to take audience surveys about current practices. I have conducted these similar queries at several recent speaking engagements.

I ask, “How many of you are given the deadline first? How many of you have at least one-half, one-third, or one-quarter of the requirements well established when you’re tasked with starting the build phase of your project?” In nearly all the cases, the answers are frighteningly the same: a fixed deadline, for what seems like an impossibly short schedule, and undefined scope.

The Problem with Overpromises

The problem with this scenario is that project teams invariably slide into a dynamic where the deadline is firmly solidified, but the promised scope becomes wildly excessive, resulting in disaster. Without reliable

Continued on page 2.

executive summary

This issue continues to offer practical advice on how to use metrics to make better decisions.

First, I continue to bang away at the topic of managing risk in my article “Sizing Up Your Promises and Expectations.” When we examine IT project overrun statistics, we see that the small percentage of projects that manage to make both cost and schedule do so by the only method known to mankind when the deadline looms like a freight train — they cut function to the core. I take a look at how people get into that sordid mess in the first place. It begins with the initial overpromise, either through tacit allowance of scope growth or by blind over commitment on what the team takes on at the onset. In this space, we have to talk about the issue of “sizing” software IT projects.

Next, I’m pleased to introduce a new author to *ITMS*: Arlene Minkiewicz, chief scientist at Price Systems LLC. Her article, “Practical Software Measures Offer Solutions to Process Challenges,” talks about the priorities of successful measurement programs. She articulates how important it is to understand the problems that you’re trying to solve and to let that guide you in deciding what to do first. This is about aligning your measurement program to get the most out of your efforts.

Then, Stan Rifkin of Master Systems gives great advice in “How to Select Software Project Macro-Estimation Tools.” He offers his personal checklist that can help you get to the heart of the matter when looking for automated models to help with that all-important “scenario analysis” when it comes to decision time on critical IT projects.

All in all, this trio of topics is designed to help you get down to measuring what matters most and to use metrics in a meaningful way.

Michael C. Mah, Editor

september 2000 vol. VI, no. 9

Sizing Up Your Promises and Expectations.	1
Practical Software Measures Offer Solutions to Process Challenges.	1
How to Select Software Project Macro-Estimation Tools.	13



Practical Software Measures Offer Solutions to Process Challenges

by Arlene Minkiewicz

A good measurement program is one of the cornerstones of any successful process-improvement program. In fact, measurement is essential if you want to be able to quantitatively identify improvements. But knowing this doesn’t help when you find yourself responsible for the implementation of a measurement program in your organization. There are just a few questions you want answered. Where do you start? What are the right things to measure? Why me?

Successful measurement programs generally don’t try to accomplish too much at one time. It is important to understand the problems you are trying to solve with your measurement program and select the measures that best meet your goals. It is best not to try to solve all of

Continued on page 9.

Continued from page 1.

software-estimation capabilities, organizations badly overpromise. It seems to be in our very nature. This seems to happen in both inhouse development and in many situations where IT is outsourced.

Even if the initial promise is realistic, there's another problem — scope growth, or requirements creep. Along these lines, another question in my audience survey is, "How many of you know when to say 'no' as the requirements and scope grow on your projects? For example, when do you know that the number of work requests has crossed the line into an 'impossible zone' for a project with a fixed deadline?" Very few hands go up.

I contend that as long as people struggle with these issues, we will see an elongation of project schedules. To staff up on a project with a given deadline, overpromise on what will be done, suffer the inevitable slips, panic about the end date, and then pull functionality out at the last minute (to the "core") to make the schedule, destroys process productivity. That path costs more, takes longer, and produces more defects. In some cases, it's not uncommon for a 10-month schedule to become 15 months, and for the effort and costs to double as staff is ramped up during the chaos of the final months.

Our Competitiveness in the Marketplace Demands a New Core Competency

What is an IT manager to do? The walls of IT applications backlogs combined with the need to accelerate schedules creates a doubly painful situation where the reflexive response is to overpromise.

For many organizations, this dynamic is so pervasive that it is by far the number one killer of process productivity, resulting in projects being late and over budget, with poor quality. There comes a point for some

where the effects are so extreme that to focus attention on other issues affecting IT productivity is to pursue an abstraction. Other issues affecting productivity become almost irrelevant!

In fact, I can recall one project that my team benchmarked several years ago. The project size was more than 4 million lines of code. Productivity was phenomenal and reflective of the technology that was employed by the team. Ironically, this actually contributed to the problem of scope growth and churn. Why? Because management felt that it had a free ticket to do so because of how efficiently the code was able to be produced.

In the end, the project failed because the requirements grew wildly out of control. Consortium members fought over scope and lost sight of common interests, pulling the project in five different directions. The project crashed and burned, and more than US \$100 million was written off in losses. Management never deployed the system.

The size of the project was grossly underestimated at first and then ballooned into a monster. Ed Yourdon summed it up well: "If you underestimate the size of your project, common sense says that it doesn't matter which methodology you use, what tools you buy, or even what programmers you assign to the job."

That demands a new core competency for many IT organizations: managing scope; managing your promises; your commitments; the growth of the promise; the churn; the size of the amount of software that you commit to deliver. The rapid change in the marketplace demands that you handle these dynamics well. If you don't, you can be assured of bad outcomes. If you become great at handling this, you'll have a critical edge, and your productivity and creativity will show.

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 Mullaney, Tel: +1 781 641 5101, E-mail: amullaney@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.

