

What Makes Measuring Software So Hard?

Stan Rifkin, *Master Systems*

We often hear that it is difficult to get software measurement into practice. Traditional measurement addresses the decisions that support increased quality, increased programmer productivity, and reduced costs—key elements for organizations strategically focused on operational excellence. But what if the organization's highest priority isn't operational excellence? This article shows that such organizations have different measurement needs and presents ideas on how to

address those needs—thereby making measurement more appealing.

While the disparity discussed here involves measurement, it applies to all areas of software process improvement. For example, the Software Engineering Institute's Capability Maturity Model for Software is silent on two of the three strategies of high-performing organizations—customer intimacy and product innovation. Like traditional measurement, the Capability Maturity Model applies only to organizations wanting to be operationally excellent.

How top organizations do it

Michael Treacy and Fred Wiersema, in *The Discipline of Market Leaders*, conducted a survey of 80 top organizations to find out how they out-achieved their competitors.¹ Their survey revealed that these companies needed to focus on only one of

three market disciplines—operational excellence, customer intimacy, or product innovativeness—to succeed.

Operationally excellent organizations have a formula for their service or product. They offer a small and limited menu of choices, but they deliver excellently and at a competitive price. McDonald's and Federal Express focus on operational excellence.

Customer-intimate organizations seek a different market niche—a total solution.² Whatever the customer wants they add to the menu. These organizations offer a long, custom-made menu for each engagement. Some financial-service institutions might call customer intimacy a way of getting a greater share of the customer's wallet, offering spending alternatives ranging from investment opportunities to travel services.

Product-innovative organizations pride themselves on maximizing the number of

Implementing software measurement can be difficult if an organization's market strategy fails to align with traditional measurements. The author points out the need to understand an organization's strategic objectives and then tailor its measurement practices to fit.

We need to see that the potential rejection of our measurement efforts might be an appropriate response to measures that do not fit their strategy.

turns they get in the market. They introduce many new products, selling innovation and features as opposed to, say, price. Intel, 3M, Sony, and Bell Labs focus on product innovation. They measure their success by the number of new product introductions, the number of patents, and the number of Nobel prizes.

Treacy and Wiersema point out that a successful organization must have threshold characteristics of all three disciplines, even though it focuses on and excels at only one. They cite IBM as a company that at one time didn't have the right balance of the three. IBM focused on customer intimacy and failed to pay attention to price (operational excellence). Competitors not as strong in customer intimacy gained inroads to IBM customers through lower prices.

Traditional measurement falls short

Traditional software measurements—those espoused by the Software Engineering Institute and Quantitative Software Management, for example—apply almost exclusively to organizations focusing on operational excellence. They typically have little to offer customer-intimate and product-innovative firms.

The IT or IS departments in many software development organizations strive for customer intimacy and will do virtually anything their clients request. They get to know their clients very well, sometimes better than the clients know themselves. For example, a payroll service that has seen every variation of payroll processing ends up knowing more about the process than an in-house payroll department does. It could take over its customers' payroll departments.

Microsoft's focus is product innovation. It touts its new, glitzy features, not its up time or reliability. It wants to earn and own its clients based on new features, not by offering operationally excellent software.

Measurement and strategy: four cases

What we are missing is a more global view, one that listens to and responds to our measurement customers. We need to see that the potential rejection of our measurement efforts might be an *appropriate* response to measures that do not fit their strategy. We need to problem-solve jointly with our

clients to develop new classes of measures that simultaneously meet our high standards for objectiveness and their high standards for relevance. Now let's look at four cases where we've tried to implement this view.

Wall Street brokerage house

In one project, I worked with a prominent Wall Street brokerage house to develop new classes of measures. Their mergers and acquisitions people were interested not in software costs or quality, but in time to market. During the frantic time they were putting a deal together (such as an initial public offering), they needed the IT people to respond quickly so that the firm could earn as much as possible by offering as many services as they could. It was a question of wallet share—a customer-intimate approach.

We recognized that the traditional measures of schedule and budget variance would be meaningless in such a context. We agreed on a measure of the percentage of the total deal that did not go to the brokerage. The IT department then offered a realistic plan for continually reducing that (missed wallet share) figure. By clearly focusing on the wallet share, we rejuvenated the almost-dead software process improvement program in the mergers and acquisitions department, and we put new life into the corporate measurement program.

Civilian government agency

In another case, my client—a computer-oriented defense contractor—was developing software for a government civilian agency. The agency wanted project measures, but the agency's projects were not managed in the traditional way and therefore could not be measured traditionally. The government agency wanted the contractor to do what it requested, not study the request and offer alternatives or pushback. The agency was not as focused on cost, quality, or duration as it was on getting what it wanted in reasonable terms. This customer-intimate approach made the menu of services as long as the customer requests. Naturally, the contractor had to deliver the systems within a threshold value of cost, quality, and duration, but customer responsiveness was the primary focus.

Because the government agency often

changed its mind about requirements, rendering previous work inapplicable, the contractor faced rework that negatively affected cost and schedule compliance. The agency's audit department noticed that the part of the agency that ordered software and paid for it was not taking the contractor to task for missing deadlines and budget estimates. The contractor had to respond to the agency's audit department, but knew it couldn't use traditional project measurements.

I recommended several measures:

- of the total spent by the customer, how much went to competitors (to be minimized);
- time spent in adversarial encounters (to be minimized);
- time spent with the customer understanding its business (to be maximized); and
- the number of people on staff with credentials like the client's (to be maximized).

This set placed traditional project measurement on a second tier, managed by a separate office, and focused day-to-day programming on customer intimacy. The company implemented these measures immediately because of their high face value.

Computer services contractor

Another case involved a computer services firm that provided a civilian government agency with the computer programming and operations for a particular payment made to qualified applicants. Because the contract was up for renewal, the computer services firm wanted to propose adopting a set of measures that would indicate the firm's operational excellence. The contractor and the agency failed to come up with measures that resonated with the programming and operations staff, even though the measures came from the Practical Software Measurement Support Center (www.psmc.com).

It turned out that the government agency was really focusing on product innovation. It felt behind the times in terms of technology and really wanted a new, modern IT provider, not a better, cheaper, faster provider of old technology. In fact, there was no business driver for the desire for more modern technology, only a (vague) belief that such technology would reap financial benefits to the government in terms of potentially lower

costs and greater flexibility. The measures we agreed on were

- planned versus actual implementation of a set of new technology introductions;
- hours spent training the government client on the principles of that new technology; and
- reliability measures directly related to the government organization's business, such as the cost of government rework due to provider payment errors, idle government worker hours due to system downtime, and government time spent in meetings or on the phone with qualified applicants due to provider service failures.

These measures replaced the traditional measures previously used, such as percent of system availability, data entry error rates, and a threshold number of abnormal operational terminations per day. Neither the contractor nor the government agency had heeded the traditional measures, which did not relate to the government mission or daily reality. The computer services firm implemented the new measures as fast as they were communicated. Again, they had high face value.

The nonprofit world

Creating software measurement that aligns with an organization's primary value proposition applies to nonprofit organizations as well. For example, the United Way of America has adopted product innovativeness as its primary value proposition because it believes that product innovation is the only sustainable strategy for a national charity.

Measuring according to the right strategy

Clearly, market disciplines other than operational excellence have different process and measurement needs. Here are a few ideas from my practice about how to address those needs.

Customer intimacy

Customer-intimate organizations seek flexibility so that they can extend their menus infinitely. To align software measurement with customer intimacy, we must measure flexibility and wallet share. For ex-

Clearly, market disciplines other than operational excellence have different process and measurement needs.

We need to develop a whole new set of measures for all those customer-intimate and product-innovative organizations that have avoided measurement thus far.

ample, in peer reviews we need to closely examine the elements that limit future options, such as a limit on the number of items in a list and built-in, hard-coded “magic” numbers. Also, we need to judge comprehension during reviews because the artifacts will constantly be expanded and enlarged as a strategy and must be understandable. Accordingly, we would seek measures of comprehension and understandability as surrogates for maintainability and changeability.

We should measure configuration management for customer-intimate organizations by the number of interfaces they manage. After all, we seek a plug-and-play architecture where we can remove and replace components, worrying only that the components obey the interface requirements. Probably the most important ingredient of a customer-intimate system is a systems architecture, so a simple, appropriate measure would be counts of architecture checks and violations.

Product innovativeness

Product-innovative organizations concentrate on features over quality, reliability, cost, and flexibility (unless those are the specific features being optimized, which is rare). Users of innovative products have a certain patience required with new products, such as the Palm Pilot, Walkman, Watchman, wearable cell phone, Linux, and Windows 2000.

Innovative organizations often let traditional planning fall by the wayside. They value innovation as more important than planning; plans are not a deliverable. Their planning often takes the form of creating a diversity of investment alternatives, assuming that some “bets” will fail to pay off. We see this particularly in pharmaceutical firms. Their management doesn’t require researchers to discover a particular drug by a particular deadline, but focuses on regular discoveries in the pipeline and, on balance, a healthy proportion of winners.

The challenge here for those of us who care about process is to create lightweight, generic processes that can be applied with large helpings of intelligence and judgment.³

As much as those of us with a process focus might hate to hear it, innovative organizations require only “good enough” quality. Features, not quality, are the deliverable. Therefore, quality goals should focus on

thresholds, benchmarks, and especially time to market. Our measure here should be comparative: how does our quality stack up against those we view as competing for our market share?

Companies with product-innovative (or customer-intimate) strategies are organized differently than those with an operational-excellence strategy. Product-innovative organizations have high differentiation (meaning many experts) and high integration (getting disparate, possibly competing experts to serve in the interests of a common, corporate goal). One measure I use is a count or proportion of the number of people in the organization whose job is to integrate those competing interests to make a product happen. In Microsoft’s applications area (office and programming-language products), such people head 10-person teams, so both the count and ratio are high relative to customer-intimate and operationally excellent firms.

GQM to the rescue?

Those of us in the measurement game know the Goal/Question/Metric process well and use it regularly.^{4,5} GQM is a method of cascading from business goals to decisions needing information, to determining what to measure to supply that information. Because GQM flows down from goals, we might assume that it reflects the choice of market disciplines. In fact it does, provided that the GQM practitioner knows the three market disciplines and probes the management about them. Too often the GQM practitioner listens to management and is a faithful scribe but fails to question the focus, balance, and alignment of goals with strategy. I can find no reference to GQM that indicates a thorough examination of organizational strategy; business goals yes, strategy no.

So where does this leave us? Measuring operational excellence is more or less a solved problem. We need to develop a whole new set of measures for all those customer-intimate and product-innovative organizations that have avoided measurement thus far. When we do, implementing software measurement will be as easy as implementing strategy. In organiza-

tions where there is a fit between measurement and strategy, we will be able to implement measurement programs without people hating us. ☺

Acknowledgments

I learned most of this by working with John Title of Computer Sciences Corporation. The measurement leader who made me ask myself many of these questions is David Card, whose measurement leadership I have always appreciated. I gratefully acknowledge audiences at the 1999 NASA SEL Software Engineering Workshop and 2000 SEI Software Engineering Process Group National Meeting for their feedback, to David Blaine, and especially to *IEEE Software's* editor in chief and anonymous reviewers.

References

1. M. Treacy and F. Wiersema, *The Discipline of Market Leaders: Choose Your Customers, Narrow Your Focus, Dominate Your Market*, Addison-Wesley, Reading, Mass., 1995.
2. F. Wiersema, *Customer Intimacy: Pick Your Partners, Shape Your Culture, Win Together*, Knowledge Exchange, Santa Monica, Calif., 1996.
3. M. Fowler, "Resources to Slim Down Your Software Process," *Software Development*, vol. 8, no. 12, Dec. 2000, p. 34.
4. R. van Solingen and E. Berghout, *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*, McGraw-Hill, London, 1999.
5. R. Park, W. Goethert, and W. Florac, *Goal-Driven Software Measurement—A Guidebook*, tech. report CMU/SEI-96-HB-002, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, 1996; www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html (current 12 Apr. 2001).

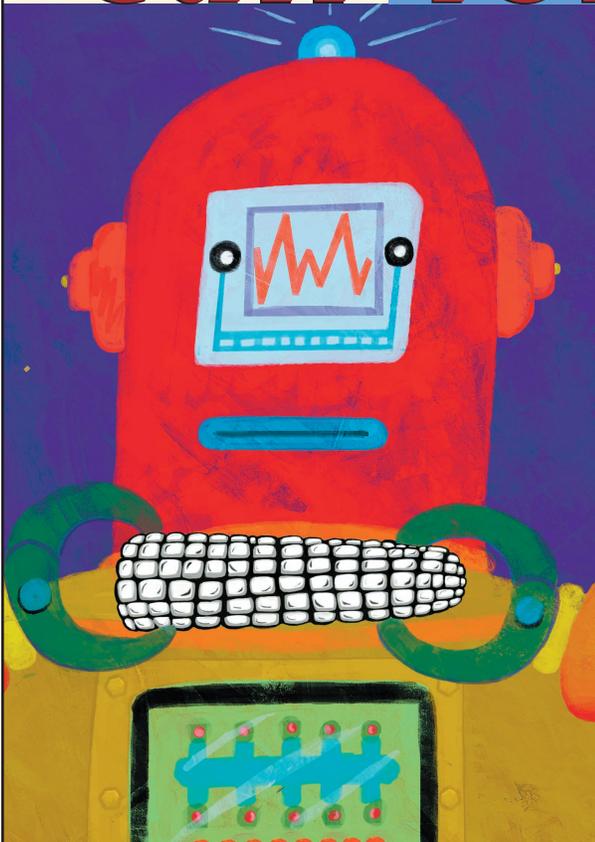
About the Author



Stan Rifkin is a principal with Master Systems, an advisory services firm that specializes in helping organizations for whom computing is strategic. He worked at the Software Engineering Institute on implementing software process improvement and is co-chair of the 2002 Software Engineering Process Group Conference. He was the chief information officer at the American Association for the Advancement of Science and the head of systems development at the National Headquarters of the American Red Cross. He has a BS in business administration, an MS in computer science, and is completing a doctorate in education. He is a member of the IEEE Computer Society, the ACM, the Foundation for the Empirical Study of Programmers, the Academy of Management, the Institute for Operations Research and Management Science, the Project Management Institute, and the Future Search Network. He is on the editorial board of *Empirical Software Engineering*. Contact him at sr@master-systems.com.

2001

Call for Papers



IEEE Intelligent Systems seeks papers on all aspects of artificial intelligence, focusing on the development of the latest research into practical, fielded applications. Papers should range from 3,000 to 7,500 words, including figures, which each count as 250 words.

Submit one double-spaced copy and a cover letter or e-mail to

**Magazine Assistant
IEEE Intelligent Systems
10662 Los Vaqueros Circle
PO Box 3014**

Los Alamitos, CA 90720-1314

phone +1 714 821 8380; fax +1 714 821 4010
isystems@computer.org

For author guidelines, see
<http://computer.org/intelligent/author.htm>

**IEEE
Intelligent
SYSTEMS**
& their applications