IT METRICS FROM CUTTER INFORMATION CORP.® STRATEGIES

HELPING MANAGEMENT MEASURE SOFTWARE & PROCESSES AND THEIR BUSINESS VALUE

IT Organization, Benchmark Thyself

by Michael Mah

Last month in the article entitled "Meditations on Which Metrics Matter," I described the importance of two perspectives of IT measurement. One was the measurement of outcome, which addresses the gains that are accomplished from the implementation of a technology or an IT application. In many companies, the responsibility for this aspect of measurement might fall within the individual business units — the end user — and not with IT itself — the provider of technology.

The other perspective I discussed was measurement of output. This aspect deals with benchmarking the productivity of an IT department. The goal is to quantify the capacity of IT to deliver applications for use by the individual business units or end users of IT. The responsibility for this largely falls within IT.

In some ways, these two perspectives are quite separate issues. For instance, an IT department can undertake heroic efforts on a very complex project and succeed in delivering a system. Take, for example, a system that might result in large amounts of revenue to the company — one that enables them to enter a part of the marketplace that was not otherwise possible and generates huge returns. What if the "productivity" exhibited by the project during its design, construction, and implementation was not stellar — and perhaps for good reason? There might have been a great deal of cutting-edge technology that required immense development research. Things took time. There were unforeseen labor costs. It was *hard.*

To take a nonholistic view of this project would do everyone involved a great injustice. If low values for productivity metrics were used to judge

Continued on page 2.

Discipline of Market Leaders and Other Impediments to Measurement

by Stan Rifkin, Master Systems, Inc.

We often hear that it is difficult to get software measurement into practice. At least one important reason for this is that traditional software measurement is not aligned with the strategic objectives of the organization. When software measurement is aligned with an organization's market discipline, the implementation is accelerated.

As stated above, one of the reasons it is difficult to get measurement implemented is that it is unaligned with organizational objectives. For example, measurement is traditionally used to increase quality, increase programmer productivity, and reduce costs. Oddly enough, these are not the highest-priority objectives for a number of organizations; therefore, traditional measurement is difficult to implement in those organizations.

Continued on page 6.

executive summary

The first article in this month's ***ITMS*** follows up on "Meditating on Which Metrics Matter," an article that appeared in the February issue of ***ITMS***. Last month's article focused on the importance of a usable and practical approach to IT metrics, one that is intended to help us understand IT productivity and make better decisions in the process.

This month, I describe the fundamental concept behind the "how-to's" for building your own productivity baseline — without needing a consultant. It starts with the Carnegie Mellon Software Engineering Institute's "minimum data set" discussed last month, but it can be extended to further levels of granularity. The essential goal is to move away from a "numbers-numbing" approach to metrics that is characteristic of arcane numeric tables and ratios. Instead, I set the stage for a graphical approach where a picture says a thousand words. To illustrate the point, I use examples drawn from health statistics maintained by the Centers for Disease Control and Prevention and the National Center for Health Statistics.

The next two articles are from guest authors whom I greatly admire. The first is by Stan Rifkin who, prior to his move to Master Systems, Inc., was a key figure at the Software Engineering Institute. His thought-provoking article discusses why it is often difficult to implement measurement because of misalignments between metrics and organizational strategy. What he says might make you see things in a different light, and he offers some new concepts you may want to enact in your organization.

Then Jim Mayes of BellSouth tackles the often overlooked problem of one-dimensional thinking in metrics. In this article, he addresses a holistic and practical approach to achieving speed, cost, reliability, and business benefit for IT projects.

Overall, the direction we're heading toward with ***ITMS*** is "try this at home." We think you'll be pleased with the results.

Michael C. Mah, Editor

march 2000 vol. VI, no. 3

CUTTER CONSORTIUM

Continued from page 1.

the team unfairly, it would deny the fact that great things were achieved to overcome the technical challenges many laypersons might not appreciate. To attack the project leader for the project's high costs per module, line of code, or function point would be a travesty.

At the other end of the spectrum, an IT department might achieve very high levels of application development productivity for a system that provides a small-to-modest business benefit to the corporation. If this IT department were in the 90th percentile for speed, cost performance, and reliability, but projects were deployed that did little for the company's competitive position in the marketplace, that would be an ineffective and nonstrategic use of IT.

Therefore, it would behoove progressive-minded executives to capture both aspects of IT measurement. It's vital to know both the productivity (capacity) of your IT organization and the tactical leverage (benefit) that is achieved by the thoughtful and strategic use of technology. Obviously, it would be desirable to have the best of both: great outcomes with high business benefit from IT that are produced at a high output speed — lots of a good thing.

But how might output or outcome be visually portrayed to communicate comparison against a frame of reference? Let's look at the medical field to illustrate how that is achieved in the area of health statistics.

"Compared to What?": A Frame of Reference (Pink for Girls, Blue for Boys)

I recently had the unfortunate need to bring my three-year-old son to the emergency room at 2 am. He woke up hysterical with a sharp pain, and my wife and I could not calm him down. Luckily, everything turned out fine. Doctors and nurses in the ER helped us through the mini-crisis.

We were wrapping up paperwork before heading back home, and while I was waiting to leave, a series of charts and graphs hanging up on the hospital wall caught my eye. They are graphs familiar to every parent, but that night they sparked an awareness. I've used these charts and graphs to illustrate a point while speaking at software conferences ever since.

The items in question are growth charts. They're used by every pediatrician to "benchmark" height and weight for children as a function of age, from birth through age 18 — pink for girls, blue for boys. I looked at the bottom of one of the charts and saw a footer that read, "Health Resources Administration, National Center for Health Statistics, Centers for Disease Control." This organization maintains a database of health statistics that makes it possible for all of us to better understand health issues, giving us a framework for comparison and for understanding causes and effects.

For years, I've been sharing duties with my wife when it comes to doctor's appointments for the children. Seeing the charts on the wall of the ER reminded me that my children's pediatrics office keeps these benchmarks of both our children, David and Tara, which are updated at every appointment. Their height and weight are plotted on charts provided by the National Center for Health Statistics (NCHS) and serve as a frame of reference against which I can compare. There are several metrics of interest, including height (measured as length before age three), weight, and head circumference. David's chart is shown in Figure 1. His older sister Tara's birth-to-age-three chart (she's now seven) is shown in Figure 2. You'll notice that the trends are nonlinear, as is the case with almost every piece of metrics data I've seen.

Information on growth charts and more is available to anyone through NCHS. The Center was formed in 1960 with the merging

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.

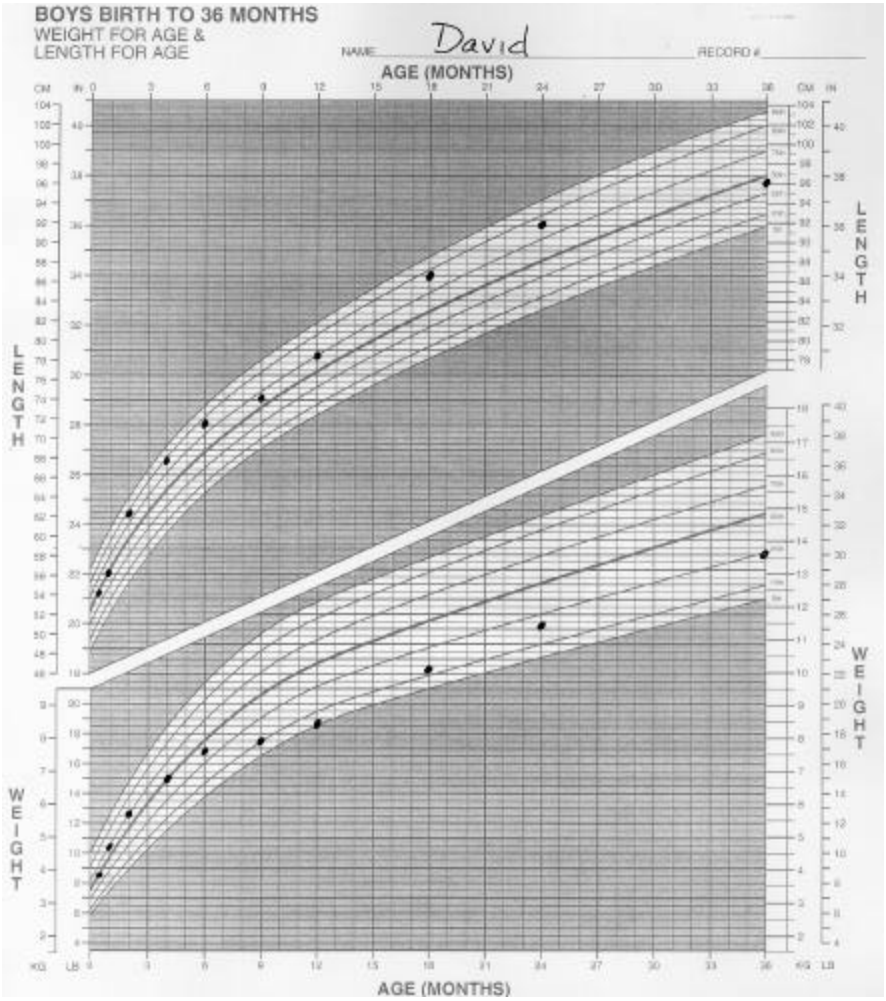

Figure 1 — David's growth chart: length and weight versus age (birth to age three).

of the National Office of Vital Statistics and the National Health Survey. It is part of the Centers for Disease Control and Prevention (CDC) under the Public Health Service Act. The Act authorizes data collection, analysis, and dissemination on a broad range of health-related areas. I went to CDC's Web site (www.cdc.gov) and nosed around. There I found a very valuable overview document describing the programs and activities for NCHS. From this document, I gleaned a few useful items that are related to data collection:

> Information plays a crucial role in public health and health policy. NCHS obtains statistics through a broad-based program of ongoing and special studies…. These fundamental public health and health policy statistics meet the needs of a wide range of users.

In the section on the National Vital Statistics System, it described the types of data collected, the data collection method, and its presentation of the data:

> NCHS has two major types of data systems: systems based on populations, containing data collected through personal interviews or examinations; and systems based on records, containing data collected from vital and medical records.
>
> NCHS cooperates with the States to develop and recommend standard forms for data collection and model procedures to ensure uniform registration of the events.
>
> The National Vital Statistics System provides technical assistance to the States through handbooks, instruction manuals, software, and special training courses.

What's fascinating about this document is that it not only outlines the various surveys

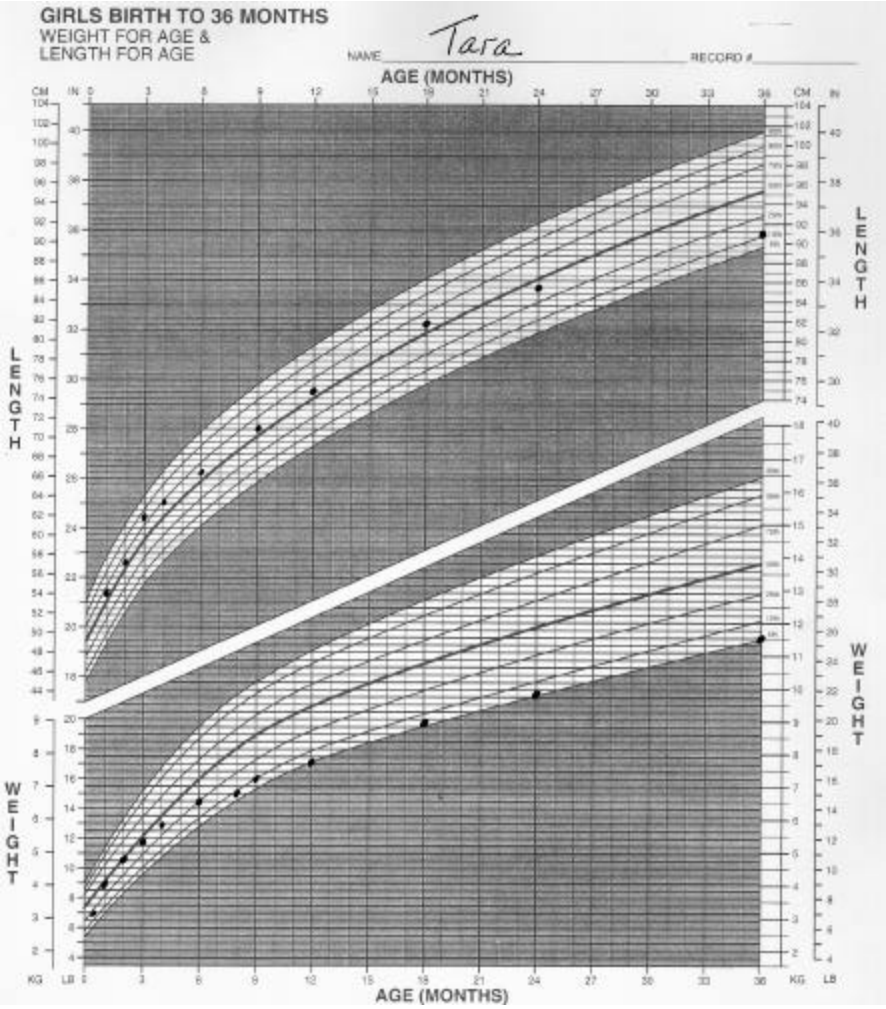

Figure 2 — Tara's growth chart: length and weight versus age (birth to age three).

and data systems, but it also lists the data sources, sample characteristics, planned periodicity (how often benchmark data is updated), and future plans. Measures are not static, they are dynamic and evolving. These are living benchmarks.

The same should apply to your own metrics databases. Keeping them up to date makes them valuable for strategic and tactical decisionmaking. This is especially vital in a rapidly changing marketplace.

Interpreting the Metrics

Even though the doctor's office collects the information, pediatric growth charts are simple to understand and interpret. Reading the metrics is easy; you don't need to be a scientist or "metrics analyst" to understand what the picture tells you. To use the old cliché, a picture is worth a thousand words (or a thousand numbers). This is a visual explanation that answers the question "Compared to what?" The same should apply to IT metrics: you shouldn't have to be a rocket scientist to analyze the data.

Once you can read the information, what do you do with it? Let's return to David's chart (see Figure 1 on page 3). As you can see, up to age 3, David was a long and slender fellow. His length (height) consistently was in the upper quartile. All but the last of the data points were above the center, or average line, falling between the 75th and 90th percentile. On the other hand, his weight was in the lower quartile. Particularly from nine months on, he consistently was in the 10th to 20th percentile. Skinny guy.

From when he was 18 months until 24 months, I could not imagine that the height came from his Asian heritage. Not many people on my side of the family were very tall. But if you look at my wife's family,

which is Greek and Irish, David's grandfather is 6'2" and played competitive baseball in college. Ah, genetics. Then at his 3-year appointment, his height plotted on the average. That seemed to make sense. We'll see what happens at his 4-year checkup.

Tara is quite different, as you can see in the chart in Figure 2 on page 4. Her height is more on the average line, while her weight is in the lower quartile from birth to age three. Now seven, on her most recent charts, the metrics have both drifted into the lower quartiles. She's a petite little thing. Although these measures look low, these attributes come in handy with her gymnastics: she can do a mean back handspring and is great on the uneven bars.

Collecting the Data: You Can Be Your Own NCHS

NCHS has a 40-year lead time on the mechanisms for uniform data collection, analysis, and dissemination of health-related metrics for the medical industry. The framework is reliable and proven. It even has a research program on automated statistical and graphical technology, including automated mapping of statistical graphics and statistical atlases. And they turn numbers into pictures (á la Yale University professor Edward Tufte), as in the growth charts examples.

Moreover, the information is made available electronically on the NCHS Web site. Information is instantaneous and broad, providing easy access to a wide range of data. There are links to other social and federal agencies, a data warehouse with detailed statistical tables, and a query capability that allows users to direct statistical questions to NCHS technical information specialists.

You can assimilate this kind of framework into your IT organization and thereby fundamentally change how it uses metrics.

Starting with the Software Engineering Institute's (SEI) "minimum data set" (see ***ITMS***, February 2000), an organization can establish consistent standards to gather profiles about its IT projects and build the equivalent of its own growth charts.

Start Charting

On the X-axis, you would start with an independent variable. On growth charts, this independent variable is obviously age; the dependent variables are height and weight. With successively higher values for age, there are higher values for height and weight. You can see these relationships visually.

For IT projects, you have several options from which to choose. I recommend starting with these variables:

- Use project size (i.e., code, function points, objects, modules) as the independent variable on the horizontal or X-axis, with size increasing from left to right.
- Create a vertical axis with measures for different charts, such as schedule, effort, and defects, respectively. Using these measures, you will be able to see how fast your projects are completed from small to large projects, how much effort they expend, and how buggy they are.
- From there, experiment with any combination of independent and dependent variables. You might want to see defects versus team size. Later on, as you add metrics to the minimum data set, you can plot other trends. One might be business benefit or expected revenue versus project size or business benefit versus schedule.

With examples like these you might find a pattern that shows whether small projects tend to bring more benefit than large ones, shorter projects bring more benefit than longer ones, and so on. What you find might surprise you. Whatever measures you choose, you're on the road to creating your own baseline. You'll be amazed by what you can do with data like this when it comes to things like outsourcing, service levels, process improvement, project estimation, and organizational learning.

Managing the Data Without Hiring a Consultant

Here's how to start and maintain your data collection process, without having to hire a consultant.

Establish a Routine

- At the time a project is deployed, **hold a meeting to record the "vital statistics" of a project.** After completion, before

people scatter to other projects, gather the core metrics.

- While you're at it, **write down what worked** and what didn't — all the environmental characteristics of the project and the lessons learned, warts and all.
- **Tell people the purpose of the endeavor; make them co-owners of the process.** Explain that even if the project struggled, the post-implementation review isn't about blame and attribution; it's about acquiring learning from the experience to help with the next project.

Organize What You've Learned

- **Keep the knowledge of what happened in a library of information** that people can access later.
- **Keep this record electronically.** Add the project to a metrics database (preferably Web-centric) that will help build a growing, living baseline to better understand the organization's IT capacity. Use an application service provider model as the basis of the architecture. This ensures that all the collective wisdom is maintained in one place for the organization to tap into. Knowing IT capacity will ensure that new projects will be more realistic, staying within the organization's capability (aka technology bandwidth). Hopefully, this will also provide a sanity check for future promises, so that teams will not be saddled with targets that are far beyond what is reasonable.

Present What You've Collected

- **Plot multiple charts on one view.** Sometimes the relationships between metrics are more obvious when you see them side by side. For example, a chart with speed versus project size juxtaposed with a defect chart will show you the impact of accelerating schedules on reliability. To give an example: ongoing metrics research on industry data shows that it is not unusual for defects to rise sixfold when you double the project staff in an attempt to achieve an "Internet speed" schedule. I've described this as the 200/20/6x rule: double the staff by 200%, shorten schedules by 20%, increase defects by a multiple of 6. Scary phenomenon, but important to know during planning and execution.
- **Sort data by selection sets.** This will reveal the various patterns that inevitably emerge for different types of projects. You'll see how new development behaves compared to major enhancements, minor enhancements, broke/fix maintenance, and other classifications. You'll also reveal patterns that might be exhibited between different lines of business. Network applications, billing systems, customer care, and enterprise/financial projects might all exhibit different levels of productivity, and for good reason.

Where You Can Go from Here

These steps are a good place to start for do-it-yourself metrics. In the coming months, we'll feature case studies and articles describing how companies have implemented metrics frameworks like those mentioned. There is also plenty to say about productivity baselines for outsourcing, process improvement, organizational learning, conflict management, and negotiation.

Charts like the ones discussed here often yield pleasant surprises, too. They're in the form of the "long-necked giraffes" I referred to in the February issue of ***ITMS***: projects that reveal just how special they are when measures tell a previously unknown story. They have a strange habit of popping up in ways IT professionals may not have expected.

Upcoming issues of ***ITMS*** will offer an overview of productivity statistics and trends from industry research as well as a challenge to the IT industry for a new metrics initiative.

Continued from page 1.

This article seeks to bring together two normally disparate subjects: organizational strategy and software measurement. It looks for what is commonly a misfit between strategy and measurement and proposes a set of antidotes. The misfit might be, for example, between a company's organizational strategy and the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) for Software or the International Organization for Standardization's ISO 9000 standard for software quality systems.

A third area where you may find that misfit for metrics initiatives is in difficulty of implementation. In other words, because of the misfit between organizational strategy and software measurement as traditionally practiced, implementing software measurement is impeded.

The Discipline of Market Leadership as a Guidepost

The Discipline of Market Leadership is a survey of how 80 organizations out-achieved their competitors. The authors found that the answer to organizational success was focused on one of three market areas or disciplines:

- Operational excellence
- Customer intimacy
- Product innovativeness

Operationally excellent organizations have a "formula" for their services or products. Their menu of choices is limited, but within that menu, they deliver excellently. Common examples are McDonald's and Federal Express.

Customer-intimate organizations seek quite a different market niche: a total solution. Whatever the customer wants gets added to the menu. The menu is long and custom-made for each engagement. Financial service institutions are a great example. For them, customer intimacy is a way to get a greater share of the customer's wallet, since there are very few other venues for all the services that financial institutions now offer (checking and savings accounts, overdraft protection, certificates of deposit, credit and debit cards, traveler's checks and money orders, foreign currency exchange, travel arrangements, insurance). All of the "big five" accounting firms are customer intimate.

Product-innovative organizations pride themselves on maximizing the number of turns they get in the market. They introduce many new products, selling innovation and features as opposed to price. Examples are Intel, 3M, Sony, and Bell Labs. They measure their success by the number of new product introductions, the number of patents, and/or the number of Nobel Prizes.

The authors of *The Discipline of Market Leaders* are quick to point out that all organizations must have at least threshold characteristics of all three disciplines, but they must focus on and excel at only one.

One example of lopsidedness cited is IBM. At one point, its legendary customer intimacy was outweighed by its inattention to price (or operational excellence). The result was that competitors that were not as strong in customer intimacy could still make inroads to IBM customers with a lower price.

Measurement for Operationally Excellent Organizations

Measurement of the type we are used to, the type espoused by SEI and the International Function Point Users Group, for example, applies almost exclusively to organizations wishing to be operationally excellent. Our current measurement or improvement methods typically have nothing to offer customer-intimate and product-innovative firms.

The problem is that many software development organizations themselves do not strive to become operationally excellent, so we have neglected these areas. We hear keynote speakers at national conferences and their criticism of the process. These speakers talk about how they have to wrestle resisters to the ground, how managers handle time bombs and remain clueless, that would-be adopters are too impatient. We hear them say that disaster is imminent, businesses will be ruined, software professionals are irresponsible and guilty of gross malpractice, and, in the end, that everyone involved simply has bad character!

In fact, this criticism stems from nothing more than a mismatch of goals. There is, for example, a large set of software development organizations that strive for customer intimacy and essentially will do anything their clients request. Those organizations get to know their clients very, very well — sometimes better than the clients know themselves. An example of this might be a payroll service that has seen every variation on payroll and knows more about payroll processing than any inhouse payroll department could. The most customer-intimate payroll service providers could easily take over their customers' entire payroll departments!

To take another example, what about Microsoft? What do you think its market discipline is? Its discipline is product innovation. It touts its new, glitzy features, not its up-time or reliability. It wants to own/earn its clients based on new features, not by offering software that is operationally excellent.

In this context, CMM is silent on product innovativeness and customer intimacy. It applies only to organizations wanting to be operationally excellent. The same is true for traditional measurement.

Missing: Measurement for Customer-Intimate and Product-Innovative Organizations

What are we missing in all of this? 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 is not an indicator of bad character or resistance, but may be an appropriate response to measures that do not fit the strategy. We need to problem solve together with our clients to develop new classes of measures that simultaneously meet our high standards for objectiveness and their high standards for relevance. Let me relate several efforts in which I have participated.

Example 1: The brokerage house. One brokerage house was not interested in software costs or quality, but rather what it called time-to-market. During the frantic time that a deal (such as an initial public offering) was being put together, the IT department was asked to respond quickly. The response had to be quick enough that the broker could earn as much as possible by offering as many services as possible during the short services-negotiation phase of the initial public offering lifecycle. It was a question of wallet share, which is a customer-intimate measure. So the brokerage really wanted the customer to maximize spending with the brokerage. That meant it needed the longest menu of services possible. It appeared to be a time issue, but if we would have tried to improve delivery time, we would have missed the point — time was not the major variable at all. It was flexibility: already having a systems architecture that could accommodate the requested services without having to engage in time-consuming new development. What looked like a time question was in fact a flexibility concern: Was the systems architecture sufficiently flexible to incorporate the new features/services with little additional programming?

We settled on a measure of the percentage of the total deal that did not go to the brokerage. IT's job, then, was to offer a realistic plan for continual reduction of that "missed wallet share" figure. Incidentally, this brokerage has a CMM-based software process improvement program that was frustrated, underfunded, and generally neglected. The new measure invigorated and revived the improvement program by taking the focus off irrelevant, operational-excellence goals and shining on what really counted for the business: winning as much of the initial public offering deal as possible.

Example 2: The defense contractor. One computer-oriented defense contractor said it wanted project measures: a one-time, one-budget record. But when pressed, it became clear that projects were not managed — and therefore not measured — in the traditional way. The government client wanted a provider that would do what it requested, not one that would study the request and offer alternatives or push-back. Cost, quality, and duration were not important to the client, only that it got what it wanted in reasonable terms. This, too, is a customer-intimate approach, one that makes the menu of services as long as the list of customer requests.

Naturally, the provider has to deliver the systems within a threshold value of cost, quality, and duration. But already there were many other providers that performed better in terms of cost, quality, and duration and were rated too low in customer responsiveness to be considered! In fact, the client changed its mind often, rendering previous work inapplicable. This caused rework that would traditionally be held against the provider. Traditional project-oriented measurement was irrelevant in this setting.

We recommended several measures: the total spent by the customer; how much went to other providers (to be minimized); time spent in adversarial settings (to be minimized); time spent with the customer understanding its business (to be maximized); and number of people on the staff with credentials like our client's (to be maximized). All are

customer-intimate measures, not project performance ones, since the contractor is not really held to traditional project performance standards.

Example 3: The computer services firm. A computer services firm had been the prime contractor for a long-time government client. The firm provided all of the computer programming and operations for a particular type of payment that the government entity made to deserving applicants. The contract was up for renewal (that is, to be recompeted) and the incumbent wanted to propose a set of measures going forward that would indicate its operational excellence.

The usual suspects were offered in discussions with the provider (now bidder), but those measures did not seem to resonate, even though they were "reasonable." It turns out that the government organization was feeling 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 lower costs and greater flexibility.

The measures we settled on were:

- Plan versus actual implementation of a set of new technology introductions
- Hours spent training the government client on the principles of that new technology
- Reliability measures directly related to the government organization's business, for example: 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 applicants due to provider service failures

These measures were *instead* of other, traditional measures such as percentage of system availability data-entry error rates and a threshold number of ABENDs (abnormal end of task) per day, none of which related to the government mission or daily reality. This, too, is a customer-intimate strategy: expanding the menu of services to include new technology and then measuring around the effectiveness of the menu.

Customer-Intimate Traits

Customer-intimate organizations seek flexibility so that they can extend their menus (infinitely). Accordingly, in order to be aligned with that organizational strategy, they need measures of flexibility and wallet share. For example, in peer reviews, the items to be examined most closely should be the elements that limit future options, such as a limit to the number of items in a list and built-in "magic" numbers. It is also critical to judge comprehension during reviews; artifacts will constantly be expanded and enlarged as a strategy, so they have to be understandable.

Configuration management for customer-intimate organizations should be measured by how many of the interfaces are managed. It is the interfaces, after all, that matter in an ever-expanding system. This will enable multiple end-to-end solutions.

Probably the most important ingredient for customer-intimate systems providers is the existence of a systems architecture. The details of systems architecture, particularly software or applications architecture, are beyond the scope of this article; suffice it to say that architecture deals with the highest level of abstraction, the one expressing the relationship among the largest entities and their patterns of connection and interaction. Therefore, one simple measure related to customer-intimacy strategy would be the count of architecture checks and violations.

Product-Innovative Traits

The mark of a product-innovative organization is a concentration on features at the expense of quality, reliability, cost, and flexibility (unless those are the features being optimized, which is rare). Users of such products have a certain patience that is required with all new kinds of products, such as the PalmPilot, Walkman, Watchman, wearable cell phone, Linux, and Windows 2000.

One of the disciplines to fall by the wayside of innovative organizations is traditional planning. Planning is not as important as innovation. One often hears, "The plan is not a deliverable!" Planning for these organizations

is more about a diversity of investment alternatives — planning that some "bets" will fail to bear fruit and creating a diversified portfolio. We see this plainly with pharmaceutical firms, which do not require that a particular drug be discovered by a particular deadline, but rather that discoveries are regularly in the pipeline and, on balance, that there is a healthy proportion of winners. (Anyway, what would their plan be — "Budget for 1.4 stunning breakthroughs per fortnight?")

For those who care about a process focus, the challenge here is to create lightweight, generic processes that can be applied with large helpings of intelligence and judgment. I suppose the measure of "lightweightness" is really "fit": how well do the processes fit our strategy?

Those of us with a process focus hate the phrase "good-enough quality." But that is what is required in innovative firms. Again, quality is not the deliverable — features are. Therefore, goals around quality are pegged to thresholds, benchmarks, and, especially, time to market. Again, our measure would look at comparative fit: how does our quality stack up against competitors with similar features and time-to-market requirements?

Organizations with customer-intimate or product-innovative strategies are organized differently than those with operational-excellence strategies. Paul Lawrence and Jay Lorsch find that product-innovative companies, in particular, have both high differentiation and high integration. Differentiation refers to experts; integration refers to the job of getting disparate, possibly competing experts to serve in the interests of a common, corporate goal. One of the measures I use (yes, I measure more than product and process) is a count or proportion of the number of people in the organization whose job it is to integrate those competing interests to make a product happen. In the applications area of Microsoft (Office and programming language products) for example, there is such a person who heads a 10-person team, so that both the count and ratio are high at Microsoft relative to customer-intimate and operationally excellent firms.

We Need More

The implication for measurement is that a wholly different set of measures would apply to the customer-intimate and product-innovative activities compared to the technical activities; the technical problem is more or less solved with the measures we have. Now, we as a profession need to turn to the other two disciplines of market leaders and offer them something!

Acknowledgments

I learned most of this by working with John Tittle of Computer Sciences Corporation. The measurement leader who made me ask myself many of these questions is David Card. Finally, I acknowledge the many SEI Software Engineering Process Group conference keynote speakers/cheerleaders who have claimed that those who resist have bad character. This has irritated me into writing on this topic; the speakers' failure to ask (and answer) "Why?" stimulated my thinking in the first place.

About the Author

Stan Rifkin is a principal with Master Systems, Inc., an advisory services firm that concentrates on improving the processes by which organizations manage and develop software. Each year *Wall St. and Technology Magazine* selects two CIOs of the year; last year Rifkin was the only one to have served as consultant to both of these CIOs. He has worked at the Software Engineering Institute, where he was a project leader in the process improvement program. Rifkin's speciality is implementation, deployment, and putting best practices into practice. Rifkin can be reached at sr@Master-Systems.com.

References

Lawrence, Paul, and Jay Lorsch. *Organization and Environment: Managing Differentiation and Integration*. Harvard University Press, 1967.

Treacy, Michael, and Fred Wiersema. *The Discipline of Market Leaders: Choose Your Customers, Narrow Your Focus, Dominate Your Market*. Addison-Wesley, 1995.

Wiersema, Fred. *Customer Intimacy: Pick Your Partners, Shape Your Culture, Win Together*. Knowledge Exchange, 1996.

Achieving Business Objectives: Balancing Time, Cost, and Quality

by Jim Mayes, Estimation Consultant

bal.ance (bal´ans), *n.*, *v.*, … 7. act of balancing; comparison as to weight, amount, importance, etc.; estimate… — *v.t.* 14. to weigh in a balance. 15. to estimate the relative weight or importance of; compare; *balance probabilities.*

— *The American College Dictionary*

The objective of any software project is to optimize time, cost, and quality relative to the expected business value to be received from the software product. To ensure that this happens, the client organization (or business unit) should be responsible for determining the time-cost-quality (TCQ) drivers related to the business, and the IT organization should determine the TCQ drivers related to the software. A partnership between the IT and client organizations is required to achieve balance between software estimates and business objectives.

The "Holy Grail" of Estimation

The Holy Grail of the software estimator is a perfect estimate: one that satisfies the client's business needs, meets the business needs of the IT organization, and makes *everybody* happy. However, as the saying goes, "you can please some of the people some of the time, but you can't please all of the people all of the time." Software project objectives can be identified and successfully achieved only if there is a partnering relationship between the IT and client organizations. Both parties must build trust by sharing information for making business decisions, and both must focus on the business objectives, business values, and cost drivers. The estimate negotiation process should definitely be more constructive than the one involved in buying a new car. Therefore, the IT organization should not have a one-size-fits-all attitude with regard to estimates provided to clients.

Every estimate opportunity is a search for this "grail," along with an accurate prediction of the outcome of the software project itself. Is perfection achievable? Probably not, since it is obvious that no one knows exactly what the actual results of a software project will be until after it is completed. But this is exactly the basis for achieving balance. A *fact* is what has really happened, something known to have happened, or a truth known by actual experience. Experience is key, because unless actual project data is captured, there is no basis for balancing time, cost, and quality. To *estimate* means to form an approximate judgment, calculation, or opinion, and to submit approximate figures for the work to be done. The more the estimate is based on fact or actual project results, the more accurate it will be in satisfying the business needs related to time, cost, and quality.

What Is a Perfect Balance?

Perfection is defined as the state of highest degree of proficiency. The client usually has a different view of "perfect balance" than the IT organization. The IT organization generally provides estimates based on the optimization of time, cost, and quality, but this general optimization may not meet the business objectives. So even if the estimate is perfect in the eyes of the IT organization, the customer may not accept it; perhaps the customer's deadline drives the business objectives or the customer has a limited budget for making the project viable. Alternatives (provided by the IT organization) should be discussed with the client.

Business Value

The wild card in the TCQ balance triangle is business value. A state of perfect balance can be achieved when the client and IT organizations work together to balance time, cost, and quality, relative to business value. The business value associated with time, cost, or quality must also be traceable to the business objectives that the software product is expected to achieve. Think of the TCQ scale as having three platforms that are all linked to a center point for support, as shown in Figure 3. This central point equates to customer satisfaction and business objectives. For example, as more business value is placed on the schedule platform (shortened

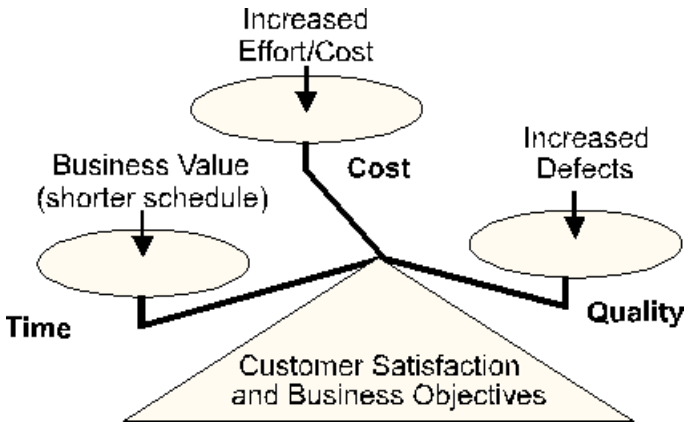

Figure 3 — Balancing time, cost, and quality versus business objectives.

schedule), the scale goes out of balance. If the software development process is stable (process, technology, and people are optimized), the defects on the quality platform and the cost (effort) on the cost platform must increase to balance the TCQ scale.

The perfect balance of time, cost, and quality versus business value is a business decision that should be made by the client. This business decision is related to specific business objectives, and the software product may be only a small piece of the puzzle. A responsible, mature IT organization provides the information necessary for the client to make informed and justifiable decisions related to business objectives. IT can partner with the client organization, with regard to business development and work management, by focusing on the business objectives.

TCQ Process for Achieving Balance

Although perfection may not be attainable, it is possible to minimize the uncertainty and risks associated with software project estimates. This is the goal of the TCQ process, along with aligning the estimate with business objectives. This process includes the following steps, each explained further in subsequent paragraphs:

- Identification of business objectives and constraints
- Project data collection and analysis
- Software estimate validation
- Phased estimation
- Partnering negotiation

Identification of Business Objectives and Constraints

Identification of the business objectives, cost drivers, and project constraints is an important first step in the estimation process for achieving balance. The project manager and estimator must understand what is driving the customer request relative to the business objectives, as illustrated in Figures 4 and 5.

Figure 4 shows how priorities can be assigned based on the probabilities allowed for exceeding a particular goal, all related to business value. For example, if controlling cost has the highest business value, followed by schedule and quality, then the desired probabilities may be identified as 90% cost, 65% schedule, and 40% quality. This is the starting point for partnering negotiation, which lasts throughout this process as different iterations of the estimate are reviewed.

Project/Process Data Collection and Analysis

The key to linking business objectives, assigning business value, and associating the impact on time, cost, and quality is capturing actual project data. Key data elements are:

- **Size** — generally captured in the form of function points or source lines of code
- **Effort/cost** — actual staffing rate and effort, including overtime and other costs
- **Schedule** — phase start and end dates; milestone dates
- **Defects** — defect rate and severity level
- **Project characteristics** — technology, process, and people

Figure 5 illustrates how the data is linked to time, cost, quality, and business value. The data should be stored in a data repository for data analysis and for use in top-down estimation. Historical project and process data has a variety of uses related to balancing time, cost, and quality, such as:

- Analyzing project characteristics over time
- Providing detailed information on cost drivers
- Calibrating such things as production and defect rates

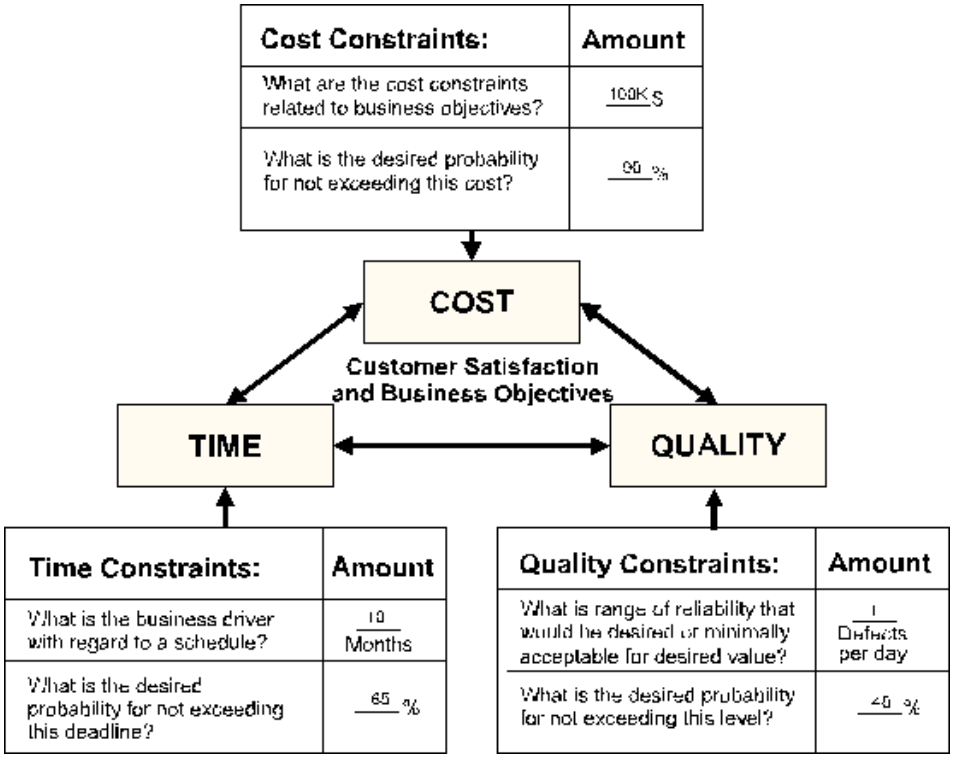

Figure 4 — Identification of business drivers: customer view.

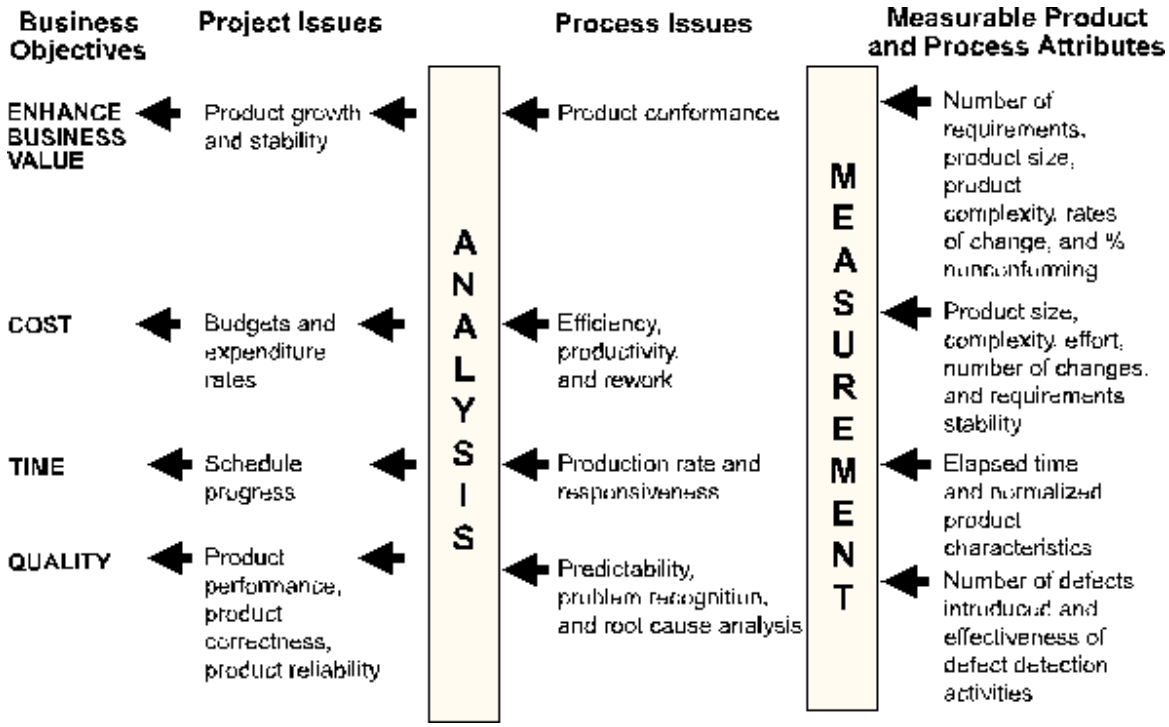

Figure 5 — Measures for achieving balance and improvement.

- Analyzing scope creep in project size during development
- Illustrating the relationships associated with estimated size, schedule, staffing, and defect rates
- Providing information to the client for justifying estimates and for making business decisions related to time, cost, and quality

Software Estimate Validation

Software estimate validation, illustrated in Figure 6, involves taking bottom-up task-level estimates created using standard project management tool methodologies and comparing them to top-down phase and milestone estimates created using quantitative historical project data. The top-down estimate provides the quantitative data for determining when balance has been achieved. This comparison is based on the project constraints driven by the business values and objectives identified by the client. This quantitative analysis facilitates a feedback loop to the client for justifying the validity of the estimate and for making business decisions. The bottom-up estimate is not expected to match perfectly with the top-down estimate; however, based on statistical variance analysis, it should be within upper control limits (UCL) and lower control limits (LCL) appropriate to the lifecycle phase being estimated.

As long as the estimate is varying above and below the mean within the control limits, which equates to common cause variance, then the estimate(s) would be considered as "validated." If the estimate is outside of the control limits, or consistently above or below the mean, this would indicate a special cause variance. This is normally when root cause analysis is required, based on the balance to time, cost, and quality. If the desired balance cannot be achieved, the results are then shared with the client, along with additional information for making business decisions that may be appropriate for the project, such as resizing (changing the scope); reassessing the business value and needs relative to time, cost, and quality; or canceling the project altogether.

Phased Estimation

Phased estimation takes validation one step further. At the beginning of the project (preplanning) and after each of the subsequent phases (planning, analysis, and design), a new estimate is prepared for the next phase and the remainder of the project. Validation is repeated for each phase estimate. At each point, a "gateway" decision

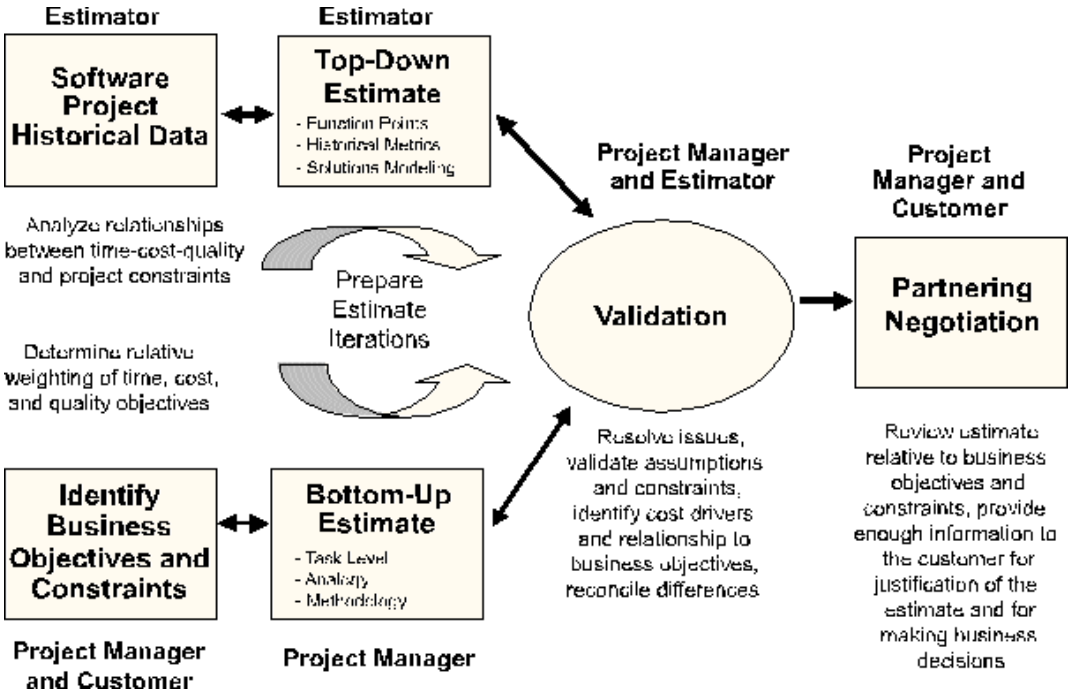

Figure 6 — Software estimation and validation.

Table 1 — Example of Estimate Validation and Phased Estimation Tolerance Guidelines

The % tolerances are based on expected change from the end of each phase forward.

The top-down and bottom-up estimates being compared should reflect the desired probabilities with regard to time, cost, and quality identified by the customer. This should be used to validate whether those probabilities are feasible and to illustrate other optimized solutions.

±**Time or cost/effort or quality** — Acceptable time, cost/effort, and quality tolerances.

Tolerance assessment — Acceptable (time, cost, and quality factors within tolerance), Moderately Acceptable (one factor is out of tolerance), and Not Acceptable (two or more factors are out of tolerance). Factors not within tolerance should be analyzed, and changes to the top-down, bottom-up, or both estimates may be required.

NA = Not applicable

Estimate Phase	Estimate Procedure[1]	±Time	±Cost/ Effort	±Quality
Feasibility	The estimator prepares a top-down estimate, based on an estimated function point (FP) size determined by comparison to other projects.	NA	NA	NA
Approximation	The estimator further refines the top-down estimate, based on further definition of the project. If there is a bottom-up estimate, both the estimates should not apply more than 75% contingency for scope creep.	±25%	±63%	±12.5%
Proposal and Launch	The project manager provides a bottom-up estimate based on very high-level requirements. The estimator further refines the top-down estimate. The estimates should not apply more than a 50% scope creep.	±20%	±50%	±10%
Planning	The top-down estimate will be revised. The bottom-up estimate will be provided for the total project and the analysis phase. Not more than a 40% contingency should be applied for scope creep.	±13%	±38%	±7.5%
Analysis	An FP count is performed on the baseline requirements and high-level design. The top-down and bottom-up estimates should not use more than a 32% scope creep contingency.	±10%	±25%	±5%
Design	Another FP count is performed based on the detailed design document. The top-down and bottom-up estimates should not apply more than a 5% scope creep contingency.	±5%	±13%	±2.5%
Review and Monitor — Change Management	The project team regularly reviews project progress and identifies new work elements for which it must prepare new baseline estimates. If a substantial increase in scope is suspected, an FP count should be scheduled. This may require renegotiating the estimate.	±5%	±13%	±2.5%

[1]An estimate is provided at the beginning of the project with an agreed upon ±10% commitment estimate, for at least one phase and a ±25% goal level estimate for the remaining phases/milestones. This estimate would be updated after each phase (until the construction phase), at least one phase forward with an updated ±10% commitment, and for the remainder of the project with an updated ±25% goal.

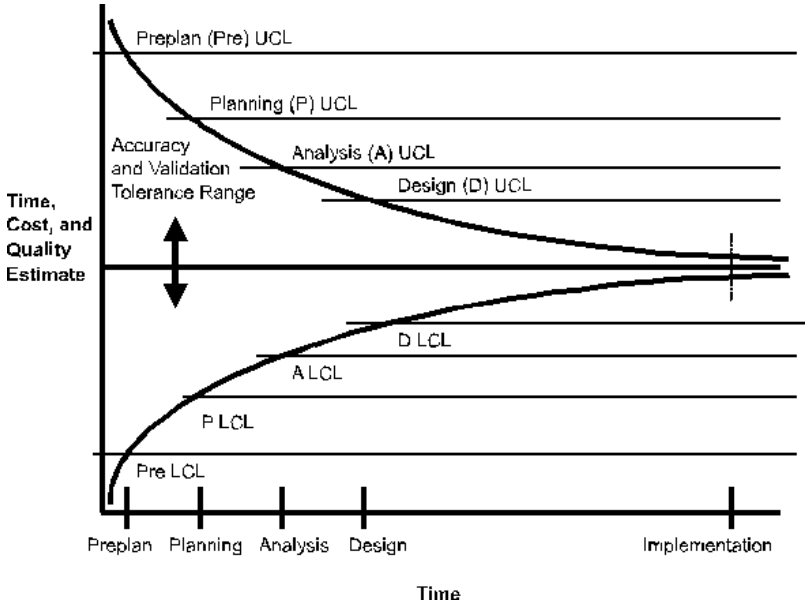

Figure 7 — Phased estimation and validation control chart.

should be made by the client, based on TCQ versus business value, as to whether the project should be continued or canceled. The UCL and LCL used for validation are adjusted appropriately for the phase being estimated, as shown in Table 1 (on previous page) and Figure 7, above.

Partnering Negotiation

The negotiation step involves a constructive dialog between the project manager and the client that starts when the estimate request is submitted and continues throughout the estimation process. The negotiation that occurs when the estimate is presented should focus on the business objectives and constraints identified at the beginning of the process. Neither party should be kept in the dark, guessing what the other party wants. For both parties to reach an agreement, the business and cost drivers should be reviewed. What usually happens is that the customer identifies a cost restraint and also states that schedule and quality are not negotiable. This should be addressed in the beginning; no matter what is declared nonnegotiable, the business objectives and constraints should still be identified and prioritized. When the estimate is presented, the project manager should provide enough information to show that the estimate provides the best optimization of the client's TCQ constraint probabilities. If the estimation process is approached from this viewpoint, and alternatives are provided by the IT organization, successful partnering negotiation can be achieved.

Conclusion

Balancing time, cost, and quality with business value is achievable only when there is a partnership between the client and the IT organization. Sharing information and trust are key elements to achieving customer satisfaction with the IT organization. Joint ownership of the business objectives as a mutual goal is the way to work through the issues objectively. The TCQ process for achieving balance includes identifying business needs and constraints, project data collection and analysis, software estimate validation, phased estimation, and partnering negotiation.

References

Cohen, Herb. *You Can Negotiate Anything.* Bantam Books, 1980.

Covey, Stephen R. *The 7 Habits of Highly Effective People: Powerful Lessons in Personal Change*. Simon & Schuster, 1989.

Fisher, Roger, and William Ury. *Getting to Yes: Negotiating Agreement Without Giving In.* Penguin Books, 1981.

Florac, William A., Robert E. Park, and Anita D. Carleton. *Practical Software Measurement: Measuring for Process Management and Improvement.* CMU/SEI-97-HB-003, 1997.

Putnam, Lawrence H., and Ware Myers. *Measures for Excellence: Reliable Software on Time, Within Budget.* Yourdon Press, Prentice Hall, 1992.

About the Author

Jim Mayes is currently an estimation consultant with BellSouth and is a certified function point specialist. He has more than 26 years of experience in software development and lifecycle management. Over the past 5 years he has been directly involved in providing quantitative software project estimation, data analysis, and metrics related to software process improvement and outsourcing initiatives at BellSouth. Mayes can be reached at www.softwareestimator.com. Tel: +1 770 587 4219; Email: Jim.Mayes@bridge.bellsouth.com.