

Forrester Consulting

MAKING LEADERS SUCCESSFUL EVERY DAY

July 2007

Changing The Cost/Benefit Equation For Application Development Metrics

*A Commissioned Study Conducted By Forrester
Consulting On Behalf Of Borland Software*

FORRESTER®



Headquarters

Forrester Research, Inc., 400 Technology Square, Cambridge, MA 02139 USA
Tel: +1 617/613-6000 • Fax: +1 617/613-5000 • www.forrester.com

Table Of Contents

Executive Summary	3
Metrics Are Of Strategic Value To App Dev Shops	3
Portfolio, In-Flight Project Metrics, And Post-Mortem Project Metrics	5
App Dev Shops Struggle To Balance The Costs And Benefits Of Metrics	6
Metrics Collection Is Complex And Costly	6
Application Development Shops Gather Superficial Metrics.....	7
The Path To Better, Cheaper Metrics	10
Transparent Collection Of Actionable Metrics Changes The Cost-Benefit Equation	11
Better Metrics Enable Corrective Action, Not Just Punitive Action	12
Conclusions	13
Appendix A: Supplemental Material.....	14
Background And Methodology	14
Related Forrester Research	14

© 2007, Forrester Research, Inc. All rights reserved. Forrester, Forrester Wave, RoleView, Technographics, and Total Economic Impact are trademarks of Forrester Research, Inc. All other trademarks are the property of their respective companies. Forrester clients may make one attributed copy or slide of each figure contained herein. Additional reproduction is strictly prohibited. For additional reproduction rights and usage information, go to www.forrester.com. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change.

Executive Summary

As part of a study commissioned by Borland Software, Forrester Consulting conducted 20 in-depth interviews with project managers, development managers, and executives in charge of application development and project management organizations to better understand the role of metrics in today's application development organizations. We asked respondents about the metrics they currently gather, how they collect them, and how they apply them. We also asked about the metrics they would like to gather. We found that application development organizations use metrics for highly strategic activities such as project selection, process improvement, and communicating IT's value to the business. These organizations are hungry for better metrics to inform these processes, but when they attempt to gather more and better metrics, they invariably select the wrong metrics to gather and employ inefficient methods of collection. This typically sours them on metrics and leaves them without the critical data they need to manage their projects and their overall organization.

Metrics Are Of Strategic Value To App Dev Shops

The ongoing collection of meaningful metrics is essential to the success of application development organizations as well as the success of individual application development projects. Without metrics, key functions like portfolio management, process improvement, and management of relationships with business customers are nearly impossible to perform successfully. The application development managers and executives we interviewed apply metrics primarily in the following three ways (see Figure 1):

- **Project portfolio management.** Without data about the capacity of the development organization and the amount of work currently underway, executives are unable to make informed decisions about workload capacity, turnaround time, and for monetary designation. When executives lack such data, they must estimate their organization's delivery capacity — living with undesirable results when they guess incorrectly. The app dev organizations that we interviewed informed us that metrics are an essential ingredient in their project approval and project portfolio management processes. At the level of the individual project, metrics are used in a similar fashion: to understand their expected accomplishments in a given time period — be it an activity like testing or a set of functionality to be delivered.

“When we adopted Agile nine months ago, we started relying more heavily on metrics so that we could plan for future iterations and know how much we could produce and how consistently.” (Public sector company)

- **Communication with the business.** Application development metrics matter outside of app dev, as well; nearly half of the firms we spoke with indicated using metrics to communicate with their business customers. The most common reason to share metrics with the business is to provide them with visibility into the work that IT is performing on its behalf and the benefit that the business is accruing. Keeping business customers in the loop when projects run into trouble is critical to the health of the business/IT relationship; although difficult, these conversations are better to have sooner rather than later. Sharing data is important in its own right, but it's also a prerequisite for remedial action. Accurate and meaningful metrics permit business and IT to make tradeoffs among scope, schedule, and budget; a third of those we interviewed cited these types of tradeoffs as a way in which they apply metrics.

“We do reporting twice a month as much to keep stakeholders engaged as for our own benefit.” (Multinational publishing company)

“We did a study and found that for every dollar we invested in IT, we were getting a 25 cent return. It was partially true and partially that we weren’t getting enough credit. The business accepted our software and thanked us, but it never measured the value it accrued. We committed ourselves to demonstrating the value IT was delivering to the business.”
(Pharmaceutical company)

- Process improvement.** Shops that measure the results of their projects and their overall portfolio have the benefit of knowing how successful they are. But shops that measure both their results and the processes they use to achieve them have the ability to correlate the two and identify process improvements that can generate even more success. The processes in question may be portfolio management processes, project management processes, development processes, or even specific practices within these larger processes. A third of the firms interviewed cited a desire to improve development and project management processes as a reason for beginning to collect better metrics. And even more — nearly half of interviewees — cited process improvement as a way in which they apply metrics. Many of the shops we interviewed said that they employ metrics to continuously improve their internal operations, even in the absence of any formal, discrete process improvement initiative.

“We do this because we believe in continuous improvement. We want a sustainable, repeatable, and predictable process. We also want to have a reputation like a dry cleaner: in by 9 o’clock, out by 3 o’clock.” (Property and casualty insurance company)

“The primary driver for collecting these metrics is to monitor and test out our development methodology.” (Life insurance company)

Figure 1: Application Development Organizations Use Metrics To Inform Strategic Activities



Base: 20 application development professionals at \$1B+ companies
(multiple responses accepted)

Source: 20 1-hour interviews with application development professionals commissioned by Borland Software and conducted by Forrester Consulting.

Portfolio, In-Flight Project Metrics, And Post-Mortem Project Metrics

Metrics can be applied in these and many other ways, but they can also be applied at different levels of scope and at different points in time (see Table 1). Each of these three levels is an important part of a comprehensive metrics program:

- **Portfolio metrics.** Executives and others with portfolio-level responsibilities, such as quality assurance, enterprise architecture, and the PMO, review metrics to get a sense of the health of the overall application development organization. But aggregated project metrics aren't the only kinds of portfolio metrics. In fact, some metrics are inherently about the portfolio and not about individual projects; examples include resource availability and utilization, the relative emphasis on maintenance versus new development, and conformance with process and architectural standards.
- **In-flight project metrics.** In-flight metrics are those gathered while the project is still underway. In-flight metrics are most useful to the development team itself, although others may also be interested. In-flight metrics can enable decisions that actually change the course of a project and affect post-mortem project metrics. Unfortunately, the metrics used for this purpose are usually projections of post-mortem project metrics rather than true in-flight project metrics. Lacking real in-flight project data, app dev teams rarely have objective data about the true status of their projects or about the root causes of their problems.
- **Post-mortem project metrics.** This is the application of metrics after project closure and is intended to judge the project's relative success. On-time, on-budget, and on-scope are the most commonly used measures, but they really say more about estimation accuracy than about whether a project took as long as it ought to have. Metrics like business value delivered and customer satisfaction are far more valuable but also far more difficult to collect.

While all three of these types of metrics are necessary, in practice, most of the metrics that app dev shops collect concern the overall portfolio and completed projects. Few shops collect in-flight project metrics; many believe that they do, but the metrics that they use are actually just measures of conformance to plan and projections of post-mortem project metrics — for example, on-schedule and on-budget. The lack of in-flight project metrics that really describe the work being performed on a project is a major fault of most application development metrics programs, and it's one that most shops aren't even aware of.

Table 1: Sample Portfolio, Post-Mortem Project, And In-Flight Metrics

Portfolio metrics	In-flight metrics	Post-mortem project metrics
Budget	Test coverage	Cost
Resources	Defect rate	Timeline
Customer satisfaction	Functionality delivered	Scope
Mix of work	Productivity	Customer satisfaction

Source: A study of application development metrics commissioned by Borland Software and conducted by Forrester Consulting

App Dev Shops Struggle To Balance The Costs And Benefits Of Metrics

Application development organizations recognize the value of metrics; however, their use of metrics doesn't compare to that of their colleagues in other parts of IT. The firms with which we spoke usually pointed to IT operations, the call center, and supplier management as poster children for successful use of metrics. The work these organizations do is different from that of the application development group: IT operations measures machines as much as it measures human beings, call centers measure a very small number of highly defined activities, and supplier management groups are in the business of analyzing spend.

Metrics Collection Is Complex And Costly

What prevents application development organizations from using metrics as well as their peers in IT operations, the call center, and supplier management? To find out, Forrester asked app dev managers and executives what obstacles they face in gathering metrics. The interviewees pointed out that:

- **Most app dev metrics collection is performed manually.** The No. 1 obstacle to collection of meaningful metrics is the manual effort required. Nearly half of the 20 people we interviewed cited this as a challenge they face (see Figure 2). Even development shops that obtain their metrics from tools are only able to do so because someone has manually entered the raw data. That burden usually falls either on the project manager or the development lead, who must ask individual contributors for status updates; or on the developers and other team members, who must themselves input data, such as hours spent and task status. In fact, several of the development leads and project managers we spoke with told us they spend nearly a third of their time on metrics collection. Manual metrics collection is incredibly tedious, highly error-prone, and prohibitively expensive. Of course, tools that automate the collection of metrics are expensive, too.

"The problem is that not everyone has a seat on our project management tool. Getting data from a group of 26 people can be a nightmare. I would say that between 20% and 40% of my time is spent gathering metrics." (Brokerage firm)

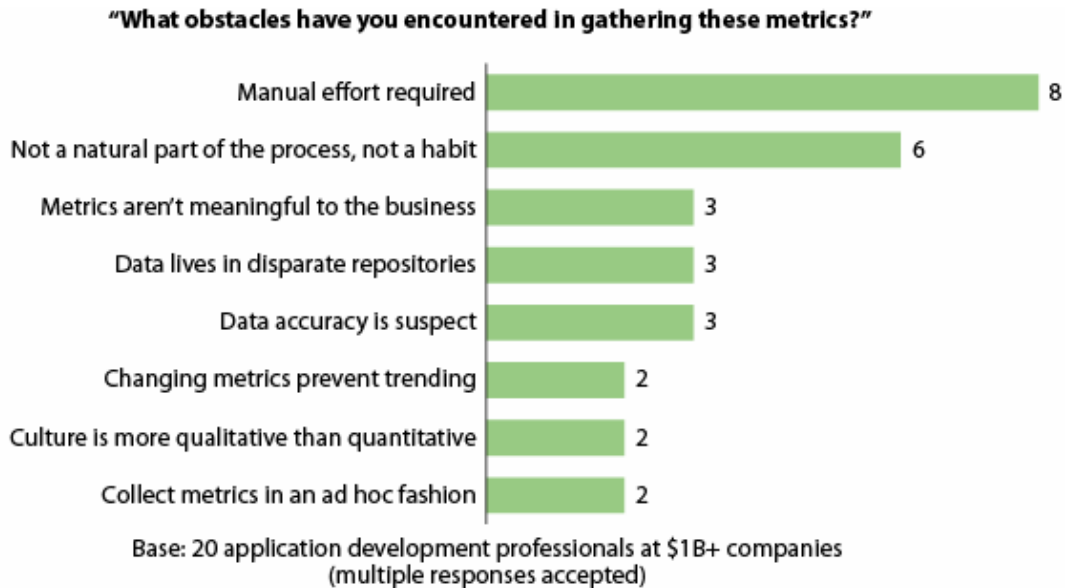
"The time required to gather and analyze the metrics is our biggest problem. The data is in multiple places, so you're pulling it together manually. It can take a few minutes or a few hours, depending on which metrics you're gathering." (Multinational publishing company)

- **Reporting and analyzing app dev metrics is technically complex.** It's in trending and aggregation that most of the value of measurement is found. But eight of the 20 app dev organizations we interviewed as part of this study are unable to trend or aggregate the metrics they collect. This limits their ability to get value out of their investments in measurement. What's holding them back? The problem often stems from disparities introduced when methods of metrics collection change. It's also often rooted in technical challenges caused by tool heterogeneity, requiring shops to pull data from multiple repositories that have fundamentally different methods of storing data. This is a common challenge, since metrics about the full development life cycle will necessarily span multiple development tools — and most likely development tool vendors, as well. Sometimes it's also just a simple lack of reporting and analytics skills that gets in the way. After all, when these skills are in short supply they are arguably better applied to development projects than to reporting about development projects. Even when resources with these skills are available, they come at a price.

“It matters how well the system is integrated and how well you can consolidate across three to four systems. We collect some things on a monthly basis. Ideally we’d do so more often, but they take so much time to gather that it’s not possible.” (Bank)

“We have a few folks administering our tools who can respond to our requests for data. But even when we can get a hold of them, we don’t have a lot of confidence that the inputs correspond to the outputs.” (Diversified financial services company)

Figure 2: Manual Methods Stand In The Way Of Metrics Collection Efforts



Source: 20 1-hour interviews with application development professionals commissioned by Borland Software and conducted by Forrester Consulting

There’s no getting around it: Metrics are expensive, independent of which skills are brought to bear or whether the means of collection are manual or automated. Almost all of the firms we interviewed voiced concerns about balancing the costs and benefits of metrics collection. Consequently, even shops that value metrics are hesitant to gather them because they lack confidence that they’ll get an adequate return from the money they invest. When they do move forward, they often encounter resistance from those who don’t believe in the value of measurement, from those who don’t want to be measured, and from those who don’t want to endure the pains of manual metrics collection.

Application Development Shops Gather Superficial Metrics

Concerns about the potential return on investment in metrics aren’t without cause. Not only are metrics expensive, but when application development organizations do bite the bullet and invest in metrics, they often fail to get the results they’re looking for. Unsure which metrics to gather, app dev professionals more often than not settle for metrics that are easily acquired but aren’t particularly valuable. For example, they have:

- **Post-mortem project metrics that don’t help much on app dev projects.** The most common post-mortem project metrics — on-time, under-budget, and on-scope — are generic project management metrics, unsuitable for application development. Interim on-time and on-budget measurements mean little in a phased life cycle, as nothing is “done” until the app has been tested and prepared for deployment. Managers who have had their hopes of on-time delivery dashed by never-ending test cycles that uncover an infinite

stream of defects know this all too well. Because these metrics are so common, most app dev professionals fail to realize just how subjective they ultimately are, and they place a dangerous amount of faith in them. Executives are even more likely to do this, as they usually lack insight into the methods of metrics collection.

“We ask people if they’re on track, but we don’t have any data that tells us that 60% of the time is gone but only 50% of the code is done. We’re asking, but we don’t have answers.” (Diversified financial services company)

“A lot of managers and executives want to track ‘percent complete’ for tasks and requirements. But it’s meaningless — in development, you don’t know how long a task will take until you’re done with it. A lot of our teams give up and track ‘percent complete’ by dividing number of days passed by number of days allocated!” (Transportation and logistics company)

- **Plenty of data on results, little data on root causes.** App dev managers are usually under pressure to deliver and have trouble finding the time to report on their progress. They make time to report metrics when their bosses demand it, but the data they provide is invariably subjective — an amalgamation of low-level estimates (“I’m 80% done with this requirement”), with a dollop of gut feeling thrown in (“We’ve been slacking on unit testing, so we’ll probably run into trouble come system testing”). The progress that they report is usually accompanied by their own explanations rather than by data indicating exactly why the schedule has slipped or the budget has been blown. Why don’t they use metrics to pinpoint the source of their problems? Because there is a defined set of success criteria for a project, but a myriad of reasons why those criteria might not be met. It takes courage to stop working and measure all the potential sources of trouble when you’re afraid that the very act of measurement might slow you down.

“We want to know not just success rates but also what the success rates are based on. Not just budget and time, but the reality that is letting us be successful or keeping us from being successful. If we’re always missing the end of design date, is it because of bad requirements or not scheduling enough time? We need more metrics that analyze the root cause of the problem, not just an answer.” (Regional financial services company)

- **No meaningful measurements of productivity.** Productivity measurements are incredibly valuable to application development organizations, enabling them to identify trends in their output and to estimate timelines and budgets determined upfront and in the course of the project. But very few application development organizations gather productivity metrics with any degree of accuracy, and most don’t even bother trying. Of the 20 firms interviewed for this study, only two gathered any such data: One was an Agile shop that measured the number of hours spent on stories and their constituent tasks, and the other measured each team’s work in terms of business value, including business value delivered in a given period of time. Other interviewees expressed extreme interest in productivity metrics but despair over the two most common available metrics: lines of code and function points. What’s holding them back? The problem is that software is difficult to size, and traditional measures like lines of code and function points are either crude or too expensive to use on a broad scale. Until an objective and inexpensive technique for sizing software emerges, app dev shops will either go without any such data or settle for developing their own internal units of measure and strive to maintain maximum consistency.

“We need some measure of productivity. We just know if we hit the date or not. We don’t know whether the date itself was even realistic.” (Transportation and logistics company)

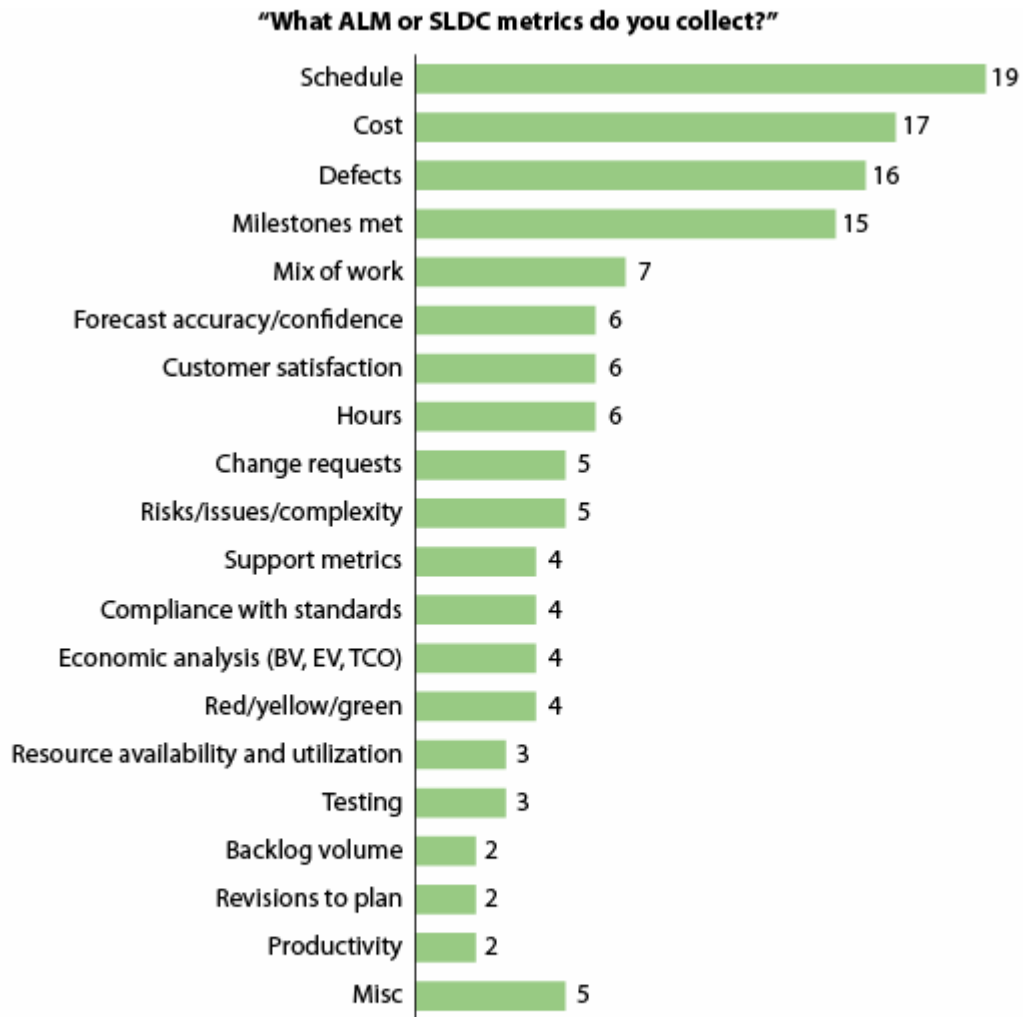
“The problem with measuring quality of estimates is that you might have been right that it would take three years, but it really should have taken one.” (Governmental financial organization)

- **Little accurate data on estimated or actual business value.** Without metrics about business value, application development organizations are unable to communicate with their customers about their contributions to the bottom line or even to prioritize the work in a way that makes real business sense. Few companies are disciplined about the business value of their application development efforts. Among the 20 firms we interviewed, nearly twice as many wanted to gather data on the business value of app dev projects than actually did so. Most business customers quote whatever business value proposals they think they’ll need to secure a green light. These figures are rarely checked after the fact, the business has limited interest in proving the useful nature of the IT organization. Even one exception to this rule — a firm that uses business value as its primary app dev metric — says that it arrives at its business value calculations through “creative accounting.”

“We aggregate IT and sell it to the business units in bulk. So whoever screams the loudest goes to the front of the line and gets service. To be able to better judge which projects we should be doing and what we’re providing back to the business, we need a measure of business value.” (Healthcare company)

“If the implementation is running long or more problems are being encountered, having some sense of business value would allow me to go to management and say that the cost-benefit calculation still supports us on what we’re doing — or that it doesn’t.” (Utility company)

Figure 3: Schedule And Cost Are The Most Commonly Gathered Application Development Metrics



Base: 20 application development professionals at \$1B+ companies (multiple responses accepted)

Source: 20 1-hour interviews with application development professionals commissioned by Borland Software and conducted by Forrester Consulting

The Path To Better, Cheaper Metrics

Two factors — the cost and complexity of metrics collection and the reliance on superficial metrics — conspire to deter application development organizations from attempting to improve their metrics programs. Once application development shops have summoned up the courage to undertake metrics programs, they often select the wrong measures and the wrong collection methods. As a result, they don't receive the expected value. They do, however, incur as much cost as they had expected, if not more. This has the unfortunate effect of souring the organization on metrics altogether. A utility we spoke with explained, "We've been through metrics development efforts that have proven completely worthless. This has led us to take a more qualitative approach."

Transparent Collection Of Actionable Metrics Changes The Cost-Benefit Equation

We asked app dev managers and executives how they know whether they're doing a good job gathering metrics. In response, they described to us what an ideal metrics program could look like. Shops looking to initiate or improve metrics programs can use this ideal metrics program as a target for their own efforts. Here are the attributes our interviewees identified:

- **Metrics that project teams themselves value.** Metrics aren't just for managers and executives. If metrics are really going to change behavior, they need to be appropriate for consumption by individual contributors on development teams as well as managers and executives. In fact, members of development teams should be involved upfront in deciding what in-process metrics matter most to them and how they'll apply them. This establishes team responsibility for the metrics that result and decreases the chance that they'll resist measurement or distrust the data. When the members of the development teams believe in the metrics they've helped to collect, they are much more likely to adapt their behavior when metrics indicate there is a problem. The teams that contribute to the metric collection also get value out of them, and they are more likely to find that the net costs of gathering them are justified.

"The metrics we do without prodding are the ones to focus on because the team itself perceives them as valuable." (Newspaper publishing company)

- **Metrics that enable informed decision-making.** Metrics that aren't actually going to drive decision-making are not valuable metrics. Application development organizations all too often collect particular metrics because they've always done so, because their managers expect them to do so, or because other companies do so. Dashboards are often cluttered with two or three times as many metrics as anyone actually uses, simply for the sake of looking like rich data sources. This is downright wasteful. A successful metrics program is a carefully examined metrics program, where every metric is scrutinized regularly, and the cost of collection is compared with the value captured when it is applied.

"If there's not a decision that's going to be impacted by the metric, then it doesn't need to be gathered, as it's more academic than anything. The reason I'm capturing this metric is because I want to gain visibility into something with an eye toward optimizing it. If after some time of looking at the data I still can't tell what it means, then it's not a good metric." (Transportation and logistics company)

- **An evolving set of metrics.** Successful metrics initiatives start small and are built out over time, often using the goal-question-metric method as a guide. This helps ensure that metrics are gathered only for the sake of measuring progress toward goals, rather than for the sake of measurement alone. Because goals change, metrics should change, as well. For example, we spoke with a telecommunications company that intentionally scales up its measurement efforts in problematic areas or projects and then backs down once the problem has been addressed. High-level metrics remain in place to keep an eye on the situation, but more granular metrics are phased out. Shops that take this type of an evolutionary approach to measurement — carefully weighing the costs and benefits of each new metric — have an easier time keeping the costs and benefits of their metrics program aligned. They are more likely to avoid the pitfall of measuring everything measurable, whether or not it ought to be.

“I’m looking at this as a journey. We’re starting small and evaluating and adjusting and moving on until we get to a point where we have the right number of metrics necessary to identify corrective action.” (Educational publishing company)

“We’ll put metrics in to track certain areas we need to work on. Once it’s fixed, we’ll pull back on the metrics. Then we move along. We hone in on what we culturally, technically, practice-wise want to change and balance it out with an opportunity to bring to conclusion by celebrating conclusion.” (Telecommunications company)

- **Metrics that can be gathered automatically.** Removing the need for manual data entry should be one of the top objectives in a metrics initiative. There are many tools that automatically capture metrics, and there are also some that help aggregate and report on metrics gathered by these tools. This takes much of the pain out of metrics collection and dramatically improves data accuracy. Tool support for the aggregation of data across multiple disparate repositories — and potentially the repositories of underlying tools from multiple vendors — is just as important as tool support for automated metrics collection. Application development organizations should take a hard look at exactly how much time is spent gathering and analyzing metrics and compare the cost of this time with the cost of acquiring, implementing, and maintaining a tool that does some or all of this work automatically.

“We want to minimize the number of human interactions with the system.” (Brokerage firm)

Better Metrics Enable Corrective Action, Not Just Punitive Action

How can application development shops jump-start their metrics programs and make significant progress toward the ideal state outlined above? Two techniques — iterative, incremental development processes and more disciplined estimation of business value — can transform metrics programs. Here’s how:

- **Iterative processes provide more opportunity for low-cost metrics collection.** It’s easier to collect metrics at the end of a project than it is to do so while the project is underway. Of course, these post-mortem metrics are inherently less valuable, as it is too late to change the course of the project. When software development projects are divided into several iterations, however, “post-mortem” metrics can be collected at the end of every iteration. These metrics can be used to judge the health of the project and process and to communicate with the business. If a project has 100 requirements and six iterations and only 20 have been fulfilled by the end of the third iteration, it’s clear that it’s time for intervention. One healthcare company we interviewed works this way; because it also gathers productivity data, it can introduce and then evaluate process changes in terms of their impact on its productivity.

“Our iteration cycle is every three weeks, which means future planning is every three weeks. We then look ahead to the overall release for how we reprioritize things.” (Healthcare company)

“The projects that we have are reasonably long term but we have lots of interim deliverables. You get more immediate feedback at every milestone. ‘Milestones met and planned’ is also a key metric for me.” (Newspaper publishing company)

- **Shared responsibility for business value enables better prioritization.** Of the firms we interviewed, only one was disciplined about the use of business value metrics. That firm, a pharmaceutical company, had made business value the center of its measurement

program. Many firms ask for business value estimates but find that their customers provide wildly inflated figures simply to get their projects approved. At this company, finance actually comes around after project completion to collect the projected savings, or at least to ensure they have been redeployed. As a result, customers try to estimate just enough savings to get their project approved, but not as much as they really expect. The entire IT organization is oriented around these types of metrics, expertise in this subject is common, and there are plenty of people willing to poke and prod — both on the business side and the IT side.

“Business value metrics keep us focused and help us evaluate potential tradeoffs we might want to make. We’re very focused on the deadline, so we may drop features to cut out a month of development time and still get most of the benefit. That’s how we divide the work and chunk it out in planning and execution.” (Pharmaceutical company)

Conclusions

Application development organizations know that metrics matter, but they often fail to implement adequate or meaningful metrics. As a result, they get poor returns from their metrics investments. This convinces them that application development metrics aren’t worth their time and attention, justifying their reliance on superficial data or even just gut feel. This doesn’t have to be the case. To change the cost/benefit equation for metrics, application development organizations should:

- **Refuse to settle for just schedule and budget metrics.** Sticking to traditional measures like on-time and on-budget won’t do much to help you deliver more value to the business. The purpose of these measures is more about predicting project completion than about ensuring it. These metrics are often based on incomplete, inaccurate, or subjective information, and they are insufficient if they are not complemented by metrics specific to the actual activities and deliverables in the application development project. Look for opportunities to gather more concrete data from in-flight projects that will help assess the progress each project is making along the way.
- **Remember that the most accessible metrics aren’t necessarily the best metrics.** The primary concern in determining what to measure should be relevance, not expediency. Even if the cost of collection is minimal, there is still cost involved in monitoring and analysis. The very presence of these less relevant metrics can provide a false sense of comfort or a conviction that metrics aren’t actually useful. Metrics that are already available to you are a great place to start, but be sure that it’s your goals, rather than your tools, that are your primary guide.
- **Improve the accessibility of the metrics you care most about.** The cost of metrics gathering and analysis is not static. Adding tools to support automatic metrics collection and metrics aggregation can drive down the cost of your overall metrics. When combined with more careful metrics selection, this automated collection can change the cost/benefit equation for metrics. This is a necessary step in the implementation of better application life-cycle management practices because proper coordination of functional silos — like business analysis, architecture, development, and testing — requires reporting spanning the coordinated silos.
- **Pay attention to resistance and to level of interest.** Be sensitive to pushback against metrics collection. One head of process improvement told us, “I measure our success by the number of people who come to my office to complain.” If practitioners don’t see value in certain measures, it may be that there isn’t any. To determine whether this is the case, first

ensure that metrics are widely disseminated and accompanied by analysis. Then investigate how metrics are really being used. Are they cited in conversation? Are they used as evidence in debates? Or are they merely dashboard fodder? If it's more the latter than the former, you may not be getting returns from your metrics investment.

Appendix A: Supplemental Material

Background And Methodology

To better understand how application development organizations use metrics, Borland Software commissioned Forrester Consulting to conduct 20 1-hour interviews with application development practitioners, managers, and executives within enterprise IT organizations. Some of the interviews were with more than one respondent. The incentive for participation was a summary of the results of our research.

We interviewed managers and executives in IT organizations that perform a substantial amount of custom application development. Of these respondents, six were directors of application development; four were VPs of application development; three were VPs or directors in charge of the PMO; three were project managers who work on application development projects; two were VPs in charge of IT process improvement and/or measurement; one was a director of business systems analysis; and one was a CIO.

Eight of the respondents worked at companies in the finance and insurance sectors; three in media, entertainment, and leisure; three in utilities and telecommunications; two in the public sector; two in manufacturing; one in retail and wholesale trade; and one in business services. Four of the respondents were from companies with annual revenues between \$1B and \$2B, four from companies with annual revenues between \$2B and \$5B, nine from companies with annual revenues of more than \$5B, and three from large governmental organizations or large companies that do not publicly report their revenues.

The goal of this research was to determine:

- What metrics application development organizations gather today and which metrics they wish they gathered.
- How these metrics are applied and by whom.
- What obstacles application development organizations face when gathering metrics.

Related Forrester Research

See the March 19, 2007, "Role Overview: The Application Development Professional In 2007," report.

See the September 1, 2006, "The Root Of The Problem: Poor Requirements," report.

See the August 18, 2006, "The Changing Face Of Application Life-Cycle Management," report.