1-04-25 Creating and Implementing a Balanced Measurement Program
Dana T. Edberg

Payoff

Although IS measurement programs have long posed challenges in terms of focus and implementation, they can provide IS managers with the opportunity to improve software development and maintenance based on information not intuition. This article presents a framework that helps IS managers ensure that all critical aspects of the IS function are represented in a measurement program and that the metrics developed for each of them are based on the delineation of appropriate goals. Creating this comprehensive, custom-tailored program provides IS managers with the information they need to formulate targeted actions for improvement.

Problems Addressed

It is still unclear why many information systems projects continue to fail, and why some succeed. Understanding the reasons for project success or failure, however, provides IS managers the information they need to form actions that enable the IS function to move forward and improve. The best way to gain this necessary knowledge is from a comprehensive IS measurement program. Measurement is sometimes viewed as an objective in itself rather than as a way of supporting organizational goals. Much of the available advice on the application of measurement to the software development and maintenance process focuses on the intricacies and integrity of specific forms of measurement rather than on understanding the strengths and components of a comprehensive IS measurement program. This article focuses on the latter by describing a flexible measurement framework adaptable to specific organizational requirements. The article:

- Explores the concept of a measurement program.
- Uses the balanced-scorecard structure to develop a framework that serves as a guideline for the measurement procedure.\textsuperscript{17}
- Divides the measurement procedure into manageable components.
- Provides guidelines for the development and successful implementation of a measurement program.

It has been said that the best way to understand the real behavior of software development and maintenance is to institute a measurement program that helps clarify patterns and guides decision making.\textsuperscript{18} Measuring key attributes of software development and maintenance reveals behavior patterns that can be analyzed and interpreted to devise methods that better control and improve these functions.

What is a Measurement Program?

A measurement program is designed to help people understand and interpret an organization's processes. No company, for example, would dream of operating without a way of tracking and analyzing its financial transactions. A financial accounting system is the fundamental data collection process within an organization's financial measurement program. One measurement used by accountants in this program is the accounts-receivable turnover ratio. This ratio indicates how well assets are used by determining how many times accounts receivable is turned over each year. Calculating this ratio requires a system that collects the net credit sales and the average accounts receivable for the organization.

Although executives would not ask a credit manager to improve operations without first determining the current and potentially optimum accounts-receivable turnover ratios, they frequently ask IS managers to improve operations without any idea about important current and projected data and ratios. Whereas extensive computerized tracking systems support the financial and managerial measurement systems that underlie an organization's operations, IS managers are left with intuition and guesswork to control their part of the organization. The historical lack of systems to track and analyze key characteristics of the IS function sets the function apart from other aspects of business operations. Cutting costs or enhancing productivity is thus especially problematic in IS because it is usually accomplished without a detailed picture of expected versus current operations.

A measurement program spans the past, present, and future of a company's operations. Data from past operations are used as a historical baseline to measure present and future processes and projects. Present tasks provide current collection opportunities and data that is analyzed against the historical baseline to guide improvements in future performance. A measurement program thus consolidates data from the past and present to provide the insight that helps IS managers take action for future improvement.

A measurement program requires the formulation of specific, quantifiable metrics. Metrics are quantitative values obtained by measuring characteristics of a system or process. A metric is a piece of information that is analyzed, interpreted, and used to monitor progress toward improvement.

Goals of a Measurement Program

One of the most frequently asked questions about an IS measurement program is, “What are the best metrics?” A more appropriate question is, “What are the goals of IS and how can they be measured?” Understanding and defining the goals of an IS organization helps clarify the metrics that are a part of the measurement program. These goals, and the related goals of the measurement program, must be delineated before the counting begins.

Focusing on goals rather than on metrics is important for several reasons. People notice what is measured in an organization and frequently modify their behavior to make those measurements appear more favorable. Although laboriously counting the number of lines of code written per programmer per day encourages programmers to write more code and results in a lot of code, the code may not solve business problems. Tracking each computer development project and painstakingly updating visibly placed Gantt charts may prod employees to deliver systems on time, but the systems may have many defects that make them difficult to use or maintain. Surveying the satisfaction level of computer users may result in actions that produce happy users, but it could also have the effect of generating a costly and inefficient systems development and maintenance process.

Measurement helps to highlight specific areas in an organization and encourages people to focus their attention and energies on those areas. To leverage this measurement spotlight,
IS managers should identify a set of combined goals that achieve the objectives of the organization. Once goals are established, managers should identify specific questions to be answered, which, in turn, leads to the metrics to be collected. This structure is termed the goal/question/metric (G/Q/M) approach.\textsuperscript{19}

**The Goal/Question/Metric Approach**

The success of the G/Q/M approach is exemplified by the experience of the Motorola Corporation. Managers at Motorola identified seven goals they believed would improve their systems development organization.\textsuperscript{20} For each goal they defined specific questions that needed to be answered to determine whether improvement had occurred. Each question was then defined in terms of an analytical equation, and the variables in the equation were divided into specific metrics that could be collected. For example, one goal was to increase defect containment. The managers defined the following two questions and related metrics to evaluate progress in this area:

- What is the currently known effectiveness of the defect detection process before release? Total defect containment effectiveness equals:
  
  \[
  \frac{\text{number of prerelease defects}}{\text{number of prerelease defects} + \text{number of postrelease defects}}
  \]

- What is the currently known containment effectiveness of faults introduced during each constructive phase of software development for a particular software product? Phase containment effectiveness for phase I equals:
  
  \[
  \frac{\text{number of phase I errors}}{\text{number of phase I errors} + \text{number of phase I defects}}
  \]

Using the G/Q/M approach gave Motorola the opportunity to clarify the semantic use of specific terms such as errors versus defects and the boundaries of given development phases while formulating questions and metrics. (For purposes of clarification, an error is a problem found during testing of the phase in which it was introduced, and a defect is a problem found later than the phase in which it was introduced.) Managers were able to use the information generated from the measurement program to pinpoint errors and defects by software development phase. This information helped them identify areas in the software development process that required changes and to make the needed changes based on information instead of intuition.


What Can a Measurement Program Measure?

The complexity of developing and maintaining business systems makes it difficult to isolate the activities and areas for which goals, questions, and metrics should be formed. The balanced scorecard approach to general business measures encourages managers to expand beyond traditional financial metrics and more thoroughly integrate organizational strategy with resulting performance.  

The balanced scorecard defines four different perspectives for performance measurement:

- **Financial.** How do we look to shareholders?
- **Internal business.** What must we excel at?
- **Innovation and learning.** Can we continue to improve and create value?
- **Customer.** How do our customers see us?

Although these perspectives were originally defined for measuring an entire organization, they can be translated to the IS world. There they respectively are the project, product, process, and performance perspectives.

Perspectives of Performance Measurement

Exhibit 1 depicts the importance of the four perspectives to IS measurement. They are discussed in-depth in the following sections.

Balanced Scorecard Framework for IS Measurement

**Project**

The objective of the project perspective is to understand and measure the characteristics of a specific development or maintenance project by focusing on the attributes that make each project unique. Many organizations use a development or maintenance project as the vehicle to gather data concerning personnel effort and costs. Within the project perspective, an IS organization has the opportunity to use this data to create metrics that provide insight into how funds for software development and maintenance are used. What is measured depends on the scope of the project. The attributes of a given project provide information about such factors as personnel utilization, project estimation, and the nontechnical activities associated with a project.

**Product**

This perspective emphasizes the attributes that differentiate a product from others. The metrics applicable to this perspective are used to understand the growth and progression of a development and maintenance product. It is important to understand the internal scope

---

and attributes of each specific product that composes a system. Managers can use this information to devise new testing procedures, determine individual product defects, or improve the accuracy of product estimation. Because a product frequently lives longer than the project used to create it, product data is gathered over the life of a product. Information generated from the data helps a manager better determine the life expectancy of a given product.

**Process**

The Software Engineering Institute (SEI) capability maturity model has brought renewed focus to the software development and maintenance process. The process perspective highlights the desire to modify the process used to develop and maintain information systems so that procedures reflect the best practices discovered in industry and within a given organization. The measures of this perspective consider organizational and human social interactions, as well as the methodological and technical implications of the development/maintenance process.

**Performance**

The performance perspective measures the outputs of an information system. It encompasses measurements that track both the traditional technical measures of performance as well as metrics that indicate the success of the system as defined by an organization's strategies and policies. Defining this perspective requires defining system success.

**Importance of Balancing the Scorecard**

Using the balanced scorecard approach helps IS managers ensure that all aspects of the IS function are appropriately represented in the measurement program. Instead of focusing on a single area, such as personnel productivity for a specific project, the balanced scorecard provides a framework from which to view each perspective of IS operations.

Using the scorecard thus helps IS managers delineate critical areas and define appropriate goals, questions, and metrics for each of them. The importance of the perspectives varies across organizations. For example, one company may be more interested in the performance of IS as a whole, whereas another might be more concerned about the productivity of project development activities. The balanced scorecard framework does not attempt to dictate the relative emphasis on each area but instead serves as a guideline for managers during development of specific measurement goals.

**Sample Data for IS Metrics**

Once an organization decides to implement a measurement program, the problem is usually deciding what not to measure, rather than deciding what to measure. Many organizations collect data for so many different metrics that participants in the measurement program become cynical and the effectiveness of the program is greatly reduced. In other organizations, the problem is to determine what kinds of data are available to be collected. Exhibit 2 lists categories of metrics for each perspective of the balanced scorecard framework, data for potential metrics in each category, and sample metrics. The exhibit is not comprehensive but rather an abbreviated list of possibilities for each perspective of the framework.
**Metric Categories, Data, and Samples for the Four Perspectives of IS Performance Measurement**

<table>
<thead>
<tr>
<th>Project Perspective</th>
<th>Sample Data</th>
<th>Sample Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Financial, Type and Scope</strong></td>
<td>Total estimated and actual time and estimated and actual cost per predefined project activity, type (e.g., development, maintenance), estimated and actual project function points.</td>
<td>Cost per function point (FP).</td>
</tr>
<tr>
<td><strong>Personnel</strong></td>
<td>Experience level, experience type and education of personnel, years using a specific development environment, number of contractor personnel, number of employees.</td>
<td>Productivity ratings: time/FP for different levels of experience and education.</td>
</tr>
<tr>
<td><strong>Methodology</strong></td>
<td>Type(s) used, level of automation, testing techniques, number of models.</td>
<td>The metrics for methodology are summarized for the entire software process rather than for a particular project.</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>Number of meetings, meeting type and length, number of requirements and design changes, pages of documentation, hours of customer training.</td>
<td>Percent of time in meetings and by meeting type.</td>
</tr>
</tbody>
</table>
### Product Perspective

<table>
<thead>
<tr>
<th>Category</th>
<th>Sample Data</th>
<th>Sample Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Financial, Type and Scope</strong></td>
<td>The same data and metrics used for a project are also applied to a single product. One project could result in many products, or it might take many projects to produce a single product. Product measurements exist over the life of the product, whereas project metrics are closed out when the project is completed.</td>
<td></td>
</tr>
<tr>
<td>Quality</td>
<td>Number of defects and errors, number of test cases, number of change requests, number of changes, amount of usage, complexity rating, number of reused modules, number of support calls.</td>
<td>Number of defects/FP.</td>
</tr>
<tr>
<td>Results</td>
<td>Business objectives translated into quantitative goals.</td>
<td>Inventory percent level.</td>
</tr>
<tr>
<td>Efficiency</td>
<td>Amount of memory, disk, processor cycles, response time, operator time.</td>
<td>Average/peak response time.</td>
</tr>
</tbody>
</table>

### Process Perspective

<table>
<thead>
<tr>
<th>Category</th>
<th>Sample Data</th>
<th>Sample Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Organization</strong></td>
<td>Number of general meetings, type of meetings, communication methods, hours by activity; amount of office and desk space.</td>
<td>Maturity level assessment.</td>
</tr>
<tr>
<td><strong>Personnel</strong></td>
<td>Data in addition to that gathered for a given project include: vacation days taken, vacation days worked, number of working days, number of employees, number of contractors, number of training days.</td>
<td>Metrics in addition to those listed for a given project include:</td>
</tr>
<tr>
<td><strong>Methodology</strong></td>
<td>Gathered by project.</td>
<td>Overall productivity by methodology.</td>
</tr>
</tbody>
</table>
As the exhibit indicates, there are more types of data than there is time available to collect them. A measurement program would not be cost-effective if the data necessary to produce all interesting metrics was collected. The best method is to focus on basic metrics such as size, defects, effort, and user satisfaction before moving on to other metrics.

Under the goal/question/metric approach, the choice of metrics follows the definition of goals and formulation of specific, answerable questions. To achieve a balanced scorecard of measurement, IS managers must ensure that the metrics selected span each of the four perspectives. If one perspective is not measured, a distorted picture of IS may be the result of the measurement program. Because the balanced scorecard framework is used as a guideline to pinpoint areas for control and subsequent improvement, the implications of such a distortion are enormous.

**What Makes an Effective Metric?**

It is wise to consider the criteria commonly employed to judge the usefulness of proposed metrics before selecting metrics that answer the questions deemed important to the goals of IS. A metric should be:

- **Understandable** If a metric is difficult to define or interpret, chances are that it will not be used or it will be applied inconsistently.

- **Quantifiable** Because metrics must be objective, IS managers should strive to reduce the amount of personal influence or judgment that must be attached to a given metric.

- **Cost-Effective** The value of the information obtained from a measurement procedure must exceed the cost of collecting data, analyzing patterns, interpreting results, and validating correctness. A given metric should be relatively easy to capture and compute, and measurement should not interfere with the actual process of creating and delivering information systems.

- **Proven.** Many proposed metrics appear to have great worth but have not been validated or shown to have value in the drive to improve IS. IS managers should steer clear of metrics that appear overly complex or have not been tested and shown to be consistent or meaningful.

- **High Impact** Although some metrics, such as cyclomatic complexity, offer an effective way of predicting testing time and possibly corrective maintenance time, they
may not provide enough information to make their collection and calculation worthwhile in all situations. If the products being measured have relatively similar levels of complexity, it is more helpful to gather metrics with a more significant impact. For example, it is well-documented that one programmer can make a program very complex, whereas another can produce elegant, concise code. The effects of different code on actual testing and correction time, however, pale in comparison to the effects of incomplete or inaccurate design specifications. Therefore, the metric with the most impact in this case relates to the accuracy of design specifications rather than to program complexity.

Implementing a Measurement Program

Although use of a measurement program appears to be a rational management approach backed by documented successes, some organizations find implementation a difficult undertaking. Implementing a measurement program is not a trivial task but rather a significant action requiring management commitment. The two key challenges in implementing a measurement program are time and communication.

Time

A measurement program is not a quick fix for a broken process with benefits that are quickly realized. Data must be gathered and analyzed over time before the program yields information that people can translate into actions that improve the development and maintenance process. It takes time to create a metric baseline, evaluate the results, and choose appropriate new actions. Then it takes additional time to compare new information about those new actions against the baseline to gauge improvements. Implementation of a measurement program is best viewed as a critical component of long-term continuous improvement.

Communication

Part of making a measurement program work is convincing people that it will lead to organizational improvements. If program participants are not convinced of the program’s importance, chances are the effort will be abandoned before meaningful data is collected and used. If people believe that the results of the measurement program will be used to unfairly distribute blame regarding projects and products, then they will not participate in the program.

A key challenge of program implementation is thus communicating prospective benefits to the diverse audiences that will collect, analyze, interpret, and apply the information. At the same time, the proposed use of the measurement information must be made clear to all participants.

Program Activities

Although the success of a measurement program cannot be guaranteed, IS managers can increase the odds that implementation will prevail by paying attention to the individual activities composing the program. Exhibit 3 shows the activities necessary to implement and maintain an IS measurement program. Each activity is described in the sections that follow.
Activities of an IS Measurement Program

**Assessment.**

The three primary functions of assessment are:

- Evaluating the current position of the organization.
- Identifying the goals of a measurement program.
- Establishing specific measurement goals.

Since the mid-1980s, formal software process assessments such as those from the Software Engineering Institute and Software Productivity Research (SPR) have been available to evaluate the software development processes of an organization. Assessment provides a clear picture of the current organizational environment and serves as a starting point from which to gauge future improvements. For example, it would be unreasonable to state that a new development methodology provided increased programmer productivity unless the level of productivity before its implementation was known and documented.

During the assessment phase, it is also important to define the goals of the measurement procedure. Another activity performed during assessment is selling the measurement program to management and IS staff. All participants in the program must understand the relationship between measurement and improvement so that they will support the resulting program.

**Formulation.**

A measurement program requires the formulation of specific, quantifiable questions and metrics to satisfy the program goals. The previously discussed suggestions for choosing appropriate metrics and sample goals/questions/metrics provide a good starting point.

**Collection.**

The collection of specific metrics requires a cost-accounting system aimed at gathering and storing specified attributes that act as input data for the metrics. This process should be automated so that collection takes as little time as possible and the danger of becoming mired in amassing huge amounts of data is avoided. Careful planning in assessment and formulation helps avoid the gathering of too much data.

**Analysis.**

The physical collection of a metric does not, by itself, provide much information that helps in the decision-making process. Just as gross sales do not reveal the financial condition of an organization, the number of function points for a project does not explain how many person-months it will take to produce that project. A metric must be statistically analyzed so that patterns are uncovered, historical baselines are established, and anomalies are identified.
Interpretation.

The function of interpretation is to attach meaning to the analysis; in other words, to determine the cause of the patterns that have been identified during analysis and then prescribe appropriate corrective action. For example, if analysis shows that users are consistently dissatisfied with systems that require an ever-increasing number of user and analyst meetings, then it may not be a good idea to schedule more meetings. A more effective approach is to look for other reasons behind the dissatisfaction. Perhaps the meetings are unproductive, communication skills are ineffective, or business problems are being incorrectly identified. The interpretation of metric analyses furnishes a direction in which to start looking for different problems and solutions.

Validation.

As shown in Exhibit 3, validation occurs throughout each phase of the measurement program. It involves asking a set of questions to ensure that the goals of the measurement program are being addressed. For example, the results of the formulation phase should be validated with two key questions:

- Are we measuring the right attribute?
- Are we measuring that attribute correctly?

The following scenario illustrates how validation questions are applied. Let us assume that one of the overall goals of an IS organization is to improve the performance of user support. Using the G/Q/M approach, the IS organization establishes a goal of improving user satisfaction with IS support. A question that supports this goal is, What is the current level of user satisfaction with IS support? IS personnel then formulate a questionnaire they believe measures the level of user satisfaction with IS support. The questionnaire is used to collect data, which is analyzed and interpreted.

Analysis shows that there is no relationship between the type, amount, or level of IS support and user satisfaction. Why not? It could be because there is no relationship between user satisfaction and IS support, or because the questionnaire was not measuring user satisfaction with IS support. Validating the questionnaire in the formulation phase helps ensure that it is measuring what it intended to measure.

Measurement as a Passive Yet Iterative Process

Measurement is a relatively passive process; it is the actions people take based on the fact that they are being measured and the new procedures that are developed based on the information generated from the measurement program that lead to improvement. The goal of a measurement program is to provide information that can be used for continual improvement of the systems development process and its related products.

Although Exhibit 3 depicts the activities of assessment, formulation, collection, analysis, and interpretation as a sequential, circular process, they are interdependent and not performed sequentially. In the scenario presented in the preceding section, the IS organization found during analysis that the identified metrics were inadequate to determine any patterns. Such a result required validation of the metrics being used and a return to the formulation phase to redefine other metrics that yield information more relevant to the goals.
Managing a Measurement Program

A measurement program is a long-term effort requiring the cooperation and coordination of a broad set of participants. One way to support the program is to establish a metrics infrastructure. A metrics infrastructure includes the following:

- A management method.
- Data collection procedures.
- Ongoing program training.

The management method should incorporate an integrated group/committee that performs each of the measurement activities. This group is similar to a conventional project steering committee and includes representatives from general management, IS management, system users, and IS development/maintenance. It differs from a project management group primarily because it survives past the implementation of the program. As a long-term, ongoing process, measurement must have a long-term, ongoing management committee.

The management group is critical to the success of a measurement program because it keeps program participants aware of the importance of their activities and provides the broad view necessary to the program’s survival. The committee determines how the program’s goals evolve over time and serves as a touchstone for their achievement.

It is also beneficial to establish a metrics user group to share experiences and coordinate training. Training is a key element of the metrics infrastructure that should be periodically provided. Training programs encompass data collection, analysis, and interpretation procedures so that project participants understand not only how to collect data, but how to apply it. The infrastructure should also include tools to automate the collection and analysis phases of measurement, as well as a consolidated data base to store all metrics data.

Because the results of a measurement program are not immediately discernible, people have to perform many tasks before visible, attributable outcome is perceived. Training is one way to ensure that all project participants understand the goals and procedures that underlie what may appear, at times, to be a slow process. It also helps to alleviate employee concerns about the application of the measurement program results.

Many of the activities of measurement require the cooperation of diverse groups of people. Even though the concepts of metrics and measurement conjure up an image of required technical expertise, the most appropriate leader for the measurement program is an individual with excellent communication and negotiation skills.

Information about programs used in other companies is helpful during the process of defining program objectives and formulating metrics. The recommended reading list provides a source for gathering this information.

Recommended Course of Action

Each day new articles describe a new practice touted to yield tremendous productivity and efficiency improvements in the IS organization. Some IS managers have discovered that it can take many years to apply the so-called best practices of other organizations recorded in the literature. A measurement program affords IS managers the opportunity to develop local proof of what really works. More important, the information produced from a
measurement program helps IS managers better understand and control the process of software development and maintenance.

Creating a custom-tailored measurement program thus provides IS managers with information about the unique behavioral patterns of their organization. In doing so, it helps these managers control their professional destinies as well.

**Bibliography**


**Author Biographies**

Dana T. Edberg

Dana T. Edberg is an assistant professor of CIS at the University of Nevada, Reno, with experience in software engineering project management in the government and private sectors.
**Project Perspective**
Provide project management and control for a specific maintenance or development project.

**Performance Perspective**
Analyze, manage, and control the results of IS.

**Product Perspective**
Analyze the efficacy of a given IS product.

**Process Perspective**
Understand, manage, and control the ongoing process of IS development and maintenance.