



# Measurement with a Focus

## Goal-Driven Software Measurement

Dave Zubrow  
Software Engineering Institute

*The collection of accurate metrics is a pointless exercise until the data is analyzed and used to predict and influence future events. The article discusses how to set up a metrics collection and analysis game plan that will advance specific business interests.*

In a recent article by William Schiemann and John Lingle, they describe “Seven Greatest Myths of Measurement.” [1] Among the points made in this article is the need to use measurement to anticipate the future rather than to merely record the past. This is the same perspective promoted by the Software Engineering Institute’s (SEI) Goal-Driven Software Measurement process [2] and the Department of Defense initiative for Practical Software Measurement [3]. The benefit and value of software measurement come from the decisions and actions taken in response to analysis of the data, not from the collection of the data. I liken software measurement activities to potential and kinetic energy: Gathering the data creates a potential, but it takes analysis and action to make it kinetic. The Goal-Driven Software Measurement approach identifies 10 steps to establish a measurement program that is aligned with the organization’s business processes. In this way, the risk of having data gathered, but not used, is minimized.

The steps of the approach are organized into three sets of activities: identifying goals, defining indicators and the data needed to produce them, and creating an action plan to guide the implementation. Business goals are translated into measurement goals [4, 5] by identifying high-level business goals and refining them into concrete, operational statements with a measurement focus. This refinement process involves probing and expanding each high-level goal to derive questions, the answers to which would help manage the organization. The questions provide concrete examples that can lead to statements that identify what type of information is needed.

However, a sense of what information is needed is not specific enough. The goal-driven approach requires that indicators, e.g., charts, tables, or other types of displays and reports, be sketched out and approved by the intended user. These indicators serve as a requirements specification for the data that must be gathered, the processing and analysis that must take place, and the schedule by which these activities should occur. The final set of activities takes the output of the preceding two sets of activities and uses them to develop an action plan. First, the existing data collection and measurement activities within the organization are analyzed to avoid duplication and identify gaps. Priorities, in terms of data to gather to produce the indicators, are assigned. Then, tasks are defined to take advantage of existing activities and to address the gaps. Part of the plan also addresses the need for the measurement activities to evolve with respect to staying synchronized with the organization’s goals and to become more efficient and effective in its own operation. The following sections summarize each of the 10 steps. In summary, the Goal-Driven Software Measurement process consists of the following:

### Identifying Goals

1. Identify your business goals.
2. Identify what you want to know or learn.
3. Identify your subgoals.
4. Identify entities and attributes related to your subgoals.
5. Formalize your measurement goals.

### Defining Indicators

6. Identify quantifiable questions and the related indicators that you will

use to help you achieve your measurement goals.

7. Identify the data elements that you will collect to construct the indicators that help answer your questions.
8. Define the measures to be used and make these definitions operational.

### Creating an Action Plan

9. Identify the actions that you will take to implement the measures.
10. Prepare a plan to implement the measures.

### Identifying Goals

**Step 1: Identify Business Goals**  
The first step in identifying and defining software measures is to identify the business goals that drive your organization’s efforts. If a strategic plan exists and is currently being followed, it can be used as a starting point. It is often worthwhile, however, to check the current commitment to the strategic goals. Without a clear sense of the organization’s strategic goals and the objectives and responsibilities for each work unit or position, there is a risk that measures will not be aligned with important issues within the organization or used. To elicit goal statements, it is sometimes useful to ask a question such as, “What do we want to achieve?” Once the goals have been identified, they need to be prioritized. This is best done in a team setting with the relevant stakeholders participating.

### Step 2: Identify What You Want to Know or Learn

If measurement activities are to be aligned with business goals, the goals must be translated into operational state-

ments. The strategic actions planned and taken with the hope of meeting the goals provide proper targets for measurement. In this step, the goals are linked with knowledge of the organization's business strategies and processes. As illustrated in Figure 1, the questions related to the goals are framed in terms of the entities (work products or activities) and attributes (the size, effort to produce, or quality of an entity) associated with the organization's work processes. Oftentimes, the description of the work processes is in the form of a mental model rather than an explicit definition. If this is the case, it is worthwhile to identify the work products, activities, and other entities that offer opportunities for measurement. The key is taking the time to think through and document what you want to know about those entities with respect to the goals previously identified.

### Step 3: Identify Your Subgoals

The preceding step usually generates many questions. Although they are stimulated by the top-level goal statement, the questions must be focused. By analyzing the questions and seeking commonality among them, subgoals can be derived. The subgoals provide a refinement of the goal and serve as a summary for the questions to which we would like answers. Subgoals are not directly derived from the goals to allow managers and other stakeholders the opportunity to brainstorm about the kinds of information they need with respect to their goals. This helps avoid the tendency to prematurely close the discussion on goals and the information that is needed. Similarly, the grouping and summarization of questions in this step provides a check that the question asked is related to an important dimension or subgoal of the original goal.

### Step 4: Identify Entities and Attributes

In this step, attention is once again turned to the work processes of the organization. The questions from Step 2 are useful in this step as well. As noted, the opportunities to gather data and measure reside in the organization's work processes. The subgoals and related

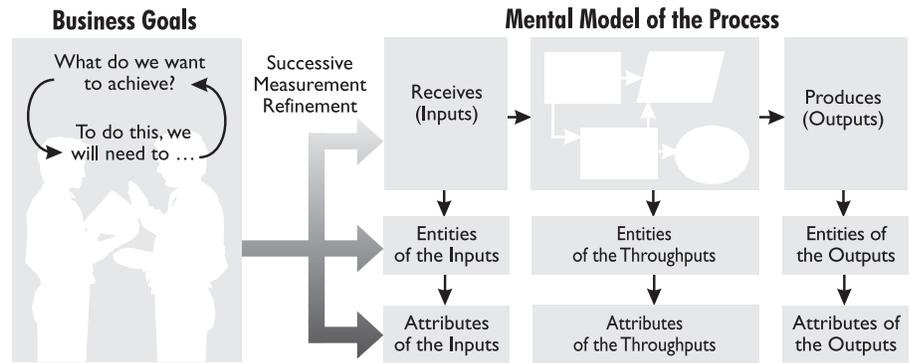


Figure 1. Creating process according to business goals.

questions define the focus for the measures. Careful analysis of the questions will usually help identify what needs to be measured; for example, "How large is our backlog of customer change requests?" The entity in this question is the backlog of change requests and the attribute of interest is its size. Note that at this point, we can further ask in what way should size be measured. This point is addressed in the next step.

### Step 5: Formalize Your Measurement Goals

In this step, a measurement goal is crafted that merges the purpose and perspective derived from the business goal with the possibilities for measurement as they exist within the organization's work processes. In addition, the goal statements express environmental or contextual factors that are important to understand for those who will design and do the measurement and analysis activities.

Well-structured measurement goals have four components:

- An object of interest (an entity).
- A purpose.
- A perspective.
- A description of the environment and constraints.

Note that in the four components, the perspective of who will use the information is explicitly documented. One means to ensure the information gathered will be used is to identify and document the user ("audience" is too passive) for the information.

As an example, the first five steps for defining goals might yield the following:

- Increasing customer satisfaction was identified as a business goal (Step 1).

- Questions asked about increasing customer satisfaction (Step 2) include
  - Do our products satisfy customer requirements?
  - Do we respond to customers in a timely manner?
  - Does our process ensure quality?
- From a set of the questions, a subgoal associated with requirements might be derived. The derived subgoal (Step 3) might be to increase the traceability between requirements and subsequent work products in the development process.
- The entities associated with the derived subgoal (Step 4) include the work products that define and validate the delivered product. The attribute of interest of the work products is the degree to which they address the requirements and perhaps the degree to which they only address the requirements. The latter captures the extent of "unrequired" features.
- Finally, we would address the derived subgoal with a formal measurement goal (Step 5), such as
  - **Object of interest:** The development process.
  - **Purpose:** Assess the degree of traceability of work products to requirements in order to control the scope of development efforts.
  - **Perspective:** Measure traceability of subsequent work products to requirements from the perspective of project managers.
  - **Environment:** New development project for military avionics. Process maturity at the site has been rated at the Repeatable Level of the CMM. Work products follow MIL-STD 2167A.

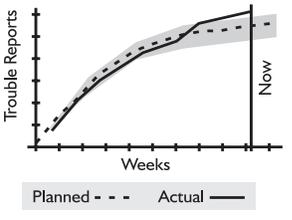
**Indicator Template**

Objective \_\_\_\_\_

Questions \_\_\_\_\_

Assumptions \_\_\_\_\_

Visual Display \_\_\_\_\_



**Inputs**

Data Elements \_\_\_\_\_

Responsibility \_\_\_\_\_

of Reporting \_\_\_\_\_

Form(s) \_\_\_\_\_

Algorithm \_\_\_\_\_

Frequency \_\_\_\_\_

Product  Process

Interpretation \_\_\_\_\_

Cross-References \_\_\_\_\_

Probing Questions \_\_\_\_\_

Evolution \_\_\_\_\_

Figure 2. *Indicator Template.*

## Defining Indicators

### Step 6: Identify Quantifiable Questions and the Related Indicators

Armed with the measurement goal statement, indicators or displays to address the goal can be sketched out. Sketching or drafting the table, chart, or report that needs to be produced helps ensure the requirements for measurement are complete. In the course of designing the indicator, issues regarding the frequency of data gathering, the timing for generating the indicator, the need to use current and historical data, etc., surface. Similarly, the indicator also elicits whether the points on the chart, for instance, represent “raw” values, percentages, or some other derived scale. To a large extent, the indicator represents the product of the measurement activities. It is the consumable for the managers and practitioners who are looking for information to support their decisions and actions. Figure 2 shows an example of a template that can be used to document the definition, inputs, and use of an indicator. Continuing with the previous customer

satisfaction example, an indicator such as the following might be created to answer the question, “What percentage of projects are producing traceability matrices between requirements and other work products?”

### Step 7: Identify the Data Elements

The indicators reflect what data elements are needed. For instance, to produce the preceding indicator, the total number of projects per quarter and the number of projects having traceability matrices per quarter are required. Identifying the data elements, however, is not the same as defining them.

### Step 8: Define the Measures

To continue the customer satisfaction example, definitions are needed for

- Projects.
- Criteria to determine whether they have traceability matrices, i.e., must they be reviewed prior to accepting them for this measure.
- How to assign projects to the periods for reporting, e.g., the quarter in which the project completes its design review.

These definitions are critical to achieve proper interpretations of the data. Note, however, that the definitions need to be created with the purpose of the indicator in mind; that is, they should be consistent with providing an answer to the question that the indicator addresses.

Developing a complete and unambiguous as possible definition can be arduous. To aid this task, the SEI developed a series of measurement framework checklists for common software measures such as size, effort, milestones, and defects [6-8].

## Creating an Action Plan

### Step 9: Identify the Actions for Implementation

Knowing the data needed and having defined them, the existing situation within the organization can be analyzed with respect to your measurement needs. Existing sources of the needed data should be identified. The data elements needed may be found in a variety of

sources including project plans, defect tracking systems, the configuration management systems, and effort reporting systems. Likewise, data that is needed but is not available should be analyzed with respect to the amount of effort required to obtain the data. Considerations at this step include whether new forms, tools, or training would be required to obtain the data. Additionally, you must prioritize the currently unavailable data in terms of the indicators that depend upon the data. For each data element, you should determine its status with respect to the following:

- Does an explicit definition of the measure exist?
- Have the frequency of collection and the points in the process where measurements will be made been determined?
- Has the time line required to move measurement results from the points of collection to databases or users been established?
- Are there forms and procedures to collect and record the data?
- Have storage and access mechanisms and procedures been determined?
- Who is responsible to design and operate the database?
- Who will collect and who can access the data?
- How will the data be analyzed and reported? Who is responsible for the data, and who will receive the reports?
- Have the supporting tools been developed or acquired?
- Has a process guide to collect the data been developed?

In our example, reporting on the existence of traceability matrices may not exist. This gap would then be addressed in the action plan. For instance, to capture this data, the organization may need to add this to a project review or audit checklist for the Software Quality Assurance Group.

### Step 10: Prepare an Action Plan

Once a gap analysis has been completed between the data needed and the existing measurement activities, prepare an action plan. Documenting the tasks to

*See METRICS, Page 15*

want to improve everything. The challenge is to stay focused and use the CMM for software as your guide, but do not attack more than you can handle at one time. Remember: SPI is continuous improvement. It is iterative. Do what you can do in the time allotted, then go back and pick out more things once you have been allocated more time to do them.

## Conclusion

Although there are other points to ponder when attempting this journey down the CMM path, these are the most frequently found errors made that I have documented. Good luck on your journey. ♦

### *METRICS, from page 26*

be performed in an action plan allows the Measurement Team and manager to track progress with respect to the implementation of the measurement activities. An outline for an action plan follows:

- 1.0 Objective.
- 2.0 Description.
  - 2.1 Background.
  - 2.2 Goals.
    - Business Goals.
    - Measurement Goals.
    - The Goals of This Plan.
  - 2.3 Scope.
  - 2.4 Relationship to Other Software Process Improvement Efforts.
  - 2.5 Relationship to Other Functional Activities.
- 3.0 Implementation.
  - 3.1 Activities, Products, and Tasks.
  - 3.2 Schedule.
  - 3.3 Resources.
  - 3.4 Responsibilities.
  - 3.5 Measurement and Monitoring.
  - 3.6 Assumptions.
  - 3.7 Risk Management.
- 4.0 Sustained Operation.

As the measurement activities are being planned, be sure to consider how the quality and success of the measurement activities will be measured. Building the need to measure the quality and success of the measurement activities into the measurement processes will help keep the activities aligned with the needs of the organization and mitigate some of the more common reasons why measurement fails. These reasons include lack of use of the data, personnel not understanding why the data need to be collected, and measurement viewed as an

expendable, overhead activity. Following the goal-driven process outlined above provides a means to involve stakeholders, create understanding, and make measurement a part of the way the organization conducts business. Maintaining alignment between the measurement activities and the information needs of the organization helps the organization leverage information, which may otherwise not be captured, to enhance its performance. In summary, the goal-driven software measurement process directs attention toward measures of importance rather than measures that are merely convenient. ♦

### About the Author



**Dave Zubrow** is team leader for software engineering measurement and analysis for the SEI and is assistant director of analytic studies for Carnegie Mellon University. He is an Association for Software Quality Certified Software Quality Engineer and a member of the Software Division Council for the American Society for Quality Control. He has a bachelor's degree from Penn State University and a master's degree and a doctorate from Carnegie Mellon University.

Voice: 412-268-5243  
 Fax : 412-268-5758  
 E-mail: dz@sei.cmu.edu

### References

1. Schiemann, William and John Lingle, "Seven Greatest Myths of Measure-

### About the Author

**Margaret Kulpa** is a consultant with Abacus Technology Corp. in Chevy Chase, Md. She is a certified lead evaluator and is authorized to teach the SEI's Introduction to CMM and the Software Capability Evaluation class. She has performed SPI duties for over 15 corporations and has evaluated over 30 organizations. She has also written and taught Key Process Area classes for Levels 2 and 3.

Abacus Technology  
 5454 Wisconsin Ave., Suite 1100  
 Chevy Chase, MD 20815  
 Voice: 301-951-1712  
 Fax: 301-907-8508  
 E-mail: kulpamk@songs.sce.com

ment," *IEEE Engineering Management Review*, Spring 1998, pp. 114-116.

2. Park, R., W. Goethert, and W. Florac, *Goal-Driven Software Measurement*, (CMU/SEI 96-HB-002) Software Engineering Institute, Carnegie Mellon University, 1996.
3. PSM96, *Practical Software Measurement: A Guide to Objective Program Insight*, Washington, D.C., Joint Logistics Commanders, Joint Group on Systems Engineering, March 1996.
4. Basili, V. and D. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEEE Transactions on Software Engineering*, Vol. 10, No. 6, 1984, pp. 728-738.
5. Briand, L., C.M. Differding, and H.D. Rombach, "Practical Guidelines for Measurement-Based Process Improvement," *Software Process Improvement and Practices*, Vol. 2, 1996, pp. 253-287.
6. Park, Robert E., et al., *Software Size Measurement: A Framework for Counting Source Statements* (CMU/SEI-92-TR-20), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., September 1992.
7. Florac, William A., et al., *Software Quality Measurement: A Framework for Counting Problems and Defects* (CMU/SEI-92-TR-22), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., September 1992.
8. Goethert, W., et al., *Software Effort Measurement: A Framework for Counting Staff-Hours* (CMU/SEI-92-TR-21), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., September 1992.