Measuring Process Improvement

Tracking your project goals

by Neil Potter and Mary Sakry

There has been much talk in the software industry regarding software process improvement. The emphasis has often been on passing various "tests," such as the ISO certification and SEI CMM Levels. If you are taking one of these routes, you may be wondering if you are actually improving anything—or just learning to jump through new hoops for the purpose of an audit. Process improvement doesn't have to be academic, or solely focused on documentation. It can, and should, be used to solve real problems and make real gains.

QUICK LOOK

- **Keys to tracking your process improvement program**
- **Measurement based on your project's goals and problems**

This article discusses how to track progress in your process improvement program, and offers four questions to consider as you measure your progress: Are we achieving our project goals, solving our development problems, and making progress on our improvement action plans? What are our savings in time and money? Are we making headway on our chosen process model or standard? What lessons have we learned? Answering these questions lets you know how well your improvement program is going, provides visibility early for detecting problems, and gives you data to make your future plans more effective.

We'll also look at some examples of how companies track improvement; you can tailor these examples to fit your needs, or use them as a starting point to generate your own measurement ideas. Our intent is to stimulate you to think about the types of measures that would be useful in your process improvement environment.

Almost all process improvement programs—ISO, CMM, or a homegrown hybrid—are built from similar components. For the purposes of this discussion, we'll use the Shewhart four-step improvement cycle for planning, executing, and managing improvement programs. It divides the process into four phases: *PLAN* (planning the improvement effort), *DO* (executing the plan), *CHECK* (measuring improvements), and *ACT* (acting on the data from the CHECK phase). The DO and ACT phases are subjects for other articles; here our discussion will focus on the PLAN and CHECK components.

Plan: Defining Goals

To measure your progress in anything—a marathon, a weight-loss regimen, or a software process improvement program—you have to have a plan. That plan provides clear standards and goals against which to measure your advancement. In software processes, effective improvement planning is based on the business goals and problems of the organization. Example goals might include the delivery of a product, the completion of a software installation, or the upgrade of a database.

Examining both the goals and problems of an organization can provide an effective scope for any improvement program. This *goal-problem* approach starts with business goals and focuses on eliminating the problems that keep you from reaching those goals. The resulting goal-problem improvement program is compelling and practical. Here are some examples from one company:

1. **Goal: Reduce product development cycle to six to nine months.**
   Problem 1.1: We don't manage changing requirements.
   Problem 1.2: Difficult to find defects before test.

2. **Goal: Successfully deliver product X.**
   Problem 2.1: Wrong files (e.g., DLLs) are put on CD—unsure of the correct ones.
   Problem 2.2: Defect repairs break essential product features.

3. **Goal: The core software product performance is increased by 20 percent.**
   (No identified problems)

4. **Goal: Developers have the essential development tools (latest compiler, memory checker, debugger, and performance analyzer).**
   (No identified problems)

5. **Goal: Customer rating is nine out of ten on product evaluation form.**
   Problem 5.1: Customers are unhappy. There are approximately 300 outstanding defects that have not been addressed.

6. **Goal: Profit remains 15 percent (costs remain the same as last year).**
   (No identified problems)

Measurable improvement plans should clearly describe the goals, the reasons behind those goals, and the actions required to achieve them. The sample improvement plan shown in Figure 1 illustrates these components for our first

Goal and Intermediate Goals *(The results you want)*	Purpose of Goal *(Why do you want to achieve the goal?)*	Actions	Sequence/ Priority *(*=essential)*
GOAL 1: REDUCE PRODUCT DEVELOPMENT CYCLE TO 6–9 MONTHS	**INCREASE MARKET SHARE BY DELIVERING PRODUCT X**		
Intermediate Goal: Manage changing requirements *(based on problem 1.1)*		Only allow changes to the application interface, not the kernel routines.	1*
		Establish a group with the authority for managing the project's software baselines [from SEI CMM Level 2].	2*
		Review the initial requirements and changes before they are incorporated into the project plan [from SEI CMM Level 2].	3
		Buy a requirements management tool.	4
		Record and track change requests and problem reports for all configuration items/units [from SEI CMM Level 2].	5
		Improve the library control system to minimize version control errors.	6
Intermediate Goal: Find defects before test *(based on problem 1.2)*		Hold formal inspections of the seven critical code areas currently delaying the product.	1*

FIGURE 1 Sample improvement plan

goal (from the list above).

The first column lays out the primary and intermediate goals of the improvement project, deriving the intermediate goals from the problem statements (2.1 and 2.2). The next column describes the primary goal's purpose, answering the question "Why do I want to achieve this goal?" or "What benefit does it provide?" The third column lists all the actions that will contribute to the intermediate goals; note that the more involved action steps will need a higher level of detail.

The fourth column records the sequence and priority of your actions. Your focus should be on achieving the intermediate goal you stated, not necessarily on doing all of the actions—so assign "essential" status to no more than 20 percent of each intermediate goal's actions, focusing on the ones that you believe will help you make the greatest progress toward the goal.

If you've laid a clear groundwork of goals in your planning phase, you've made the later phases–including the stages at which you'll be checking your progress using measurements–much easier.

Check: Measuring Your Progress

If you've laid a clear groundwork of goals in your planning phase, you've made the later phases—including the stages at which you'll be checking your progress using measurements—much easier.

As you enter your CHECK phase, your first task is to determine the few high-priority measures you care the most about. Do you care only that progress is being made toward the business goals? Do you also care about which problems are being addressed, or how much money you are spending on each goal? Let's examine the four key questions to ask as you compare your progress to your goals.

1. Are we making progress on the goals, problems, and action plan?

Goal completion: Business goals were the reason for developing the improvement plan during the planning stage. Checking progress toward the completion of each goal determines how effective your improvement effort has been. Checking the goals for achievement is straightforward. Simply check them off on the improvement plan as they are completed.

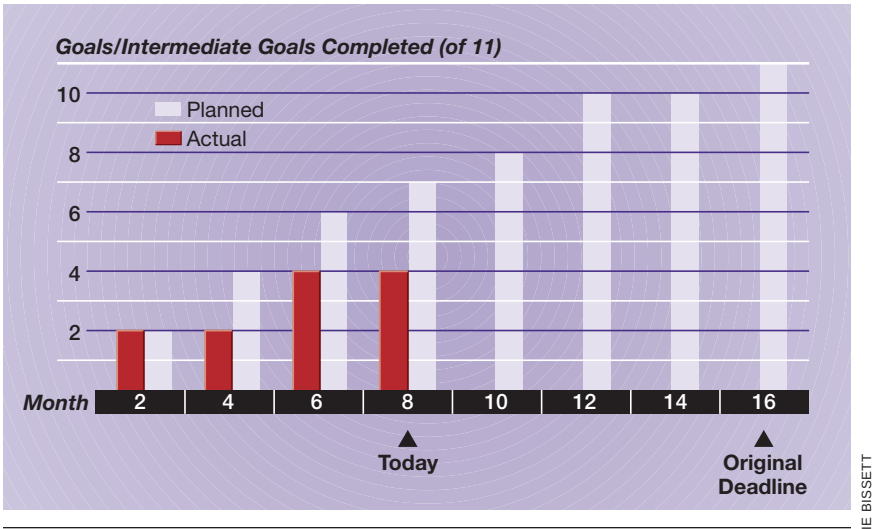
Goals that looked clear when initially written often seem ambiguous when checked for completion. When you check the goal it might need further clarification before anyone can say whether it is complete or not. For example, a goal stating, "Keep customers happy," is difficult to verify. The goal, "Customer rating is nine out of ten," is easier to verify. Do not be surprised if you must tweak the goal during the CHECK phase. With practice, you will be able to write clearer goals the first time.

What if the goal is not yet completed? Some of the goals you check will not have been reached yet. For example, the *improvement actions* for the goal "Successfully deliver product X" might be completed, but the software product might not be ready for delivery. Since the *goal* is not completed, determine if any other improvements are needed to accelerate the goal, or to make it more likely to succeed.

If it's not measurable, make it measurable: If you want to track your goals numerically, select metrics that are representative of the goal and ones for which you can obtain data. For example, one of our clients wished to increase the productivity of their employees. Productivity is defined as the output per unit of time invested in an activity, which is typically measured in size per unit of effort. It can be very difficult to use this measure of productivity across an organization where some people produce design documents, some produce project plans, and others support a help desk. Realizing this problem, the company tracked gross sales per employee. This was something that they considered important and could easily measure. It was also something on which they had existing data.

Problems (intermediate goals) and action plan completion: Problems (described by the intermediate goals) and actions can be checked off the plan as they are completed. Additionally, a trend chart can be used to show the rate of progress (see Figure 2). We recommend that you only chart the number of completed goals and intermediate goals, since charting at an action-item level can be overwhelming.

In Figure 2, we can see that by month eight it is unlikely that the improvement project, consisting of eleven goals and intermediate goals, will be complete by the desired



ANNIE BISSETT

FIGURE 2 Trend diagram tracking goal and intermediate goal completion

deadline of sixteen months. A new plan is needed based on current performance.

2. What are the savings in time or money?

Calculating the savings in time or money helps determine if the time and money invested so far have been valuable. We define savings as:

The benefit of time or money from achieving a goal (or fixing a problem)
–minus–
The investment of time or money in achieving a goal (or fixing a problem)

In the action plan illustrated in Figure 1, one of the problems identified for goal #1 was the organization's need to find defects before test (problem 1.2). An action was included in the plan to inspect selected code.

The chart in Figure 3 shows a team's speed in finding coding defects using inspection compared to system test. For each piece of code that was inspected, the number of defects found per effort-hour of inspection (find rate) was calculated. The find rate is defined as:

Number of major defects found during inspection
–divided by–
Total number of effort-hours used in inspection preparation and inspection logging phase

The average find rate was 1.6 defects per effort-hour, or one defect every 38 effort-minutes of inspection. On average, system test found 0.3 defects per effort-hour, or one defect every 200 minutes. In this case, the team invested US$5,600 and two days' training to learn the inspection process.

The project was able to determine the benefit of this new practice by taking some simple measures. Cost-benefit calculations, similar to the simple one described, allow teams to make informed decisions on which improvements are effective and where they should be applied in the project. In this case, inspections were used to clean up the code going into system test.

When you calculate the savings, keep your calculations simple. Calculate the investments and returns on a small scale and then extrapolate the data based on the duration of the improvement and the number of people impacted by the change. Keep your projections conservative and measure the savings as you deploy the new technique. Since any data you produce is very easy for others to attack, ensure that you list all of your important assumptions.

3. Are we making progress on the process model or standard, if one is being used?

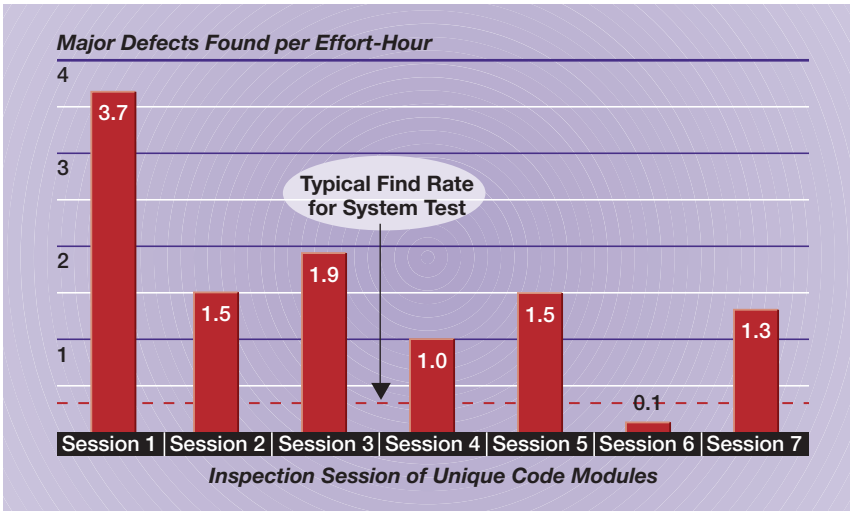
If you have adopted a process improvement model or standard, you will need a method of checking your progress. There are two easy ways to do this. First, by counting the actions completed in the action plan that came from the improvement model or standard. Second, by performing a more thorough check called a "mini-assessment."

Counting the completed actions: To check progress against a process model or standard, determine which model- or standard-related actions are complete. Progress can be measured by counting these actions as a percentage of all the actions required for the model or standard. In the plan shown in Figure 1, three of the actions are SEI CMM Level 2 activities.

Mini-assessment: Another method of tracking progress of an organization against a process model or standard is a mini-assessment. A mini-assessment obtains a quick snapshot of the improvement program. The results show which practices are being adopted and which are not. A mini-assessment is not an audit or a full process assessment, but a friendly check to determine progress.

Before a mini-assessment is conducted, it is important to decide what practices will be checked for adoption. This could include the activities described by the organization's development life cycle, SEI CMM, or ISO9001. A list of questions is developed based on these criteria. For example, does your team:

- Perform inspections or walk-throughs for key work products (e.g., code, design, test cases, plans)?
- Perform black-box testing?
- Perform white-box testing?
- Perform configuration management of all work products (from plans to code)?
- Have adequate computer network stability (compared to the problem reported in the last assessment)?



ANNIE BISSETT

FIGURE 3 Inspection find rate (for seven different code segments) compared to system test

In Figure 4, one organization has tabulated its mini-assessment data to show progress toward achieving SEI CMM Level 2, broken down into 154 individual elements. The responses fall into five categories to show partial satisfaction of the CMM components:

- **Not Applicable**
- **None** (little or no verbal or written evidence)
- **Weak** (current practice or plans are weak or inadequate)
- **Some** (project is approaching intent of Key Process Area [KPA] practice)
- **Strong** (generally speaking, project fulfills CMM intent)

The rough prediction line on the graph predicts that all 154 elements might be complete by the first half of 2004 based on actual progress since 1998. The improvement plan (see "Projected SPI Plan 1.0" on Figure 4), developed for the first half of 2000, shows that more elements will be adopted within the next six months. The rough prediction line is an early warning device to the organization. Its precision is not important in this example; it is clear that the original target date for CMM Level 2, which was April 2000, is impossible to meet.

The mini-assessment process is an effective way to understand which practices have been adopted and which have not. The information is used to understand current gaps, obtain insight on problems with deployment, and provide a basis for replanning the improvement effort.

If you want to learn how well things are going overall with your improvement program, you need to talk to the people who are being asked to change their behaviors.

4. What lessons have we learned so far?

If you want to learn how well things are going overall with your improvement program, you need to talk to the people who are being asked to change their behaviors and adopt new practices. This might include managers, developers, and testers. Lessons-learned data is not numeric, but it can provide actionable ideas that complement the data from the first three measures.

Lessons-learned data comes from interviewing individuals or using discussion groups. While a lesson-learned session can be conducted any time, three specific times are particularly useful: when a *primary goal* has been reached, when an *intermediate goal* has been reached, and when the improvement effort hits an *obstacle*.

Lessons learned are determined using the following steps:

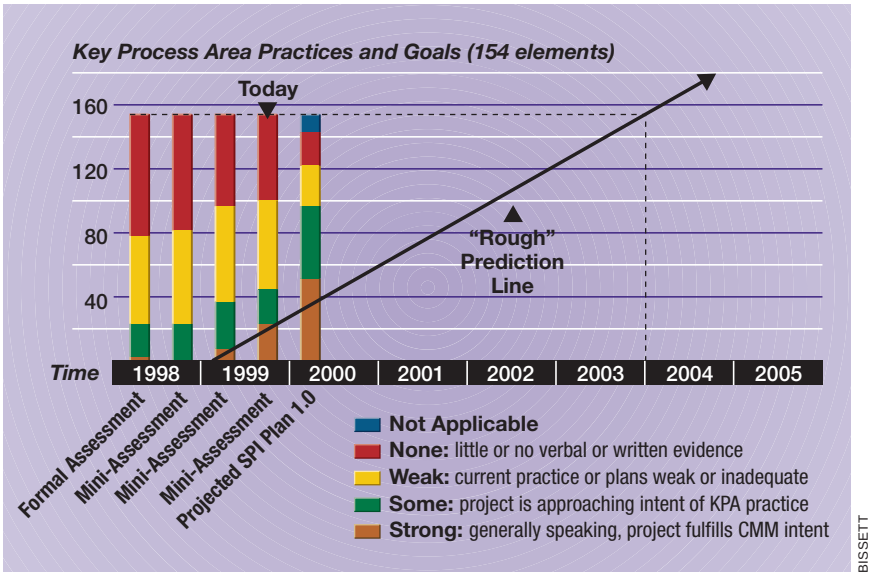
- **Clarify the scope of the session** (e.g., lessons for the deployment of test tool in division Y)
- **Determine what went right** Brainstorm items (actions) that went well
- **Determine areas for improvement** Brainstorm items (actions) that could be improved next time
- **Set priorities** (e.g., high, medium, low)
- **Assign responsibilities**

The process can take between two and four hours, depending on the scope being discussed. As a rule of thumb, break the session into two-hour segments.

When using group interviews, construct the groups to encourage the discussion to be uninhibited. Invite people who are willing to be frank and candid. Select a good objective facilitator—someone not in charge of the improvement effort.

To stimulate the discussion, ask the following additional questions:

- How are the current improvement activities tied to business goals?
- How are the current improvement activities tied to actual problems experienced by the organization?
- What efforts have you seen invested to adopt the new or improved techniques?
- Have the practices been appropriately tailored? Were pilot projects used to ensure their appropriateness?



ANNIE BISSETT

FIGURE 4 Graph from mini-assessment data showing improvement progress

- Is there evidence that things are improving? This can include anecdotal stories as well as some simple metrics.

In Summary

Measuring improvement is essential to providing the organization with feedback while pursuing business goals and solving problems. The resulting data allows for early problem detection, early correction, and improved visibility.

When you establish your measures, consider the following guidelines:

- Look at the goals and problems of your projects. What information do you need to assess where you are with respect to those goals and problems?
- Measure only items you care about.
- Use trend charts to show progress toward each goal.
- Don't overly rely on numerical data. Regularly look at the benefits and lessons learned by the members of the organization.

Making improvement takes time. But by following a Plan-Do-Check-Act cycle and tracking progress, you can make course corrections as you go and increase your chances of meeting your business goals. STQE

Neil Potter and Mary Sakry are co-founders of The Process Group (help@processgroup.com), *a company that consults on software engineering process improvement. Mary Sakry has twenty-three years of experience in software development, project management, and software process improvement. Neil Potter has fourteen years of experience in software design, engineering, and process management.*

Measurement Side Effects

When you measure something, behaviors might change inappropriately because you are taking a measurement. For example, if you measure the number of trouble reports, the number may start going down artificially because people perceive that this information is important and assume you want the number to go down. If you look closely, some people might have started combining reports, and the same number of unfixed problems are still in the product. Similarly, if you measure the number of lines of code developed each day, people might assume that the volume should go up. We once saw a programmer copy dummy code into programs and add jumps around it just to make the productivity figures rise; clearly not what was intended when the measure was instituted.

However you measure progress, it is suggested that all project improvement data be sanitized before being made public. The focus of management should be on the published, organization-wide trends. Removing project names from the data helps avoid some of the bizarre behaviors that occur when individuals optimize their "score." Each team can obtain data about its own progress for the purpose of its own improvement.

Be careful when choosing your measures, communicate their intended use, and be aware of the side effects you might be introducing.