

# Chapter 8

## Measurement and Metrics

---

### CONTENTS

<b>8.1</b>	<b><u>INTRODUCTION</u></b> .....	<b>3</b>
<b>8.2</b>	<b><u>PROCESS DESCRIPTION</u></b> .....	<b>4</b>
8.2.1	<u>DEVELOPING A METRICS PROGRAM PLAN</u> .....	4
8.2.1.1	<u>Define Goals</u> .....	4
8.2.1.2	<u>Derive Questions</u> .....	5
8.2.1.3	<u>Develop Metrics</u> .....	5
8.2.1.4	<u>Define the Collection Process</u> .....	6
8.2.2	<u>IMPLEMENTING A METRICS PROGRAM</u> .....	6
8.2.3	<u>EVALUATING A METRICS PROGRAM</u> .....	6
8.2.4	<u>METRICS REPOSITORY</u> .....	7
<b>8.3</b>	<b><u>MEASUREMENT AND METRICS CHECKLIST</u></b> .....	<b>7</b>
8.3.1	<u>DEVELOPING A METRICS PROGRAM</u> .....	7
8.3.2	<u>METRICS PROGRAM IMPLEMENTATION</u> .....	7
8.3.3	<u>METRICS PROGRAM EVALUATION</u> .....	7
<b>8.4</b>	<b><u>REFERENCES</u></b> .....	<b>8</b>
<b>8.5</b>	<b><u>RESOURCES</u></b> .....	<b>8</b>

This page intentionally left blank.

## Chapter 8

---

# Measurement and Metrics

*"You cannot control what you cannot measure." - Tom DeMarco, 1982.*

Imagine going on a road trip of over a thousand miles. This is easy because most of us have done the real thing several times. Now imagine that your car has no speedometer, no odometer, no fuel gauge, no temperature indicator. Imagine also that someone has removed the mile markers and road signs from all the roads between you and your destination. And just to complete the experiment, remove your watch.

What was once a simple journey becomes an endless series of guesses, fraught with risks. How do you know where you are, how far you've gone, how far to go, when to gas the car, should you stop here or try to make the next town before nightfall. You could break down, run out of gas, be stranded, take the wrong road, bypass your destination, or waste time trying to find where you are and how to reach your destination. Clearly, some method of measuring certain indicators of progress is essential for achieving a goal.

Imagine again going on a road trip. This time the cockpit of the car is filled with instruments. In addition to what you have been accustomed to in the past, there is now speed in feet and yards per second, and as a percentage of *c* (light speed), oil pressure in millibars, estimated time to deplete or recharge the battery, fuel burn rate, fuel weight, oil viscosity and transparency indicators, antifreeze temperature and pressure, engine efficiency, air conditioning system parameters (pressures, temperatures, efficiency), elevation, rate of climb, heading, accelerometers for all directions, indicators for distance and time to destination and from origin, and inside air temperatures for eight different locations in the car. There are instruments to count how many cars pass, vibration levels, and sound pressure levels within and outside the car. There are weather indicators for outside temperature, humidity, visibility, cloud ceiling, ambient light level, true and relative wind speeds and directions, warning indicators for approaching storms and seismic activity, etc. Along the roads will be markers for every hundredth mile and signs announcing exits every quarter mile for five miles before an exit is reached. Speed changes will be announced by signs in 5 MPH increments. To some this may seem like a dream come true, at least the cockpit part. But careful consideration will soon reveal that the driver will be inundated and quickly overwhelmed with unnecessary, confusing data. Measuring things, in itself, is no prescription for achieving a goal. It can even make the goal unattainable.

## 8.1 Introduction

Metrics are measurements of different aspects of an endeavor that help us determine whether or not we are progressing toward the goal of that endeavor. They are used extensively as management tools to provide some calculated, observable basis for making decisions. Some common metrics for projects include schedule deviation, remaining budget and expenditure rate, presence or absence of specific types of problems, and milestones achieved. Without some way to accurately track budget, time, and work progress, a project manager can only make decisions in the dark. Without a way to track errors and development progress, software development managers cannot make meaningful improvements in their processes. The more inadequate our metrics program, the closer we are to herding black cats in a dark room. The right metrics, used in the right way, are absolutely essential for project success.

Too many metrics are used simply because they have been used for years and people believe they might be useful. [2] Each metric should have a purpose, providing support to a specific decision making process. Metrics are too often dictated by leadership. They should be developed by a team under the direction of leadership. Metrics should be used not only by leadership but by all the various parts of an organization or development team. Obviously, all metrics that are useful to managers may not be useful to the accounting people or to developers. Metrics must be tailored to the users. The use of metrics should be defined by a program describing what metrics are needed, by whom, and how they are to be measured and calculated. The level of success or failure of your project will depend in a large part on your use or misuse of metrics – on how you plan, implement, and evaluate an overall metrics program.

While this chapter introduces and describes key metrics ideas and processes and can point you in the right direction, It is recommended you gain more insight, depth, and specific examples by downloading and reading those articles

and books listed in 8.5 Resources that apply to your effort. Of particular worth to DoD users is the *Practical Software and System Measurement Guidebook*.

## 8.2 Process Description

Metrics are not defined and used by themselves, but are part of an overall metrics program. This program should be based on the organization's goals and should be carefully planned, implemented, and regularly evaluated for effectiveness. The metrics program is used as a decision support tool. If the information provided through a particular metric is not needed for determining status or direction of the project, it's probably not needed at all. The role of the metrics program in the organization and its three major activities are shown in Figure 8-1.

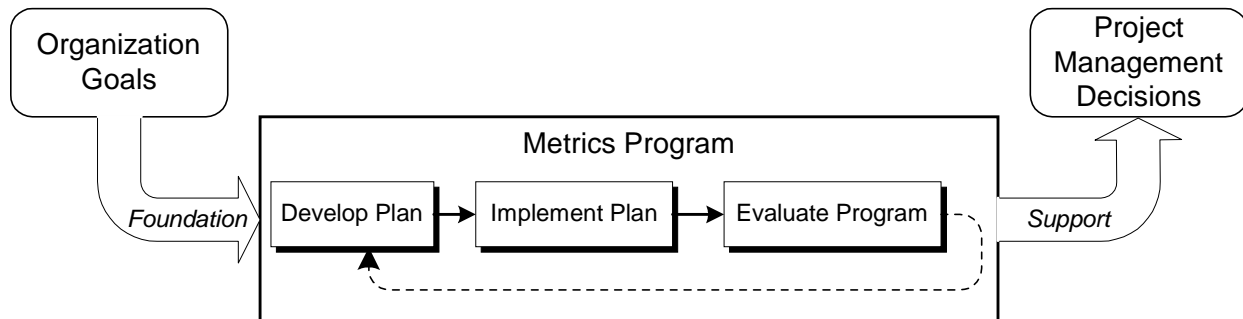


Figure 8-1 Metrics Program Cycle

### 8.2.1 Developing a Metrics Program Plan

The first activity in developing a metrics program is planning. Metrics planning is usually based on the goal-question-metric (GQM) paradigm developed by Victor Basili.

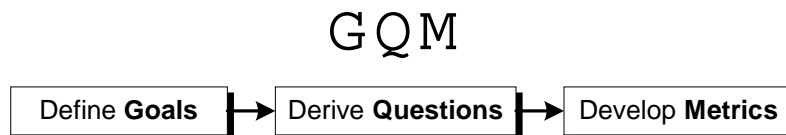


Figure 8-2 Basili's GQM Paradigm

The GQM paradigm is based on the following key concepts: [1]

1. Processes, including software development, program management, etc., have associated goals.
2. Each goal leads to one or more questions regarding the accomplishment of the goal.
3. Each question leads to one or more metrics needed to answer the question.
4. Each metric requires two or more measurements to produce the metric. (e.g. miles per hour, budget spent vs. budget planned, temperature vs. operating limits, actual vs. predicted execution time, etc.)
5. Measurements are selected to provide data that will accurately produce the metric.

The planning process is comprised of the three sub-activities implementing the GQM paradigm and one other, defining the data collection process. Each of these is discussed below.

#### 8.2.1.1 Define Goals

Planning begins with well-defined, validated goals. Goals should be chosen and worded in such a way that they are verifiable, that is, their accomplishment can be measured or observed in some way. Goals such as meeting a specific delivery schedule are easily observable. Requirements stating "software shall be of high quality" are highly subjective and need further definition before they can be used as valid goals. You may have to refine or even derive your

own goals from loosely written project objectives. The selection or acceptance of project goals will determine how you manage your project and where you put your emphasis. Goals should meet the following criteria:

- They should support the successful accomplishment of the project's overall, or system-level, goals.
- They should be verifiable, or measurable in some way.
- They should be defined in enough detail that they are unambiguous.

### 8.2.1.2 Derive Questions

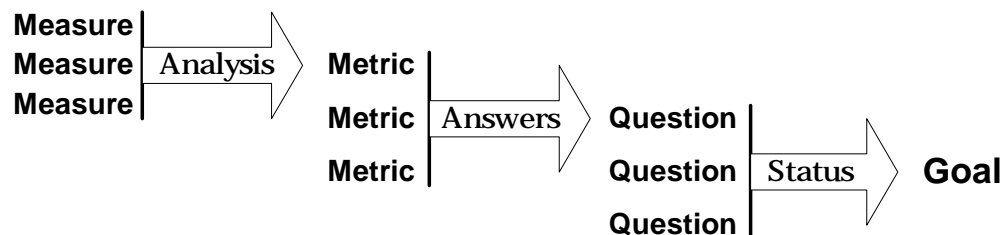
Each goal should evoke questions about how its accomplishment can be measured. For example, completing a project within a certain budget may evoke questions such as these: What is my total budget? How much of my budget is left? What is my current spending rate? Am I within the limits of my spending plan? Goals related to software time, size, quality, or reliability constraints will evoke different questions. It should be remembered that different levels and groups within the organization may require different information to measure the progress they are interested in. Questions should be carefully selected and refined to support the previously defined project goals. Questions should exhibit the following traits:

- Questions only elicit information that indicates progress toward or completion of a specific goal.
- Questions can be answered by providing specific information. (They are unambiguous.)
- Questions ask all the information needed to determine progress or completion of the goal.

Once questions have been derived which elicit only the complete set of information needed to determine progress, metrics must be developed which will provide that information.

### 8.2.1.3 Develop Metrics

Metrics are the information needed to answer the derived questions. Each question can be answered by one or more metrics. These metrics are defined and associated with their appropriate questions and goals. Each metric requires two or more measurements. Measurements are those data which must be collected and analyzed to produce the metric. Measurements are selected which will provide the necessary information with the least impact to the project workflow. Figure 8-3 summarizes the process of turning measurements into goal status.



**Figure 8-3 Flow of Measurements to Goal Status**

Choosing measures is a critical and nontrivial step. Measurements that require too much effort or time can be counterproductive and should be avoided. Remember, just because something can be measured doesn't mean it should be. An in-depth introduction to measurements, *Goal-Driven Software Measurement—A Guidebook*, has been published by the Software Engineering Institute and is available as a free download at:

[www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html](http://www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html)

In addition to choosing what type of data is to be collected or measured, the methods of processing or analysis must also be defined in this step. How do you turn the measurements into a meaningful metric? How does the metric then answer the question? The analysis method should be carefully documented. Don't assume that it's obvious.

This activity is complete when you know exactly what type of data you are going to collect (what you are going to measure and in what units), how you are going to turn that data into metrics (analysis methods), and in what form (units, charts, colors, etc.) metrics will be delivered.

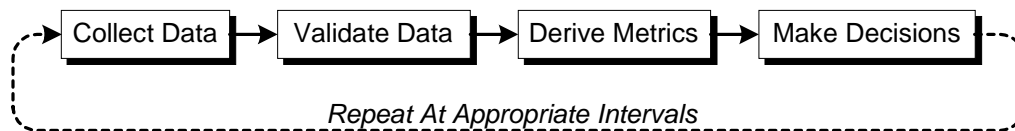
### 8.2.1.4 Define the Collection Process

The final step of the metrics planning process is to determine how the metrics will be collected. As a minimum, this part of the plan should include the following:

- What data is to be collected.
- What the source of the data will be.
- How it is to be measured.
- Who will perform the measurement.
- How frequently the data should be collected.
- Who the derived metrics will be delivered to, and in what format.

### 8.2.2 Implementing a Metrics Program

If the metrics program is well planned, implementing the program should be reduced to simply following the plan. There are four activities in the metrics implementation cycle, shown in Figure 8-4. [2]



**Figure 8-4 Metrics Implementation Cycle**

Data is collected at specific intervals according to the plan. Data is then validated by examining the data to ensure it is the result of accurate measurements, and that the data collection is consistent among members of the group if it is being collected by more than one individual. In other words, is it being measured in the same way, at the same time, etc.? Once the data is determined to be valid, the metrics are derived by analyzing the data as documented in the metrics program plan. Metrics are then delivered to appropriate individuals and groups for evaluation and decision-making activities. This process is repeated until the project is complete.

### 8.2.3 Evaluating a Metrics Program

It is likely a metrics program will not be perfect in its first iteration. Soon after its initial implementation and at regular intervals after that, the metrics program should be evaluated to determine if it is meeting the needs of the metrics users and if its implementation is flowing smoothly. If metrics prove to be insufficient or superfluous, the program plan should be modified to provide the necessary information and remove any unneeded activity. The objective of a metrics program is to provide sufficient information to support project success while keeping the metrics program as simple and unobtrusive as possible. The following are areas that should be considered when reviewing a metrics program:

- Adequacy of current metrics
- Superfluity of any metrics or measures
- Interference of measurements with project work
- Accuracy of analysis results
- Data collection intervals
- Simplification of the metrics program
- Changes in project or organization goals

### 8.2.4 Metrics Repository

A final consideration is establishing a metrics repository, where metrics history is kept for future projects. The availability of past metrics data can be a goldmine of information for calibration, planning estimates, benchmarking, process improvement, calculating return on investment, etc. As a minimum, the repository should store the following:

- Description of projects and their objectives.
- Metrics used.
- Reasons for using the various metrics.
- Actual metrics collected over the life of the each project.
- Data indicating the effectiveness of the metrics used.

## 8.3 Measurement and Metrics Checklist

This checklist is provided to assist you in developing a metrics program, and defining and using metrics. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action. The checklist items are divided into three areas: developing, implementing, and reviewing a metrics program.

### 8.3.1 Developing a Metrics Program

- 1. Is your use of metrics based on a documented metrics program plan?
- 2. Are you using the GQM paradigm in developing your metrics?
- 3. Are your metrics based on measurable or verifiable project goals?
- 4. Do your goals support the overall system-level goals?
- 5. Are your goals well defined and unambiguous?
- 6. Does each question elicit only information that indicates progress toward or completion of a specific goal?
- 7. Can questions be answered by providing specific information? (Is it unambiguous?)
- 8. Do the questions ask for all the information needed to determine progress or completion of the goal?
- 9. Is each metric required for specific decision-making activities?
- 10. Is each metric derived from two or more measurements (e.g. Remaining budget vs. schedule)?
- 11. Have you documented the analysis methods used to calculate the metrics?
- 12. Have you defined those measures needed to provide the metrics?
- 13. Have you defined the collection process (i.e. what, how, who, when, how often, etc.)?

### 8.3.2 Metrics Program Implementation

- 14. Does your implementation follow the metrics program plan?
- 15. Is data collected the same way each time it is collected?
- 16. Are documented analysis methods followed when calculating metrics?
- 17. Are metrics delivered in a timely manner to those who need them?
- 18. Are metrics being used in the decision making process?

### 8.3.3 Metrics Program Evaluation

- 19. Are the metrics sufficient?
- 20. Are all metrics or measures required, that is, non-superfluous?

- ❑ 21. Are measurements allowing project work to continue without interference?
- ❑ 22. Does the analysis produce accurate results?
- ❑ 23. Is the data collection interval appropriate?
- ❑ 24. Is the metrics program as simple as it can be while remaining adequate?
- ❑ 25. Has the metrics program been modified to adequately accommodate any project or organizational goal changes?

## 8.4 References

- [1] Perkins, Timothy K., “The Nine-Step Metrics Program,” *Crosstalk Magazine*, February 2001: [www.stsc.hill.af.mil/crosstalk/2001/feb/perkins.asp](http://www.stsc.hill.af.mil/crosstalk/2001/feb/perkins.asp)
- [2] Augustine, Thomas, et al, “An Effective Metrics Process Model,” *Crosstalk Magazine*, June 1999: [www.stsc.hill.af.mil/crosstalk/1999/jun/augustine.asp](http://www.stsc.hill.af.mil/crosstalk/1999/jun/augustine.asp)

## 8.5 Resources

Army Software Metrics Office, Computer Tutorials (under “Products”): [www.armysoftwaremetrics.org/](http://www.armysoftwaremetrics.org/)

*Crosstalk Magazine*: [www.stsc.hill.af.mil/crosstalk/](http://www.stsc.hill.af.mil/crosstalk/)

- “Metrics Tools: Effort and Schedule”: [www.stsc.hill.af.mil/crosstalk/1995/mar/metrics.asp](http://www.stsc.hill.af.mil/crosstalk/1995/mar/metrics.asp)
- “Project Recovery ... It Can Be Done”: [www.stsc.hill.af.mil/crosstalk/2002/jan/lipke.asp](http://www.stsc.hill.af.mil/crosstalk/2002/jan/lipke.asp)
- “New Air Force Software Metrics Policy”: [www.stsc.hill.af.mil/crosstalk/1994/apr/xt94d04a.asp](http://www.stsc.hill.af.mil/crosstalk/1994/apr/xt94d04a.asp)
- “Universal Metrics Tools”: [www.stsc.hill.af.mil/crosstalk/1995/sep/universa.asp](http://www.stsc.hill.af.mil/crosstalk/1995/sep/universa.asp)
- “Software Metrics Capability Evaluation Methodology and Implementation”:  
[www.stsc.hill.af.mil/crosstalk/1996/jan/metrics.asp](http://www.stsc.hill.af.mil/crosstalk/1996/jan/metrics.asp)
- “Metrics Tools”: [www.stsc.hill.af.mil/crosstalk/1995/feb/metrics.asp](http://www.stsc.hill.af.mil/crosstalk/1995/feb/metrics.asp)
- “Really Bad Metrics Advice”: [www.stsc.hill.af.mil/crosstalk/1998/aug/backtalk.asp](http://www.stsc.hill.af.mil/crosstalk/1998/aug/backtalk.asp)
- “A Methodic Approach to Effective Metrics”: [www.stsc.hill.af.mil/crosstalk/1994/oct/xt94d10c.asp](http://www.stsc.hill.af.mil/crosstalk/1994/oct/xt94d10c.asp)
- “Why the New Metrics Policy”: [www.stsc.hill.af.mil/crosstalk/1994/apr/xt94d04b.asp](http://www.stsc.hill.af.mil/crosstalk/1994/apr/xt94d04b.asp)
- “Metrics: Problem Solved?”: [www.stsc.hill.af.mil/crosstalk/1997/dec/metrics.asp](http://www.stsc.hill.af.mil/crosstalk/1997/dec/metrics.asp)
- “Metrics Tools: Size”: [www.stsc.hill.af.mil/crosstalk/1995/apr/metrics.asp](http://www.stsc.hill.af.mil/crosstalk/1995/apr/metrics.asp)
- “Quantitative Software Management Workshop - Software Cost and Schedule Estimation”:  
[www.stsc.hill.af.mil/crosstalk/1996/oct/xt96d10h.asp](http://www.stsc.hill.af.mil/crosstalk/1996/oct/xt96d10h.asp)
- “Software Quality Metrics for Object-Oriented Environments”:  
[www.stsc.hill.af.mil/crosstalk/1997/apr/quality.asp](http://www.stsc.hill.af.mil/crosstalk/1997/apr/quality.asp)
- “Metrics for Predicting Run-Time Failures and Maintenance Effort”:  
[www.stsc.hill.af.mil/crosstalk/1998/aug/predicting.asp](http://www.stsc.hill.af.mil/crosstalk/1998/aug/predicting.asp)
- “Metrics for Ada 95: Focus on Reliability and Maintainability”:  
[www.stsc.hill.af.mil/crosstalk/1995/may/ada95.asp](http://www.stsc.hill.af.mil/crosstalk/1995/may/ada95.asp)
- “Pro-Active Metrics”: [www.stsc.hill.af.mil/crosstalk/1998/aug/proactive.asp](http://www.stsc.hill.af.mil/crosstalk/1998/aug/proactive.asp)
- “Best Measurement Tool Is Your Telephone”: [www.stsc.hill.af.mil/crosstalk/2001/mar/lucero.asp](http://www.stsc.hill.af.mil/crosstalk/2001/mar/lucero.asp)
- “Metrics Tools: Software Cost Estimation”: [www.stsc.hill.af.mil/crosstalk/1995/jun/metrics.asp](http://www.stsc.hill.af.mil/crosstalk/1995/jun/metrics.asp)
- “Software Maintainability Metrics Models in Practice”:  
[www.stsc.hill.af.mil/CrossTalk/1995/nov/Maintain.asp](http://www.stsc.hill.af.mil/CrossTalk/1995/nov/Maintain.asp)
- “Earned Value Project Management”: [www.stsc.hill.af.mil/crosstalk/1998/jul/value.asp](http://www.stsc.hill.af.mil/crosstalk/1998/jul/value.asp)

*Guidelines for the Successful Acquisition and Management of Software-Intensive Systems (GSAM)*, Version 3.0, Chapter 13 & Appendix N, OO-ALC/TISE, May 2000. Download at: [www.stsc.hill.af.mil/gsam/guid.asp](http://www.stsc.hill.af.mil/gsam/guid.asp)

Literate Programming Software Metrics, many resources: [www.literateprogramming.com/fmetrics.html](http://www.literateprogramming.com/fmetrics.html)

NASA Software Assurance Technology Center: <http://satc.gsfc.nasa.gov/support/>

- Recommended code metrics, by language: <http://satc.gsfc.nasa.gov/metrics/codemetrics/index.html>



- “Software Metrics and Reliability”:  
[http://satc.gsfc.nasa.gov/support/ISSRE\\_NOV98/software\\_metrics\\_and\\_reliability.html](http://satc.gsfc.nasa.gov/support/ISSRE_NOV98/software_metrics_and_reliability.html)

*Practical Software and Systems Measurement Guidebook*, under “Downloads”: [www.psmc.com](http://www.psmc.com)

SEI Software Engineering Measurement and Analysis: [www.sei.cmu.edu/sema/welcome.html](http://www.sei.cmu.edu/sema/welcome.html)

- “Software Measurement for DOD Systems: Recommendations for Initial Core Measures”:  
[www.sei.cmu.edu/publications/documents/92.reports/92.tr.019.html](http://www.sei.cmu.edu/publications/documents/92.reports/92.tr.019.html)
- “Software Quality Measurement”: [www.sei.cmu.edu/publications/documents/92.reports/92.tr.022.html](http://www.sei.cmu.edu/publications/documents/92.reports/92.tr.022.html)
- “Software Size Measurement”: [www.sei.cmu.edu/publications/documents/92.reports/92.tr.020.html](http://www.sei.cmu.edu/publications/documents/92.reports/92.tr.020.html)
- “Goal-Driven Software Measurement—A Guidebook”:  
[www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html](http://www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html)
- “Software Effort & Schedule Measurement”:  
[www.sei.cmu.edu/publications/documents/92.reports/92.tr.021.html](http://www.sei.cmu.edu/publications/documents/92.reports/92.tr.021.html)
- “Software Cost and Schedule Estimating”:  
[www.sei.cmu.edu/publications/documents/94.reports/94.sr.003.html](http://www.sei.cmu.edu/publications/documents/94.reports/94.sr.003.html)
- “Practical Software Measurement”:  
[www.sei.cmu.edu/publications/documents/97.reports/97hb003/97hb003abstract.html](http://www.sei.cmu.edu/publications/documents/97.reports/97hb003/97hb003abstract.html)

This page intentionally left blank.