THE GOAL QUESTION METRIC APPROACH

Victor R. Basili[1] Gianluigi Caldiera[1] H. Dieter Rombach[2]

([1]) Institute for Advanced Computer Studies
Department of Computer Science
University Of Maryland
College Park, Maryland

([2]) FB Informatik
Universität Kaiserslautern
Kaiserslautern, Germany

1. INTRODUCTION

As with any engineering discipline, software development requires a measurement mechanism for feedback and evaluation. Measurement is a mechanism for creating a corporate memory and an aid in answering a variety of questions associated with the enactment of any software process. It helps support project planning (e.g., How much will a new project cost?); it allows us to determine the strengths and weaknesses of the current processes and products (e.g., What is the frequency of certain types of errors?); it provides a rationale for adopting/refining techniques (e.g., What is the impact of the technique XX on the productivity of the projects?); it allows us to evaluate the quality of specific processes and products (e.g., What is the defect density in a specific system after deployment?). Measurement also helps, during the course of a project, to assess its progress, to take corrective action based on this assessment, and to evaluate the impact of such action.

According to many studies made on the application of metrics and models in industrial environments (*see article "Software Measurement"*), measurement, in order to be effective must be:

1. Focused on specific goals;

2. Applied to all life-cycle products, processes and resources;

3. Interpreted based on characterization and understanding of the organizational context, environment and goals.

This means that measurement must be defined in a top-down fashion. It must be focused, based on goals and models. A bottom-up approach will not work because there are many observable characteristics in software (e.g., time, number of defects, complexity, lines of code, severity of failures, effort, productivity, defect density), but which metrics one uses and how one interprets them it is not clear without the appropriate models and goals to define the context.

There are a variety of mechanisms for defining measurable goals that have appeared in the literature: the Quality Function Deployment approach [9], the Goal Question Metric approach [4, 5, 6, 7], and the Software Quality Metrics approach [8,10]. This article will present the Goal Question Metric approach and provide an example of its application.

2. THE GOAL QUESTION METRIC APPROACH

The Goal Question Metric (GQM) approach is based upon the assumption that for an organization to measure in a purposeful way it must first specify the goals for itself and its projects, then it must trace those goals to the data that are intended to define those goals operationally, and finally provide a framework for interpreting the data with respect to the stated goals. Thus it is important to make clear, at least in general terms, what informational needs the organization has, so that these needs for information can be quantified whenever possible, and the quantified information can be analyzed a to whether or not the goals are achieved.

The approach was originally defined for evaluating defects for a set of projects in the NASA Goddard Space Flight Center environment. The application involved a set of case study experiments [7] and was expanded to include various types of experimental approaches [6]. Although the approach was originally used to define and evaluate goals for a particular project in a particular environment, its use has been expanded to a larger context. It is used as the goal setting step in an evolutionary quality improvement paradigm tailored for a software development organization, the Quality Improvement Paradigm, within an organizational framework, the Experience Factory (*see article*), dedicated to building software competencies and supplying them to projects.

The result of the application of the Goal Question Metric approach application is the specification of a measurement system targeting a particular set of issues and a set of rules for the interpretation of the measurement data. The resulting measurement model has three levels:

1. Conceptual level (GOAL): A goal is defined for an object, for a variety of reasons, with respect to various models of quality, from various points of view, relative to a particular environment. Objects of measurement are

- Products: Artifacts, deliverables and documents that are produced during the system life cycle; E.g., specifications, designs, programs, test suites.

- Processes: Software related activities normally associated with time; E.g., specifying, designing, testing, interviewing.

- Resources: Items used by processes in order to produce their outputs; E.g., personnel, hardware, software, office space.

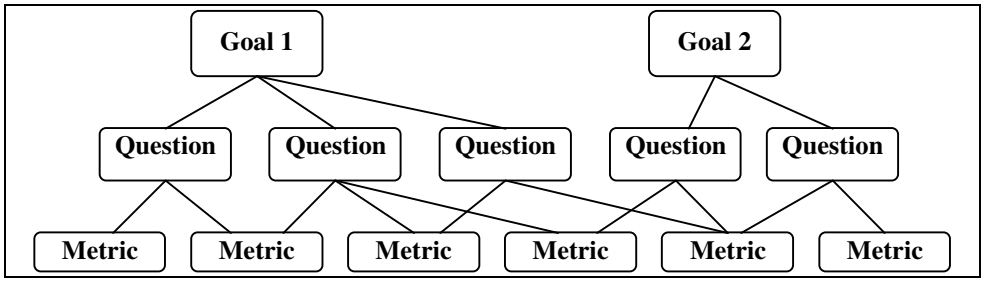
2. Operational level (QUESTION): A set of questions is used to characterize the way the assessment/achievement of a specific goal is going to be performed based on some characterizing model. Questions try to characterize the object of measurement (product, process, resource) with respect to a selected quality issue and to determine its quality from the selected viewpoint.

3. Quantitative level (METRIC): A set of data is associated with every question in order to answer it in a quantitative way. The data can be

- Objective: If they depend only on the object that is being measured and not on the viewpoint from which they are taken; E.g., number of versions of a document, staff hours spent on a task, size of a program.

- Subjective: If they depend on both the object that is being measured and the viewpoint from which they are taken; E.g., readability of a text, level of user satisfaction.

Figure 1



A GQM model is a hierarchical structure (Figure 1) starting with a goal (specifying purpose of measurement, object to be measured, issue to be measured, and viewpoint from which the measure is taken). The goal is refined into several questions, such as the one in the example, that usually break down the issue into its major components. Each question is then refined into metrics, some of them objective such as the one in the example, some of them subjective. The same metric can be used in order to answer different questions under the same goal. Several GQM models can also have questions and metrics in common, making sure that, when the measure is actually taken, the different viewpoints are taken into account correctly (i.e., the metric might have different values when taken from different viewpoints).

In order to give an example of application of the Goal/Question/Metric approach, let's suppose we want to improve the timeliness of change request processing during the maintenance phase of the life cycle of a system. The resulting goal will specify a purpose (improve), a process (change request processing), a viewpoint (project manager), and a quality issue (timeliness). This goal can be refined to a series of questions, about, for instance, turn-around time and resources used. These questions can be answered by metrics comparing specific turn-around times with the average ones. The complete Goal/Question/Metric Model is shown in Figure 2.

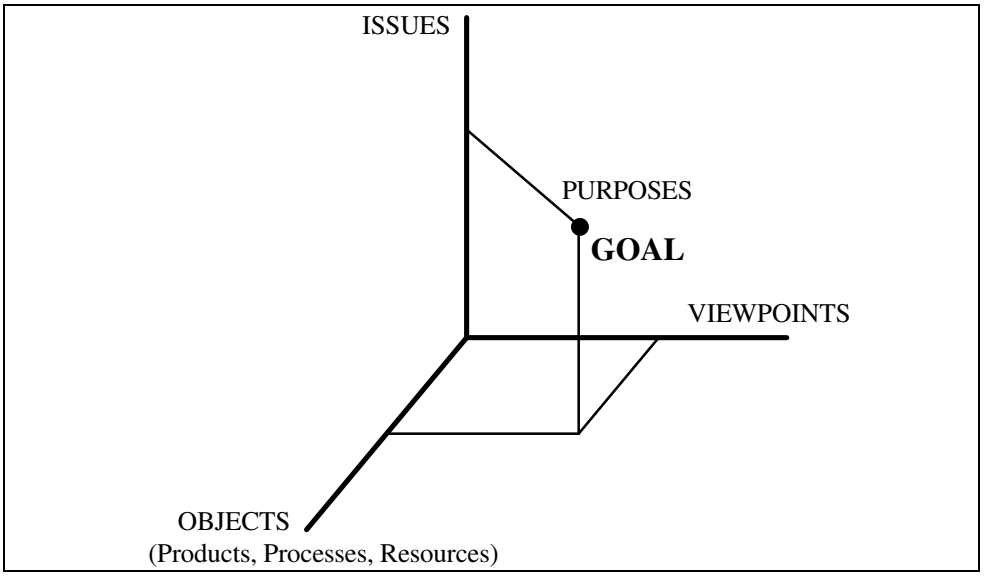
Figure 2

Goal	Purpose Issue Object (process) Viewpoint	Improve the timeliness of change request processing from the project manager's viewpoint
Question		What is the current change request processing speed?
Metrics		Average cycle time Standard deviation % cases outside of the upper limit
Question		Is the performance of the process improving?
Metrics		$\frac{\text{Current average cycle time}}{\text{Baseline average cycle time}} * 100$ Subjective rating of manager's satisfaction

3. THE GOAL QUESTION METRIC PROCESS

A GQM model is developed by identifying a set of quality and/or productivity goals, at corporate, division or project level; e.g., customer satisfaction, on-time delivery, improved performance. From those goals and based upon models of the object of measurement, we derive questions that define those goals as completely as possible. For example, if it is to characterize a software system (e.g., an electronic mail package, a word processor) with respect to a certain set of quality issues (e.g., portability across architectures), then a quality model of the product must be chosen that deals with those issues (e.g., list of functional features that can be implemented in different architectures). The next step consists in specifying the measures that need to be collected in order to answer those questions, and to track the conformance of products and processes to the goals. After the measures have been specified, we need to develop the data collection mechanisms, including validation and analysis mechanisms.

Figure 3



The process of setting goals is critical to the successful application of the GQM approach and it is supported by specific methodological steps. As illustrated in Figure 3 and in our example in the last section, a goal has three coordinates:

1. Issue — Timeliness
2. Object (process) — Change request processing
3. Viewpoint — Project manager

and a purpose:

- Purpose — Improve

Therefore, the development of a goal is based on three basic sources of information.

The first source is the policy and the strategy of the organization that applies the GQM approach. From this source we derive both the issue and the purpose of the Goal by analyzing corporate policy statements, strategic plans and, more important, interviewing relevant subjects in the organization.

The second source of information is the description of the process and products of the organization, or, at least, the ones that are within the scope of the measurement we want to perform. If, for instance, we want to assess a process, we need a model of that process and of the component sub processes. From this source we derive the object coordinate of the Goal by specifying process and product models, at the best possible level of formality.

The third source of information is the model of the organization, which provides us with the viewpoint coordinate of the Goal. Obviously, not all issues and processes are relevant for all viewpoints in an organization, therefore we must perform a relevancy analysis step before completing our list of goals, in order to make sure that the goals that we have defined have the necessary relevancy.

In this way, we end up with a specification of our goals that takes into account the structure and the objective of the organization. From the specification of each goal we can derive meaningful questions that characterize that goal in a quantifiable way. In general, we will ask at least three groups of questions:

Group 1. How can we characterize the object (product, process, or resource) with respect to the overall goal of the specific GQM model?

Example:

Question	What is the current change request processing speed?
Question	Is the (documented) change request process actually performed?

Group 2. How can we characterize the attributes of the object that are relevant with respect to the issue of the specific GQM model?

Example:

Question	What is the deviation of the actual change request processing time from the estimated one?
Question	Is the performance of the process improving?

Group 3. How do we evaluate the characteristics of the object that are relevant with respect to the issue of the specific GQM model?

Example:

Question	Is the current performance satisfactory from the viewpoint of the project manager?
Question	Is the performance visibly improving?

Once the questions have been developed, we proceed to associating the question with appropriate metrics. The factors we consider in doing this are many; among them:

- Amount and quality of the existing data: we will try to maximize the use of existing data sources if they are available and reliable;
- Maturity of the objects of measurement: we will apply objective measures to more mature measurement objects, and we will use more subjective evaluations when we deal with informal or unstable objects
- Learning process: GQM models need always refinement and adaptation, therefore the measures we define must help us in evaluating not only the object of measurement but also the reliability of the model used to evaluate it.

Taking into account these ideas, we can complete our example GQM model with some appropriate metrics. The result is shown in Figure 4.

Figure 4

Goal	Purpose Issue Object (process) Viewpoint	Improve the timeliness of change request processing from the project manager's viewpoint
Question	Q1	What is the current change request processing speed?
Metrics	M1 M2 M3	Average cycle time Standard deviation % cases outside of the upper limit
Question	Q2	Is the (documented) change request process actually performed?
Metrics	M4 M5	Subjective rating by the project manager % of exceptions identified during reviews
Question	Q3	What is the deviation of the actual change request processing time from the estimated one?
Metrics	M6 M7	$\frac{\text{Current average cycle time - Estimated average cycle time}}{\text{Current average cycle time}} * 100$ Subjective evaluation by the project manager
Question	Q4	Is the performance of the process improving?
Metrics	M8	$\frac{\text{Current average cycle time}}{\text{Baseline average cycle time}} * 100$
Question	Q5	Is the current performance satisfactory from the viewpoint of the project manager?
Metrics	M7	Subjective evaluation by the project manager
Question	Q6	Is the performance visibly improving?
Metrics	M8	$\frac{\text{Current average cycle time}}{\text{Baseline average cycle time}} * 100$

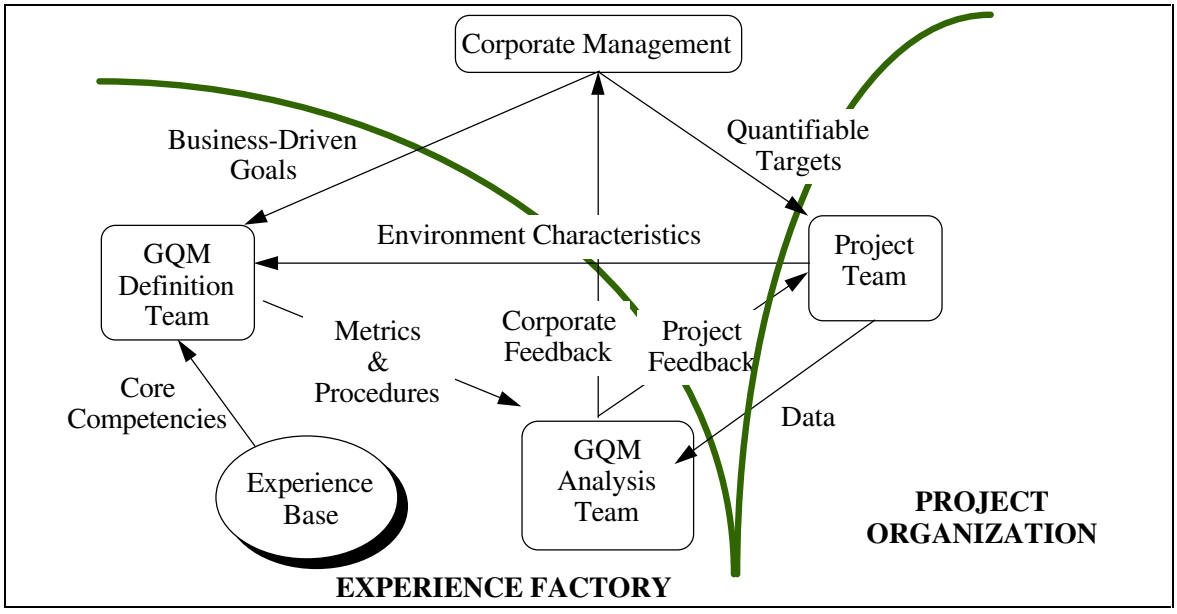
Once a GQM model has been developed, we will select the appropriate data collection techniques, tools and procedures. The data that will be collected will e mapped into the model and interpreted according to schemes previously defined by the organization.

4. CONCLUSION

In summary, the Goal Question Metric approach is a mechanism for defining and interpreting operational and measurable software. It can be used in isolation or, better, within the context of a more general approach to software quality improvement. In this last case the development of GQM models is a task performed by the experience factory which will use as inputs to the process the business driven goals provided by the corporate

management and the environment characteristics provided by the project team. Figure 5 outlines the basic roles and flows of information for this model.

Figure 5



The Goal Question Metric approach combines in itself most of the current approaches to measurement and generalizes them to incorporate processes and resources as well as products. This makes it adaptable to different environments, as as confirmed by the fact that has been applied in several organizations , e.g., NASA, Hewlett Packard [12], Motorola, Coopers & Lybrand.

REFERENCES AND FOLLOW-UP READING

[1] V. R. Basili, "Data Collection, Validation, and Analysis," in *Tutorial on Models and Metrics for Software Management and Engineering*, IEEE Catalog No. EHO-167-7, 1981, pp. 310-313.

[2] V.R. Basili, "Quantitative Evaluation of Software Engineering Methodology," *Proceedings of the First Pan Pacific Computer Conference*, Melbourne, Australia, September 1985.

[3] V.R. Basili, "Software Development: A Paradigm for the Future", *Proceedings of the 13th Annual International Computer Software & Applications Conference (COMPSAC)*, Keynote Address, Orlando, FL, September 1989.

[4] V.R. Basili, "Software Modeling and Measurement: The Goal Question Metric Paradigm," *Computer Science Technical Report Series*, CS-TR-2956 (UMIACS-TR-92-96), University of Maryland, College Park, MD, September 1992.

[5] V.R. Basili, H. D. Rombach, "The TAME Project: Towards Improvement-Oriented Software Environments," *IEEE Transactions on Software Engineering*, vol.SE-14, no.6, June 1988, pp.758-773

[6] V.R. Basili, R.W. Selby, "Data Collection and Analysis in Software Research and Management," *Proceedings of the American Statistical Association and Biomeasure Society*, Joint Statistical Meetings, Philadelphia, PA, August 1984.

[7] R. Basili, D. M. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEEE Transactions on Software Engineering*, vol. SE-10, no.6, November 1984, pp. 728-738.

[8] W. Boehm, J. R. Brown, and M. Lipow, "Quantitative Evaluation of Software Quality," *Proceedings of the Second International Conference on Software Engineering*, 1976, pp.592-605.

[9] M. Kogure, Y. Akao, "Quality Function Deployment and CWQC in Japan," *Quality Progress*, October 1983, pp.25-29.

[10] J. A. McCall, P.K. Richards, G.F. Walters, "Factors in Software Quality," Rome Air Development Center, RADC TR-77-369, 1977.

[12] R.B. Grady, D.L. Caswell, *Software Metrics - Establishing a Company-wide Program*, Prentice Hall, 1987.