Fraunhofer Einrichtung
Experimentelles
Software Engineering

Practical guidelines for measurement-based process improvement

Authors
Lionel C. Briand
Christiane M. Differding
H. Dieter Rombach

IESE-Report No. 004.96/E
Version
Dec. 31, 1996

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competetive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach
Sauerwiesen 6
D-67661 Kaiserslautern

Practical Guidelines for Measurement-Based Process Improvement[1]

Lionel C. Briand, Christiane M. Differding, and H. Dieter Rombach[2]

Abstract:

Despite significant progress in the last 15 years, implementing a successful measurement program for software development is still a challenging undertaking. Most problems are not of theoretical but of methodological or practical nature. In this article, we present lessons learned from experiences with goal-oriented measurement. We structure them into practical guidelines for efficient and useful software measurement aimed at process improvement in industry. Issues related to setting measurement goals, defining explicit measurement models, and implementing data collection procedures are addressed from a practical perspective. In addition, guidelines for using measurement in the context of process improvement are provided.

Keywords: software measurement, Goal Question Metric paradigm, process improvement

1. Introduction

Software measurement is widely recognized as an effective means to understand, monitor, control, predict, and improve software development and maintenance projects. However, effective software measurement requires that a great deal of information, models, and decisions be documented. Thus, it is a particularly difficult task for people who do not have extensive experience with software measurement.

Of particular interest to us is goal-oriented measurement [Rom91, BCR94], a strategy that consists of deriving models and measures from measurement goals in order to ensure the consistency and completeness of measurement plans. More precisely, our guidelines will be defined in the particular context of the Goal Question Metric paradigm (GQM) [BW84, BR88, Rom91, Bas93]. The main motivation for this paper is the lack of practical guidelines for planning, implementing, and using goal-oriented software measurement for process improvement. Based on our experience with measuring software development products and processes in the context of continuous improvement programs, we want to provide more guidance to people performing measurement programs in the context of the GQM paradigm. Therefore, we propose structured guidelines to address the issues most commonly encountered and practical insights into the main GQM concepts. Our guidelines are based in part upon standard literature about the GQM paradigm. Nevertheless, in many instances, we do not comply with the original GQM definitions and templates since we adapted and refined them based on experience and projects' feedback. We tried, however, to provide a complete set of guidelines integrating existing, new, and updated material.

---

[1] To be published in the Software Process Improvement and Practice Journal, Vol. 2(4), 1997

[2] L. Briand and D. Rombach are with the Fraunhofer Institute for Experimental Software Engineering, Sauerwiesen 6, D-67661 Kaiserslautern, Germany. C. Differding and D. Rombach are with the Software Engineering Group, Department of Computer Science, University of Kaiserslautern, D-67653 Kaiserslautern, Germany.

Section 2 provides motivation for goal-oriented measurement. Section 3 gives an overview over the goal-oriented measurement process. Section 4 addresses the issues related to defining relevant measurement goals in an organization. The structure of GQM measurement plans, as we see them, is described in Section 5. Their implementation and all related practical issues are discussed in Section 6. Section 7 provides some insight into various strategies for data analysis. The interpretation of results is then described in Section 8. Finally, Section 9 identifies common types of measurement-based actions for improving the development process.

2. Motivation for Goal-Oriented Measurement

Measurement is introduced in software organizations to gain quantitative insight into the development processes and the developed products. This is important in order to understand better the development process, to identify problems and improvement opportunities. Measurement activities are commonly referred to as measurement programs. A measurement plan should specify the why, what, how, and who of a measurement program. Unfortunately, in too many instances, the motivation part is overlooked.

Goal-oriented measurement is the definition of a measurement program based on explicit and precisely defined goals that state how measurement will be used. In addition, explicit models have to be defined to support the derivation of measures from the goals in a traceable and unambiguous manner. Three main categories of models may be required: descriptive, evaluation, and predictive models. For example, there may be a measurement goal dealing with productivity. In this case, a *descriptive* model is needed to define operationally what productivity is in the context of this measurement goal and environment, and what are the underlying modeling assumptions. Another measurement goal may purport to determine whether a component has a sufficient level of quality (e.g., based on a combined analysis of its complexity, coupling, and cohesion) to go into configuration management. In this case, an *evaluation* model would be required. Finally, to provide an estimated value for a dependent variable based on independent variables, e.g., project effort based on project size, team experience, and other influential project characteristics, one would need to specify and build a *predictive* model.

Advantages of goal-oriented measurement are:

- Goal-oriented measurement helps ensure adequacy, consistency, and completeness of the measurement plan and therefore of data collection. The designer(s) of a measurement program (referred to as measurement analysts) must deal with a large amount of information and numerous interdependencies. In order to ensure that the set of measures is adequate, consistent, and complete, the measurement analysts need to know precisely why attributes are measured (e.g., size needs to be measured to predict project cost early in the development process), what are the underlying assumptions (e.g., no code reuse), and in which models measures are intended to be used (e.g., regression model, COCOMO-like model).

- Goal-oriented measurement helps manage the complexity of the measurement program. Increased complexity occurs when there are too many attributes to measure and too many possible measurement scales for each attribute. In addition, the way attributes are adequately measured (i.e., their operational definition) is strongly dependent on the goal of measurement and therefore a measurement plan without a clear goal-driven structure rapidly becomes unmanageable. Without a structure that captures the interdependencies, changes are likely to introduce inconsistencies into the measurement plan.
- In addition, goal-oriented measurement helps stimulate a structured discussion and promote consensus about measurement and improvement goals. In turn, this helps define widely accepted measures and models within an organization, a crucial prerequisite for measurement success.

Summary Table: Motivations for goal-oriented measurement
• Ensure adequacy, consistency, and completeness of measurement plan
• Deal with the complexity of measurement programs
• Stimulate a structured discussion about measurement

3. Process for Goal-oriented Measurement

Goal-oriented measurement is performed through six major steps which are briefly described below. For more details, see [GHW95, Bas95**].** The process steps will be used as reference points throughout the paper so that the guidelines we provide can be mapped back into this measurement process.

Step 1: Characterize the environment. Identify relevant characteristics of the organization and of the project(s) to be measured. Typical questions are: What kind of product is being developed? What process is being used? What are the main problems encountered during projects? This characterization is mainly qualitative in nature even though previously existing data may be reused.

Step 2: Identify measurement goals and develop measurement plans. Define the measurement goals based on the information gathered during Step 1. For each measurement goal derive the important attributes to be measured by involving project personnel and management. Document the definition of the measures and their underlying motivations in the measurement plan.

Step 3: Define data collection procedures. For all measures identified during the second step, data collection procedures have to be defined, i.e., how and when the data has to be collected and who will collect it. To optimize data collection procedures and limit data collection effort, the development process is a major element to take into account.

Step 4: Collect, analyze and interpret data. Collect project data, analyze them and interpret the analysis results with the help of project personnel and management.

Step 5: Perform post-mortem analysis and interpret data. Analyze data further by taking into account a broader view than the project itself, e.g., by comparing the project results with the organization baseline. Identify the lessons learned in the project.

Step 6: Package experience. Structure and store documents, data analysis results, and lessons learned concerning the project and its measurement program in a reusable form.

4. Definition of Measurement Goals

In this section, we introduce a modifed version of the GQM goal templates to guide the definition of measurement goals and discuss its main influencing factors. The section provides guidelines regarding Step 2 of the process for goal-oriented measurement.

4.1. Applying GQM Templates to Define Measurement Goals

Practice has shown the importance of specifying a measurement goal precisely since the selection and definition of suitable and useful measures and models depends strongly on the clarity of these early decisions [BBC+96, BR88]. GQM provides templates for defining measurement goals in a precise way. This section describes the important aspects of a modified version of these templates**.** The purpose dimension has been particularly modified to make an easier mapping to the model categories in Section 5.2.3.

GQM templates structure a measurement goal based on five aspects:

- The *object of study* defines the primary target of the study, i.e., the process or product that will be analyzed. Examples of objects are the entire development process, phases like system test, and documents like the design document, or the final product.
- The *purpose* of the study expresses why the object will be analyzed. Common purposes, in increasing order of difficulty, are:
    - *Characterization* aims at forming a snapshot of the current state/performance of the software development processes and products.
    - *Monitoring* aims at following the trends/evolution of the performance/state of processes and products.
    - *Evaluation* aims at comparing and assessing the quality of products and the efficiency/effectiveness of processes.
    - *Prediction* aims at identifying relationships between various process and product factors and using these relationships to predict relevant external attributes [Fen91] of products and processes.
    - *Control* and *change* aim at identifying causal relationships that influence the state/performance of processes and products. *Control* consists in influencing the course of a project in order to alleviate risks. On the other hand, *Change* implies modifying the process from project to project in order to improve quality or productivity. *Change* requires a finer grain understanding of the phenomena under study than *control*.

- The *quality focus* states the particular attribute of the object of study that will be characterized, evaluated, predicted, monitored, controlled, or changed. Examples for quality focuses are cost, reliability, correctness, defect removal, changes, user friendliness, maintainability, etc.
- The *viewpoint* identifies the roles or positions of the people who are going to use the output of the measurement program, e.g., who interprets the data collected and uses the prediction models. These people are expected to provide strong input into the definition of the measurement program. Examples for *viewpoints* are project leader, developer, system tester, quality assurance manager, user, corporation, etc.
- The *context* of the study specifies the environment in which the study will be performed and correspondingly determine how generalizable the results will be. The information contained in the context is used to make environmental influential factors explicit, e.g., team structure and experience, application domain.

These five dimensions are summarized in Table 1. They specify completely a measurement goal [BCR94]. An example of a measurement goal using the GQM goal template is:

Analyze *the final product*
for the purpose of *characterization*
with respect to *reliability*
from the viewpoint of *the tester*
in the context of *Project X*

Table 1: Dimensions of the measurement goal templates

Dimension	Definition	Examples
Object of Study	What will be analyzed	development process, system test, design document, final product,...
Purpose	Why will the object be analyzed	characterization, evaluation, prediction, monitoring, control, change
Quality Focus	What property/attribute of the object will be analyzed	reliability, cost, correctness, defect removal, changes, user friendliness, maintainability, ...
Viewpoint	Who uses the data collected	project leader, developer, system tester, quality assurance manager, user, high-level management, ...
Context	In which environment	project X, in corporation A, ...

Every measurement goal can be expressed using this template. Goals should not cluster more than one purpose, quality focus, or viewpoint. Even though they may require similar data, this is likely to create confusion. The underlying assumption of the GQM paradigm is that it is easier

to cope with the complexity of a measurement program by clearly specifying and separating goals of measurement. Merging goals is therefore likely to be counterproductive.

4.2. Factors Affecting the Definition of Measurement Goals

Two types of factors and their respective impact on GQM goals are described in this section: improvement goals and the development process under study.

4.2.1. Impact of Improvement Goals

Software development organizations may need to

- reduce their cycle time and/or cost because of market constraints or customer pressure
- improve the quality of their software, e.g., their reliability, because of the critical application of their software systems
- gain more control over their projects through more accurate management

From such *improvement goals*, one may derive *measurement goals* that help achieve these improvement goals, e.g., identify costly or error-prone activities, identify the main sources of critical defects. In addition, a thorough understanding of the improvement goals helps the measurement analysts prioritize measurement goals.

In general, measurement goals may be derived from improvement goals in order to:

- provide relevant information to better manage projects, e.g., planning, monitoring and control the cost, quality, and cycle time.
- provide relevant information to determine potential areas of improvement with high payoff and main sources of problems, e.g., deficient method, lack of tool support, lack of training
- assess new techniques, methods, and standards quantitatively during pilot projects and field studies, and measure the impact of change in the organization

The impact of improvement goals on the five dimensions of measurement goals can be described as follows:

- *Object of Study*: The focus of improvement goals can be on various software artifacts, e.g., system development vs. maintained systems, system documentation vs. code or design documents, depending on what processes or products need to be better understood, evaluated, or improved.
- *Purpose*: The purpose of the measurement goal is derived from the improvement goal(s), e.g., improve management of projects will lead to *monitoring* and *control* purposes. However, the feasibility of their implementation also depends on the maturity of the organization under study: Is the organization starting a measurement program from scratch? Is the level of understanding of the problems low? If the answer to these questions is yes, then *Characterization* is a very likely *purpose* for most of the goals of the measurement program. Otherwise, an organization can aim directly at assessing the introduction of new technologies through field studies,

building prediction models for management, etc. without compulsorily resorting to characterization.

- *Quality Focus*: What attributes of the object of study are of interest depends on the general focus of improvement goals since the emphasis may be, for example, on cost as opposed to quality attributes such as maintainability, reliability, etc.
- *Viewpoint*: Viewpoint(s) will be selected according to the most urgent needs of the organization in terms of measurement, e.g., is project management or technology more of a problem? The organization might face very obvious technological problems (e.g., configuration management) and therefore, the viewpoint will be technical. In other cases, project management appears to be the main problem (e.g., high turnover of personnel, systematic and large delays, budget overruns) without a clear technological cause for it. One has to decide which level of management is the most likely to profit the most from a given measurement goal. In general, based on a clear identification of relevant issues, the people performing development activities affected by measurement will be chosen as *viewpoints.*
- *Context*. The measurement program should focus on the parts of the organization which are the most in need of improvement and, additionally, are key to the success of the organization. However, depending on the resources dedicated to process improvement, the scope of measurement may vary.

Table 2 provides a structured overview of the impact of improvement goals on measurement goals.

4.2.2. Role of Descriptive Process Models

Knowledge concerning the development processes is needed in order to derive relevant measurement issues. Process descriptions include phases of development, the activities that are taking place during phases, the roles and positions involved in activities, and the development artifacts produced. Assessments based on some descriptive model of the process can help identify problems precisely and therefore help run a well focused measurement program. Such assessments can be performed through structured interviews, questionnaires, and defect causal analysis [BBK+94]. Indeed, they might point out issues to be investigated further through measurement. For example, do specification errors have costly consequences? Are most faults detected early? Is rework a substantial percentage of the development effort?

The role of descriptive process models in the definition of the GQM goal dimensions can be described as follows (see also Table 2 for a summary):

- *Object of Study:* The qualitative analysis of the descriptive process model helps identifying relevant objects of study, i.e., those in need of better understanding, evaluation, or improvement. Moreover, a descriptive process model characterizes the various artifacts that a process consumes and produces. The various states of these artifacts and their transition from one state to the other are also described. In addition,

it is determined which activities produce which artifact and whether or not the completion of an artifact is an exit/entry criterion for a phase of the process. Thus, the object of study can be specified carefully and unambiguously by using the information contained in the process model, e.g., to define the object *review*, the descriptive process model can be used to identify the various activites involved in reviewing a document.

- *Purpose:* The purpose must be realistic when considering the stability of the process. If the descriptive process model shows a lot of variability within/across projects or is only superficially defined, then *characterization* is likely to be the only achievable goal. One reason is that products, phases, or activities may be inconsistent across projects, thus making difficult
    - the identification of clear milestones for monitoring,
    - the definition of common evaluation models for products or processes, and
    - the identification of (causal) relationships between process and product attributes

  However, even though this goal can only be achieved at a rough level of granularity, measurement can provide extremely useful insight in this context. If the process is stable, then purposes other than *characterization* are more easily achievable. Control or change purposes require high process conformance of the relevant activities.
- *Quality Focus:* Based on a descriptive model of the process, structured interviews may be conducted with project participants and a process assessment can be performed, thereby identifying the most urgent problems. This should help identify the quality focus(es) of interest, e.g., if cost of the product is a problem, a quality focus of interest would be the effort spent across phases and activities or the quality of the product in terms of changes performed.
- *Viewpoint:* Based on a precise definition of roles within the context of a clearly defined process model, precise viewpoints may be identified. The tasks of the viewpoint to be supported by measurement can then be clearly derived, e.g., project planning, assessing product quality based on testing results. The different viewpoints in the organization and their most important needs should then be considered by the measurement program in order to get optimal motivation and support from all organization members.
- *Context:* The descriptive process model provides a relevant insight about what a reasonable scope for the measurement program should be. Based on resources available for measurement, one may decide to reduce this scope and therefore to work in a more limited *context*, e.g., a particular type of projects, application domain, or phases/activities.

Table 2 also provides an overview of the role of process models when identifying measurement goals.

Table 1: Overview of factors influencing the dimensions of GQM goals

	Factors	
Goal Dimensions	***Improvement goals***	***Development process***
Object of Study	Object of study should focus on products or processes that need to be better understood.	• Use process model to identify relevant objects of study • Use process model to define objects of study
Purpose	Purpose must be adapted to the level of understanding of problems in the organization.	Purpose must be adapted to: • Stability of the process • Control over process conformance
Quality Focus	Quality Focus should be consistent with the priorities of the corporate improvement program (market forces, company image, ...).	Quality Focus should address the most urgent weaknesses related to the process
Viewpoint	Depending on the improvement goals, one determines the activities the most in need of measurement. These activities are identified by specifying which are the most serious management and technical problems. The personnel who performs the activities is the selected viewpoint.	• From the roles involved in the development process, identify the viewpoints to consider • The descriptive process model contains a definition of the tasks associated with these viewpoints
Context	• Choose projects that are the most in need of improvement • Choose projects that are key to the success of the organization • Consider the resources dedicated to process improvement to determine the context	Determine the scope of measurement program by selecting a set of projects with: • similar process • similar application domain or focus on phases and activities in need for improvement

4.3. Practical Constraints

This section illustrates constraints on starting a measurement program and establishing high-priority measurement goals.

4.3.1. Types of Goals

There are various environmental constraints which determine the types of goals which can realistically be achieved with measurement:

Resources

The scope of the measurement goal has to be adjusted to the resources dedicated to process improvement and measurement. One way to do so is to limit the *viewpoints* considered and the *context* of application of measurement. Thus, fewer process and product attributes are likely to be taken into account resulting in a lower measurement cost.

Organization Maturity

The maturity of organizations has an impact on the definition of measurement goals. (Maturity is meant in the SEI Capability Maturity Model sense [PCC+93].) In cases where the practices and processes in place are unstable, characterization goals will provide less accurate results because variability will introduce uncertainty in characterization results. For example, developers will misclassify fault introduction phases because fault introduction phases mean different things to different people. This may be due, in part, to unstable and fuzzy definitions of entry/exit criteria of life-cycle phases. In other words, low organizational maturity will most likely mean lower data collection reliability. In order to increase measurement payoff, development processes should be at least stabilized, if not improved. However, it is important to note that data from unstable processes may be sufficiently reliable to partially or fully satisfy improvement goals.

State of Measurement and Process Modeling

Not any measurement goal can be achieved by any organization at any stage in their measurement program and process modeling activities. For example, it is often necessary to start with *characterization* or *monitoring* goals before *evaluation, prediction, control,* or *change* goals. An organization that does not understand how its resources are spent, what its most urgent problems are, and what the main causes of those problems are, should not evaluate new technologies. This is very difficult if there is no basis of comparison. Such a basis is provided by measurement through *characterization* goals. However, when an organization and a process are well understood (i.e., well specified, documented, and relatively stable) so that precise improvement goals can be defined, then measurement goals can aim directly at assessing new technologies and building project management models for prediction.

In order to construct a useful prediction model for process management, the organization's processes have to be understood from both a qualitative and quantitative point of view. If this is not the case, it is difficult to determine at what stage of the development process prediction is needed, what information is ready to be collected at that point, and what the scope of the prediction should be, i.e., the activities, phases, and artifacts that the prediction model's dependent variable takes into account. For example, based on the design artifacts, the project manager may want to predict coding and testing effort, respectively. In addition, s/he may want to restrict the scope of prediction to technical effort and leave out activities such as administration or project support. If prediction goals are not achievable then control or change goals are out of reach since no relationships can be clearly identified.

4.3.2. Number of Goals

In general, it is a good strategy to start with a small number of goals, gain experience, and then develop the measurement program further. The larger the number of measurement goals, the

higher is the cost of measurement. This is especially true for the first goals of a program. Additional goals often require small amounts of easily collectable additional information. When deciding about the scope/size of a measurement program, a cost/benefit analysis must be performed. However, one has to keep in mind that when introducing measurement for the first time in an organization, it is always better to minimize the risks and remain on the safe side. In this case, it is important to demonstrate that measurement is useful to everybody in the organization, from both technical and managerial viewpoints. In other words, the measurement goals should address some of the issues raised by all categories of personnel at the project or organizational level. Everybody (high-level managers, project leaders, technical leaders, and developers) should feel they have something to gain in supporting such a measurement program.

Summary Table: Types of Constraints for Goal Setting
• Resources dedicated to process improvement
• Depth of understanding of the current processes
• Stability of the processes
• Viewpoints (managers, developers,...) involved in the measurement program

5. Construction of a GQM Plan

GQM measurement plans contain the information that is needed to plan measurement and to perform data analysis. This section explains the construction of GQM plans and gives an overview of their structure. Figure 3 summarizes the relationships among the introduced concepts and may be used to facilitate the reading of the section. What is presented here refers to Step 2 of the measurement process: Identify goals and develop measurement plan.

5.1. Components of GQM Plans

A GQM plan consists of a goal and a set of questions, models, and measures. The plan defines precisely why the measures are defined and how they are going to be used. The questions identify the information required to achieve the goal and the measures define operationally the data to be collected to answer the questions. A model uses the data collected as input to generate the answer to the question. The various concepts are discussed in the following subsections.

5.1.1. Questions

The GQM questions address informational needs in natural language and are mainly aimed at making the GQM plan more readable. An example of a question would be “What is the quality of requirements documents?“ or “How many failures are found by executing the test cases?“ The answer to a GQM question can be computed based on the measures that are derived from it by the means of models described in Section 5.1.3.

Usually, GQM plans are composed of a large number of questions. Basili and Rombach [BR88] proposed categories of questions which can be used as guidelines by the measurement

analysts. A modified version of the definitions of these categories (referred to as subgoals in [BR88]) are:

- *Quality Focus:* This category contains questions concerning quality attributes of interest and defines further the *quality focus* stated in the goal. Such quality attributes are defined in collaboration with representatives of the *viewpoints*, e.g., developers, project leaders. Thus, quality attributes are defined upon well-accepted assumptions and a thorough knowledge of the environment which is based on the *viewpoint's* experience. For example, it may be assumed that all developed systems belong to the same application domain and show a neglectable amount of reuse. Such assumptions may simplify the way to measure a quality attribute such as Error Density since simple product size measures can be used (e.g., delivered lines of code).
- *Process/Product Definition:* This category contains questions concerning factors that may have an impact on the values of the quality attributes. This category is referred to as process or product definition, depending on whether the *object of the study* is a process or a product. The category *process* or *product definition* is further divided into categories.

  Concerning the *process definition* the subcategories are:

  - *Process Conformance:* This set of questions attempt to capture information concerning the adherence of the actual process to the official organizational process or any descriptive process model in use. A poor process conformance would be a threat to reliable interpretation of the data. However, it is important to know to which extent the data collected are reliable.
  - *Process Domain Understanding:* This set includes questions concerning the attributes of the objects used by the process under study and the actors performing the process. The process modeling schema in [AK94] may provide guidance on the selection of questions in this category. Examples are developers' experience, quality and structure of design documents, etc.

  The category *product definition* includes questions considering the following aspects:

  - *Internal attributes* [Fen91], e.g., logical and physical attributes of the product such as size and complexity,
  - *External attributes* [Fen91], e.g., the development cost related to the product(s),
  - *Changes* made to the product(s)
  - *Operational Context* of the product(s), e.g., who is going to use the product and in which context.

The questions contained in the process or product definition category should not address every issue related to the process or product, but only those issues which have an impact on the *quality focus*. For example, if the *quality focus* describes the effort of the testing process (*object*

of the study), questions in the *process domain understanding* category may address the number of the requirements since this may have a causal relationship with the *quality focus* under study, i.e., the effort of the testing process. The number of requirements is likely to have an impact on the effort for defining functional test cases since the more requirements, the more time is needed to define the test cases. The influential factors in the *process/product definition* category that should be considered for each *quality focus* are identified through interactions with developers and project leaders. This is performed through a well-defined knowledge acquisition procedure described in Section 5.2.1 (Abstraction sheets). The hypothesized causal relationships between *quality focus* and the identified factors are motivations for the questions.

5.1.2. Measures

Measures are an operational definition of attributes [JSK91] such as the quality focus and the factors that may affect it. Goals and questions may be defined without providing a specific operational model for attributes such as productivity or complexity. However, the next step is to provide operational definitions for those attributes so that they can be measured. Some attributes are actually based on several more elementary attributes, e.g., productivity which is based on product size and effort. Therefore such attributes need to be operationalized through models that have as parameters more basic measures, e.g., Defect_Density_Model = Number of defects/LOC.

Defining a measure in an operational way includes defining its measurment scale and its range. The level of measurement (i.e., nominal, ordinal, interval, rational) of the scale will help select adequate data analysis procedures. The issue of selecting data analysis procedures is, however, a subtle issue which is discussed further in [BEM96]. The range gives information on what data values are expected and may help identify abnormal values. For interval and ratio scale data, the measurement analyst has to specify the unit of measurement. For nominal scales and ordinal scales of limited range, the measurement analyst has to state the semantics of all possible categories. For example, assuming there is a measure capturing the tester's experience, the scale could be ordinal and the range could be composed of the High, Medium and Low experience levels. As an example, the High, Medium, Low scores may be defined, respectively, as having developed functional test cases for more than five systems, at least one system, and never. Intervals or scores should be defined so that measurement results show variability across the scale. When data are collected through surveys and/or interviews, then their reliability [CZ79] should be studied carefully by assessing the measurement instruments, e.g., questionnaires, before the start of the measurement program,.

5.1.3. Models

During the definition of GQM plans, different categories of quantitative models have to be built for the following reasons:

- GQM plans have to be operationalized. Therefore, the various abstract attributes of the artifacts being studied, have to be defined in an operational way which is suitable to the

goals and makes plausible assumptions about the environment. Examples of abstract attributes are maintainability or reusability of software components. We refer to models describing attributes in an operational way as *descriptive* models.

- The way quality or productivity comparisons and evaluations will be performed has to be defined precisely, i.e., how does an object or a set of objects compare to another object or population of objects with respect to a given attribute (i.e., the *quality focus*) or rather, to be precise, one or several of its measures. We refer to such models as *evaluation* models. For example, is a component overly complex or difficult to maintain based on its internal characteristics? How does it compare to the whole population of components in similar application domains?
- The way predictions will be performed has to be defined precisely. Therefore, several questions must be considered:
    - What will the functional form and structure of the models be? Should the models be linear/non-linear, univariate/multivariate, or take into account variables' interactions?
    - What model building technique will be used? Is multiple regression adequate?
    - What explanatory variables will be used to predict the dependent variable? Will system size, team experience, and application domain be sufficient to predict system cost?

  We refer to models describing these aspects as *predictive* models.

It is important to define descriptive, evaluation, and predictive models during the definition of the GQM plan since they will drive, to some extent, the definition of the measures to be collected and the definition of data collection procedures. For example, models may impose requirements on the type of measurement scale needed (e.g., it is preferable to measure complexity on an interval scale since a regression-based predictive model will be used) or on the reliability of the data collection (e.g., high measurement reliability is required since the measure is expected to be one of the main predictors in many predictive models).

Summary Table: Model categories
• Descriptive models operationalize attributes.
• Evaluation models are decision functions based on attributes.
• Predictive models predict external attributes of the object of study.

5.2. The Construction of GQM Plans

GQM plans tend to become large and complex because they include a great deal of information and interdependent concepts. Two techniques provide support for constructing adequate GQM plans: knowledge acquisition based on abstraction sheets and descriptive process modeling. This section focuses on the content and structure of these documents and their role in the

construction of GQM plans. The different categories of quantitative models are also briefly defined.

5.2.1. Abstraction Sheets

GQM plans are constructed by defining and combining questions, models, and measures based on the *viewpoints*' experience. The *viewpoint* does not need to see all the details of the GQM plan. The GQM plan is constructed by the measurement analyst based on the *viewpoint's* experience. To support the structured interaction of the measurement analyst with the *viewpoint*, a simplified view of GQM plans has been designed [Hoi94, DHL96]. The documents are called GQM *abstraction sheets* and are used specifically for the purpose of facilitating interactions with *viewpoints*.

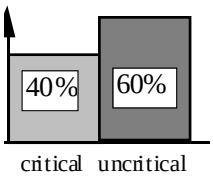
In order to capture the experience of the *viewpoints*, the GQM abstraction sheets are used as knowledge acquisition instrument during interviews. Their components, referred to as quadrants, cover the issues that *viewpoints* need to address during interviews. Abstraction sheets may be viewed as structured guidelines in order to involve the *viewpoints* into the definition of the measurement plan. During the interviews, it can be shown on a transparency to provide all the participants with an overview and stimulate group interaction. The GQM abstraction sheet completed in interviews is a major input when constructing the GQM plan since this is the main way of integrating the *viewpoints*' goals, experience, and feedback.

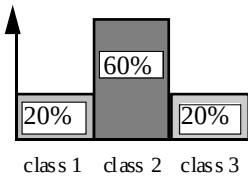
The suggested template for the components of a GQM abstraction sheet is shown in Figure 1. In the following, the content of each quadrant is described:

- *Quality focus:* This quadrant captures information that defines the *quality focus*. The information is intended to capture the *viewpoints*' intuition about the *quality focus* and transform it into an operational definition (or several if there are several alternatives to consider). The information captured here is used to construct all the required models in the *quality focus category* of the GQM plan.
- *Baseline hypothesis:* This quadrant specifies what is expected by the *viewpoints* with respect to the measures and models that define the *quality focus*, e.g., effectiveness. For example, if the *quality focus category* measures the distribution of defects across classes of faults, the *baseline hypothesis* would specify the expected distribution of faults, e.g., see Figure 1. The values of this expected baseline can be based on the intuition of the *viewpoints*. A predefined baseline will help demonstrate the usefulness of measurement by pointing out discrepancies between expectations and reality once the data are collected. In order to avoid misunderstandings, it can be helpful to represent the baseline hypotheses graphically. The same kind of graphical representations can be used to present the data to project personnel and management.
- *Variation factors:* This quadrant captures the factors that are believed by the *viewpoint* to have an impact on the *quality focus* in a particular context. These factors will trigger the need for questions, will result in the operational definition of models and

measures.They will also trigger the statement of questions in the process or product definition categories.

Figure 1: Example of a GQM abstraction sheet

<table>
<tr><td>Object of Study
unit test</td><td>Purpose
prediction</td><td>Quality Focus
effectiveness</td><td>Viewpoint
tester</td><td>Context
organization A</td></tr>
<tr><td colspan="3">Quality Focus
1. Number of detected failures
2. Proportion of critical/uncritical failures
3. Number of detected faults
4. Distribution of faults across fault classes
.
.
.</td><td colspan="3">Variation Factors
1. Quality of test cases
2. Test method used
3. Test method conformance
4. Experience of testers with tools
5. Understandability of the requirements
6. Understandability of the source code
.
.
.</td></tr>
<tr><td colspan="3">Baseline Hypotheses
1. 30
2.
 

3 40
4.
 
</td><td colspan="3">Variation Hypotheses
1. The higher the quality of the test cases, the more failures detected
2. Different testing methods detect different numbers of failures
3. The better the method conformance, the more failures detected
4. The higher the experience with the testing tool, the more failures detected
5. The better the understandability of the requirements, the more failures detected
6. The better the understandability of the source code, the more faults detected</td></tr>
</table>

- *Impact on baseline hypothesis*: The expected impact of the *variation factors* on the *quality focus* are captured here. Every *variation factor* must relate to the *quality focus*. The relationship between *variation factors* and *quality focus* must be falsifiable, i.e., testable. For example, the size of artifacts used as inputs by an activity could be considered a *variation factor* of the activity's effort. In this case, the impact on the *baseline hypothesis* could be stated as: "The larger the input artifact, the more costly the activity." This expected impact of the *variation factor* is the

motivation for including the factor in the process or product definition category of the GQM plan. If *viewpoints* cannot provide any experience-based hypothesis for the impact of a *variation factor*, this factor should not be included in the GQM plan. This helps avoid situations where GQM plans include too many irrelevant factors and get too complex.

Figure 1 shows an (incomplete) example abstraction sheet and Figure 2 an excerpt from the derived GQM plan. Specific models (descriptive, evaluation, predictive) and measures cannot be shown here. Examples for models and measures will be provided below.

A descriptive model for characterizing test methods (used to answer Question D.2 in Figure 2) would be, for example, the taxonomy shown below. However, this is a very broad and academic classification. Each organization would have to develop its own classification depending on its testing practices. An example of test method taxonomy could be:

- Structural testing
    - All paths
    - All conditions
    - etc.
- Functional testing
    - All requirements
    - All equivalence classes and boundaries
    - etc.
- Statistical testing
    - Operational profile of user type I
    - Operational profile of user type II
    - etc.

Figure 2: Excerpt of a GQM plan

Analyze unit testing
For the Purpose of prediction
with Respect to effectiveness
From the Viewpoint of the tester
in the Context of Organization A

Process Definition
Process Conformance
Question D.1: How much effort was needed to define the test cases?
Question D.2: What test methods were used?
Question D.3: How closely did the testers follow the test method?
Process *Domain Understanding*
Question D.4: How experienced were the testers wrt. the testing tool?
Question D.5: How well did the testers understand the requirements?
Question D.6: How well did the testers understand the source code?
Quality Focus
Question Q.1: How many failures were detected?
Question Q.2: What was the criticality distribution of failures?
Question Q.3: How many faults were detected?
Question Q.4: What was the distribution of faults across fault classes?

A descriptive model for a quantitative attribute would require the definition of a unit of measurement and precise semantics. For example, testing effort will be computed in person-days and will include the following activities: defining test cases, running test cases, checking test outputs, and writing test reports. Such activities would be precisely defined by a descriptive process model.

Once abstraction sheets have been completed, a first assessment of the dimension of the measurement program can be performed. Measurement analysts and users may decide to restrict the scope of the program, i.e., viewpoints and context, in order to decrease the number of variation factors to be considered. In addition, factors judged as secondary may be left out.

In addition to being used as an instrument to support interviews during the definition of GQM plans, abstraction sheets may be used to show a simplified view of the GQM plan to project personnel. This will facilitate any discussion about the GQM plan.

5.2.2. Using the Descriptive Process Model

In the context of a measurement program, descriptive process models are needed for the following reasons:

- The definition of a measurement program and its data collection procedures requires knowledge of the process under study.
- Designing unintrusive measurement programs that fit into the actual process [BDT96, BBC+96] is a fundamental requirement for success.
- The data collected will not be interpretable and amenable to process improvement if analyzed in a vacuum, without a good qualitative understanding of the process (see Section 8) [BW84, BBC+96].
- Discussions, decisions about changes, and communication of improvement decisions in an organization will require some widely-accepted model of the process under study.

This section addresses the application of descriptive process models to the establishement of measurement programs. More precisely, the information items relevant to defining a GQM plan can be classified into at least three categories:

- Definitions of phases and activities, and the data/control flows that relate them.
- Characterization of produced artifacts and their various states (i.e., under the form of a state-transition diagram) during the development process.
- Positions and associated roles in the organization, i.e., responsibilities with respect to activities and produced artifacts.

Having this kind of information at hand during the GQM interviews helps the measurement analysts ask relevant questions, identify important factors and concepts related to the *quality focus*, and define adequate measures. For example, if a *viewpoint* wishes to characterize the

effort distribution of a process, the process model can be used to determine precisely how effort is broken down into phases or activities.

Furthermore, together with the abstraction sheets, the process model is an input for the construction of a GQM plan. For example, if the interviewed *viewpoint* wishes to know the number of faults found in each of the verification activities, the process model can help identify the different verification activities and the documents that contain the relevant information about faults.

5.2.3. Definition and Use of Models

As mentioned above and based on our experience, three main categories of models are defined and constructed in a GQM plan*: descriptive models, evaluation models, prescriptive models*.

Descriptive models can be formalized as follows:

$$\mu = f(X_1, ..., X_n)$$

Where μ is a measure based on a model integrating n other measures (X_1, ..., X_n). A simple example is μ = *defect density* in a software component, which can be defined as the ratio of number of defects over size. Here X_i's would be number of defects and, for example, number of Function Points. In this case the way defects are counted would have to be defined in an operational and unambigous way. Descriptive models are commonly used in all measurement programs, but rarely explicitly defined and discussed in terms of their underlying assumptions. Building descriptive models is a matter of capturing expert's and practitioners' intuition into a quantitative model, e.g., define in quantitative terms what defect density is. Such models are usually of limited scope and are based on assumptions that are specific to the environment where they are defined. For example, a simple defect density model for code documents such as Number of defects/LOCs might make sense in an environment where all systems belong to a well defined application domain. However, when comparing systems of a different nature, developed under very different constraints (hardware, team size, etc.), this model probably does not make any sense at all. Furthermore, this measure may be meaningful at the team level but is likely to be overly simplistic at the individual level. The count of lines of code (LOC) is dependent on programming style at the individual level whereas this effect may average out at the team level.

Evaluation models capture the situations in which a particular attribute needs to be evaluated based on one or more of its measures. For example, based on different measures of complexity, the quality manager may want to determine if the structure of a component needs to be simplified or reengineered. In other words, the value range of measures needs to be mapped to alternative decisions leading to corrective or preventive actions. Therefore, evaluation models are of the form:

$$\delta = f(X_1, ..., X_n)$$

where $\delta \in \{d_1, ..., d_m\}$, the set of all possible alternative, and mutually exclusive, decisions based on the evaluation model and f is a decision function with decision criteria $\{x_1, ..., x_n\}$ as inputs. Such decision functions, e.g., when to inspect a module, can be built based on

- *expert opinion and captured decision algorithms that are based on intuition and experience*. For example, experts may decide that a cyclomatic complexity (one of the x_i's) value above 15 is not acceptable for newly developed components. In that particular case, it might be decided that only components with a cyclomatic complexity above 15 will have to be inspected thoroughly by the quality assurance team ($\delta \in$ {inspection, no inspection})
- *the analysis of data related to actual decisions*. In that case, inductive algorithms can be used to generate decision models such as decision trees [Bri93]. For example, let us assume that one collects data about the components that are selected by experts for undergoing formal inspections. One can measure the internal attributes of these components as well as those that have not been selected. Using these internal attributes as explanatory (or independent) variables, one may generate the decision trees that characterize components to be inspected. For example, if their cyclomatic complexity is above 15 and their coupling is at least at the data coupling level [CY79], then they should be inspected.

Another common application of measurement is prediction, e.g., effort and error predictions. In order to build prediction models, one needs to develop functions of either one of the following two types:

$$\hat{e} = f(x_1, ..., x_n) \qquad (1)$$

where $\hat{e}$ is the model point estimate of the variable e to be predicted (i.e., dependent variable) such as project effort. $\{x_1, ..., x_n\}$ could be the set of project characteristics (e.g., team experience, product size) driving effort.

Another type of prediction model would be:

$$p(e) = f(x_1, ..., x_n) \qquad (2)$$

where p(e) is the probability of occurrence of event e [BTH93], e.g., fault detection in a software component. $\{x_1, ..., x_n\}$ could be the set of component internal attributes (e.g., complexity, coupling) used to explain the occurrence of faults.

Many techniques may be used to build such prediction models such as:

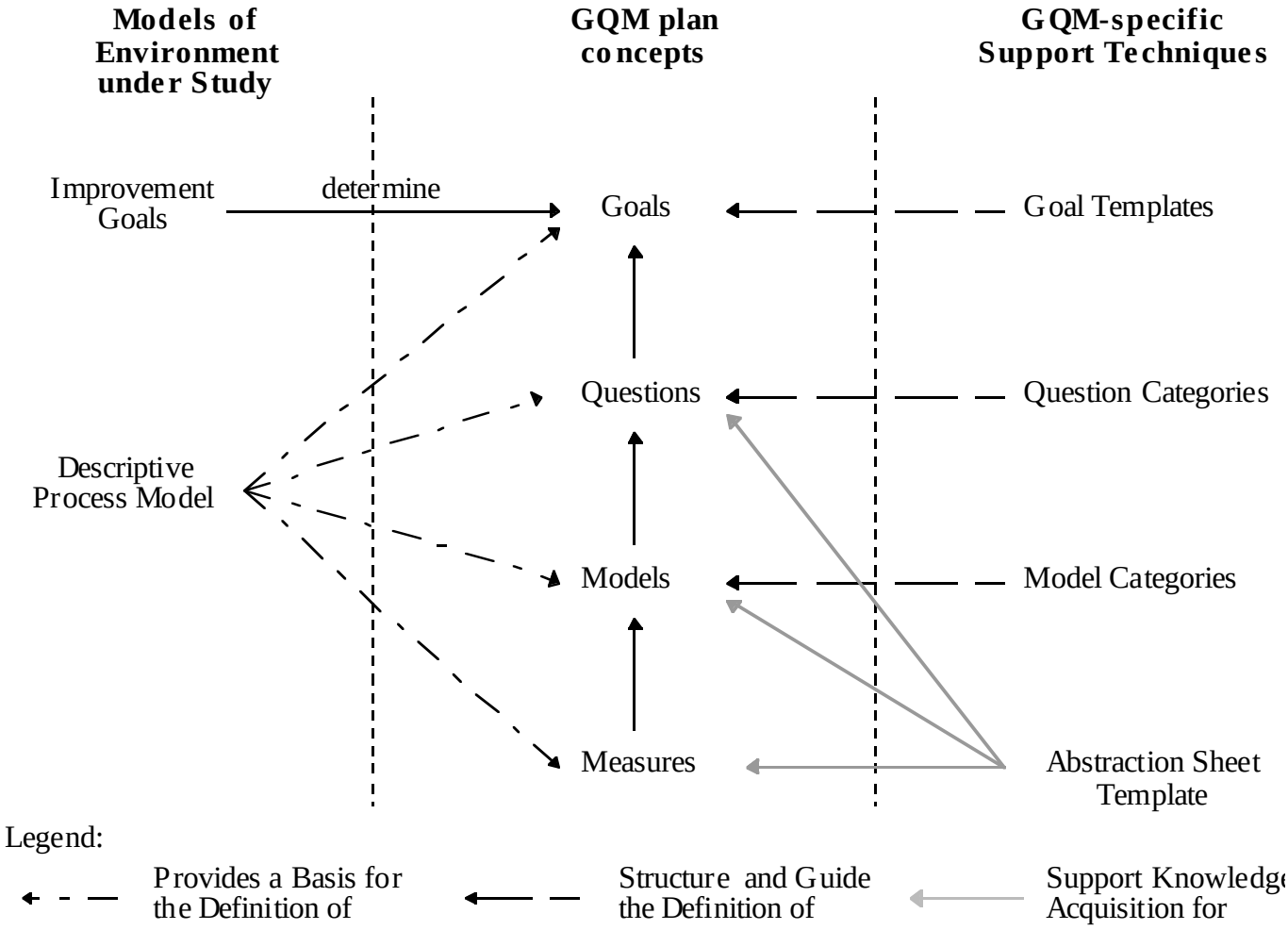
- Regression analysis [BTH93, BMB94b]
- Inductive algorithms, e.g., classification trees, Optimized Set Reduction [BBH93]
- Neural networks [KPM92]

Depending on the type of data to be used, the intended use of the model and the profile of the future users of the model, different techniques should be used [Bri93].

Different measurement purposes require the use of different categories of models:

- *Characterization* purposes use descriptive models only. These models are sufficient to provide a snapshot of the current situation.
- *Monitoring* purposes need descriptive models only, like *Characterization.* However, for monitoring, inputs for these models are collected over time and models are repeatedly computed to follow the trends/evolution of the performance/state of processes and products.
- *Evaluation* purposes need evaluation models as well as descriptive models. The descriptive models are used as an input for the evaluation models. For example, a descriptive model may be defined and used to measure the complexity of C++ classes. Then, if one needs to decide whether or not a class has an acceptable level of complexity, an evaluation model needs to be defined such that it takes class complexity in input and assign it to a decision: acceptable, not acceptable.
- *Prediction* purposes need prediction models with descriptive models as inputs. The dependent and explanatory variables need to be operationally and precisely defined through descriptive models. Then, prediction models can be derived based on historical data. For example, based on C++ class complexity, we may want to predict class inspection effort.
- *Control* purposes need descriptive, evaluation, and prediction models altogether. The evaluation models are needed for formalizing control decisions. They use descriptive models as inputs. Furthermore, the evaluation models may use the output of prediction models as an input for the evaluation of some attribute. For example, one may decide to inspect all classes above a certain level of complexity when they have been written by less experienced programmers. Such a decision criterion may be based on a subjective evaluation model or, on the other hand, may be derived from a prediction model that relates fault probability to programmer experience and class complexity. In this context, there is an underlying assumption that prediction models are the result of a cause-effect relationship.
- *Change* purposes, again, need a combination of the three kinds of models. They need descriptive models to describe attributes, evaluation models to assess the actual situation, and prediction models to predict what will happen in case of change. In this context, one needs a fine grain understanding of the cause-effect mechanisms underlying the prediction models. One needs to be sure that any change in the explanatory variables will result in the expected change of the dependent variable. For example, one may change the development process by introducing a rule stating that C++ classes that are likely to be complex must be developed by experienced programmers. Based on a prediction model such as the one mentioned above, the effect of such a change on fault probability may be assessed, assuming there is a causal effect.

Figure 3: Main Concepts and Techniques for the Construction of GQM Plans



6. Implementation of GQM Plans

Based upon GQM plans, specific data collection procedures are designed in a way so that reliable data can be collected in the environment and process under study. This section provides guidelines for their definition. In addition, practical issues, which are crucial for the successful implementation of measurement, are discussed. This section refers to Step 3 and Step 4 of the measurement process introduced in Section 3.

6.1. Defining Data Collection Procedures

After measures have been defined for each GQM goal, they have to be mapped to precise data collection procedures that provide the required level of measurement reliability. Another important criterion to consider when defining data collection procedures is intrusiveness. The cost of measurement should be minimized to the extent possible. In order to do so, several measures may have to be collected concurrently through an integrated data collection procedure. When developing the data collection procedures, decisions have to be made concerning the

point in time, the responsible person, and the best means for data collection. The descriptive process model provides an important input for these decisions [BDT96, BMS+95, BBK+94].

6.1.1. When to Collect Data

For each measure the measurement analyst has to decide which point in time is best for data collection. According to the measurement purpose and data collected, three main types of strategies can be adopted for data collection: periodically, at the beginning/end of activities and/or phases, and when an artifact has reached a certain state. These three cases are described next.

- *Periodically:* Data are usually collected periodically when one needs to build dynamic models of the development process (e.g., cumulative effort or defects over time) or when one wishes to generate progress reports, for example weekly (e.g., percentage of components that have undertaken unit test). Also, if data do not map into the development process, e.g., effort spent in non-project-activities such as training, meetings, they should be collected periodically.

  A particular case is when the data concerns the project environment, the project itself or the project staff, then they usually have to be collected only once, e.g., the design method used in the project. The best points in time for collecting such data are when the project is started or completed.
- *Beginning/end of activities/phases:* Data are usually collected at the start or completion of activities/phases when one needs to get quantitative insight into their efficiency and effectiveness, e.g., defect detection rates and cost for various testing activities. A descriptive process model is necessary to determine the adequate activities/phases where to collect each specific data item.
- *Artifact state transition:* In this strategy, data are collected when an artifact reaches a certain state, e.g., when components go into configuration management. Such an approach is usually taken when one wants to know the attributes of the artifacts, e.g., complexity of components, their level of quality, and/or their respective cost. Thus, one can, for instance, better determine the defect-prone and costly artifacts. Here again, a process model containing, for example, state-transition diagrams of products [AK94] would be helpful. They define the states of the products and their transition constraints from one state to the other.

These strategies support, respectively, three categories of application of measurement:

- Monitoring and control of software development projects, e.g., extent to which an activity /phase has been completed.
- Process improvement (within or between projects), e.g., identify inefficient or ineffective activities.
- Support quality assurance activities, e.g., identify artifacts to be inspected.

Table 3 summarizes the issues discussed above.

Table 3: Strategies for designing data collection procedures

Collection strategy	Measurement purpose	Examples	Inputs needed
Periodically	monitoring and control of projects	% of modules tested cumulative effort over time	Level of granularity of updates, e.g., weekly, monthly.
Beginning/End of activities	process improvement: Identification of inefficient and/or ineffective activities	defect detection rates and cost of testing activities	descriptive model of activities and artifacts used or produced by processes
Artifact States	quality assurance support: Identification of defect-prone or costly components	how much effort was spent on inspection, what is the observed quality?	state-transitions diagrams of products

6.1.2. Who Collects The Data

When a clear schedule for data collection has been set up, the measurement analyst has to decide who can and/or should collect the data. The first question is to determine whether a tool can automate the data collection. If the answer is no, then subjectivity in measurement cannot be avoided. Since subjectivity is inherent to software measurement, determining the right person(s) to collect the data is crucial.

When selecting the person in charge of collecting the data, one should use several criteria:

- *Expertise:* who has the technical/managerial expertise to provide the data accurately? This depends on both training/education and experience.
- *Bias:* Is there any reason for the data provider to show any (intentional or not) bias in the information he or she provides? In other words: Can the results of measurement be used to assess him/her? Is there a strong and clear commitment of the data provider's manager NOT to use data to assess individuals? Does the data provider have bias against the principles of measurement itself?
- *Access:* If the object being measured is a product, then the person who produced it, used it, or, to a lesser extent, the person who reviewed it, may very likely be in the best position to access the artifact and all necessary related information. If the object being measured is a process, then the person(s) who performed it, or, to a lesser extent, managed it, are likely to provide accurate information.
- *Cost:* Can the time spent on measurement have costly effects on the project? Is the person's schedule tight and key to the project success?
- *Availability:* Is the person available to spend time on data collection?
- *Motivation:* How committed is the person to the measurement program? Is measurement a foreign concept or a well understood principles with clear aims and benefits?

6.1.3. How to Collect Data

Designing accurate measurement instruments is crucial in order to get reliable data. There are three main categories of measurement instruments: Tools (e.g., static code analyzers like GEN++™, Datrix™), questionnaires (e.g., NASA SEL forms [Nat94]), and structured interviews. Measurement tools can be triggered automatically by development tools (e.g., configuration management tool, compilers), questionnaires can be on-line or paper forms, and there are many types of interviews that vary according to the strategy used to elicit knowledge [Eri92]. Moreover, it is important to note that optimal reliability is not always needed and that the required level of reliability depends on the measurement goals.

The decision about which instrument to use depends on the information collected. Tools can be used for objective artifact measures (e.g., LOC), questionnaires and structured interviews for process measures (e.g., effort spent on an activity) and subjective artifact measures (e.g., understandability of the requirements).

There are several issues of importance for the acceptance of questionnaires. Forms filled out by project personnel should be designed so that each person has to fill out only one specific form at a time. For a better acceptance of the data collection procedures by personnel, the forms should be adapted to the terminology, procedures, and tools (e.g., SEEs, CASE) used in the project.

Filling out the forms should be perceived as a natural part of the various activities, and should not be considered as an overhead by the management or personnel. One solution to achieve this is that data collection forms may also have other purposes than measurement. They may be used also to support some “regular” project activities, e.g., quality assurance. For instance, in order to help communication and change traceability, fault report forms can be passed from the person who discovered a failure to the person who identifies the fault, and then to the person who makes the correction. Along the way, information is conveyed, the change history is recorded, and data are collected.

Table 4 summarizes the most important decision criteria concerning data collection procedures.

Table 4: Decisions concerning data collection procedures

Type of decision	**Decision criteria**
When to collect the data	Application of measurement, i.e., monitoring, prediction, control, quality assurance.
Who collects the data	People’s expertise, bias, data access, cost, availability, motivation
How to collect the data	Tools available, Procedures and Tools used in the project

6.2. Practical Issues

This section addresses practical issues concerning measurement acceptance and reliability.

6.2.1. Getting Commitment

An important principle of the GQM approach is that the project personnel who collect the data participate actively in both the definition and the interpretation of the data. Thus they realize that the collected data is used to address their own needs and are motivated to provide reliable data. The participation of project personnel should cover the following activities:

- *Goal setting:* The measurement goals should concern developers as well as different levels of management, so that the different project viewpoints are represented by the measurement program**.** This will increase the chances for acceptance because it serves the interests of all parties concerned. For example, if a measurement program were only supporting managerial goals like project control and productivity and the developers should provide the data, then they would not have any motivation for providing reliable data.
- *Measurement planning:* Planning, i.e., the definition of questions, models, and measures, requires the participation of project personnel. Thus, project personnel and management will be involved in all important decisions about measurement. This increases chances of acceptance because it will ensure that the measurement program is well-suited. For example, quality models are intended to capture and quantify the various viewpoints on quality. Therefore, *viewpoints* should always be interviewed in order to provide inputs for the definition of such models and their validation. For example, the viewpoints on reliability may be very different. Reliability may be perceived by the project leader as being mean time to failures (a customer point of view) whereas developers may see it as the remaining fault density in the system.
- *Data collection forms and procedures:* These should be designed based on a careful analysis of the personnel's tasks and project's procedures, the organization structure, and the terminology in use. Again, the data collectors should be involved in testing and reviewing the forms. Pretesting of the forms will provide evidence of the reliability of the data collected or the lack thereof. For example, the classification of faults is an important issue which may have major consequences on the process improvements suggested. Therefore, the semantics of alternative classes should be well understood by all the data collectors.
- *Interpretation of data:* Despite the fact that measurement specialists have to analyze the data, sometimes using sophisticated statistical techniques, the interpretation of the results must be performed in close collaboration with the *viewpoints* and, eventually, the people who collected the data. Feedback sessions must be held (see Section 8) in which the results of the data analysis performed by the measurement specialists are presented to and discussed with the *viewpoints* and, eventually, the data collectors. A first feedback session often leads to a second round of data analysis since new questions about the data are usually raised.

To get commitment from high-level management, it is necessary to trace back measurement goals to strategic improvement goals for the organization. Thus, explaining why measurement is going to help achieve improvement goals is crucial and must be made explicit.

6.2.2. Training

The project personnel involved in measurement must be trained in several topics in order to ensure wide acceptance for measurement and to get reliable data.

The main topics that have to be addressed by training are the following:

- *The purposes of the measurement activities taking place in the project and their goals.* This is important in order to reassure the project personnel so that they do not think that measurement is used to control and assess them. It must be stressed that the development process is the object to be measured and not the developers as individuals. In addition, it is important to convey the idea that a better controlled process will benefit not only project leaders but developers too since they will be more likely to work within more reasonable deadlines and with more adequate resources. Furthermore, data may also be used to their advantage to support claims for an improved development environment or to address any other type of problem.
- *The principal ideas behind the GQM approach.* Project personnel should be provided with motivations for interviews, data collection procedures, and feedback sessions.
- *The important issues concerning reliability of the collected data*. For example, the data providers have to know why it is important to fill out a data collection form at a special point in time and why they should pay attention to dependencies among the data they collect, such as the breakdown of change effort across system modules.
- *The data collection tools in order to achieve a more efficient and reliable tool usage.* If the data providers are not well-trained to use the tool, they will get discouraged to collect data.

Summary Table: Topics Covered by Training
• Purpose(s) of measurement
• Role(s) of project personnel during data collection
• Reliability issues
• Tool training

6.2.3. Tools

Various tools are needed for data collection, for analysis, and for visualization. When developing a *data collection tool*, one should be aware that during the initial stages of a measurement program, the measurement plan often changes. If it does change, tools have to be adapted too, and this usually takes considerable effort.

Static analyzers are frequently used to collect product measurement data, e.g., control flow complexity. Unfortunately, many of these tools have a fixed set of collectable measures or have limited degrees of freedom when defining new measures for the tool to collect. This is inherently against the philosophy of the GQM paradigm which requires measures to be derived from goals and context information. This problem is not easy to overcome and often results in the development of additional analyzers or the use of unsatisfactory measures. However, a few tools have been developed to allow the definition of new measures such as GEN++™ for C++ code. Another danger when using commercial static analyzers is the tendency to perform massive shot-gun data analysis to find out trends in data. Such an approach is likely to yield meaningless and often uninterpretable results. Again, this goes against the fundamental principles of the GQM paradigm.

The use of data collection tools for process measurement instead of paper-based forms is particularly useful if many data items are being collected, e.g., when the project involves many people and/or measurement goals. If the project is of modest size, paper-based data collection forms can be used. In order to perform data analysis efficiently, data has to be stored in a database. This should not be performed by the project personnel, but by a support group in charge of data quality assurance and storage.

In order to *analyze the collected data* and build quantitative models, we advise acquiring one of the standard statistical packages available on the market (e.g., SAS™, Statistica™). Even though often used in practice, standard spreadsheet software is not sufficient in the long run and only offers basic analysis capabilities.

The effective *presentation of analysis results* in the feedback sessions is crucial to the measurement program [GHW95]. When relevant, the presentation of the data should be adapted to the usual presentation conventions of the organization. Furthermore, using adequate visualization techniques may dramatically improve the interaction during feedback sessions.

Summary Table: Tool Support
• Data collection
• Data analysis and quantitative modeling
• Data visualization

6.3. Data Quality Assurance Procedures

When the data has been collected, it has to go through a quality assurance process before it can be stored or analyzed. The quality assurance process addresses the following issues:

- There may be data collection forms with missing data. The measurement analysts must determine why the data is missing, e.g., were the questions not applicable, not understood, considered irrelevant.

- Data collection forms may contain outliers or values that are out of range. In this case, the measurement analysts have to make sure that the data make sense, e.g., is it sensible, in a particular context, to have an effort of five days for the review of a module, or did the person possibly mean five hours?
- Various dependencies between data collection forms and the developed artifacts have to be checked, such as:
    - If there is a failure report form after test, there has to be a fault report form after the correction has been completed.
    - The development of a module must be tracked through all development milestones such as requirements, design, coding, and testing.

  These kinds of dependencies can be easily checked through database queries when a database management system is used.

If this quality assurance process leads to the detection of missing or faulty data, these data should be discussed with the data collectors. When possible, they have to correct the data themselves so that they can improve their data collection skills. If particular data items regularly appear to have a low reliability, the data collection procedure and/or training should be reconsidered, assessed, and eventually improved.

Summary Table: Data Quality Assurance Procedures
• Missing data in data collection form
• Out of range values and outliers
• Cross-reference checks based on redundant data
• Missing forms when comparing data collected to generated development artifacts

7. Data Analysis

This section discusses the types of data analyses that may be relevant in the context of software measurement: comparison of actual data with baseline hypotheses, and validating and quantifying hypothesized relationships (causal or not) between the *quality focus* and *variation factors*. The activities described in this section refer to Step 4 of the measurement process.

7.1. Quality Focus: Comparison of Data with Baseline Hypotheses

The data collected can be used to build quantitative baselines for the development projects of the organization. It is usually interesting to compare actual baselines to the expected ones (i.e., baseline hypotheses as defined in abstraction sheets - see Section 5.2.1). This will allow the measurement analysts to:

- Explain these differences and determine whether they are symptomatic of a problem.
- Trigger discussions with developers, project leaders, and management.

- Show the usefulness of measurement by identifying departures from expectations or common knowledge.

It should be noted that the quantitative baselines and their comparison to the baseline hypotheses are computed based on the various models defined in the GQM plan (i.e., descriptive and evaluation models). For example, if one question asks about the distribution of effort over phases, the collected data are aggregated according to the descriptive model for effort distribution. This allows for the computation of the actual distribution of effort over phases. Then this quantitative baseline may be compared to the expected one (i.e., baseline hypothesis) through statistical inference testing by comparing distributions and assessing the significance of their differences.

Significant differences between the baseline hypotheses and the actual data lead to discussion points that should be addressed in feedback sessions (see Section 8). Moreover, they are likely to trigger further investigation of the data in search for factors that explain the differences. For example, if the testing phase detects more defects than expected, the analyst would look at the quality of the documents (e.g., the documents may be of poor quality and should be better reviewed prior to testing) and look at the testing technique (e.g., it may be more effective than usual). In general, the distribution of the variation factors should be examined and compared to their expected distributions. If deviations are observed, it could help explain the deviation from the hypothesized quality focus baseline, e.g., modules have a distribution skewed towards high scores on the complexity scale whereas the opposite was expected. It should be noted that such an analysis of the baselines is a required component of the preparation of feedback sessions. Feedback sessions will help select the most probable explanations among plausible alternatives.

7.2. Variation Factors: Validation of the Variation Hypotheses

Depending on the *purpose* of the GQM goal, the following strategies are applied:

- For prediction purposes, the *variation hypotheses* are tested by answering the following question: Did the variation factors have the expected impact on the *quality focus*? If the expected impact cannot be verified, then excluding the variation factor from the data collection should be considered. However, such a decision should be made carefully because such a result may also be due to:
  - the use of an inadequate modeling technique, e.g., a linear regression with an underlying exponential relationship.
  - the sampling of a too small dataset leading to the lack of statistical power [BEM95b].

  If the expected impact is observed, the identified relationships may be used to build new or more reliable models for project management, quality assurance, etc.
- For control and change purposes, assuming the variation factors have already shown to be of some impact, the analysis concentrates on determining whether or not this impact is due to a causal relationship between the quality focus and the variation

factors. Regarding causal analysis, techniques such as path analysis may be used [Ash83].

It should be noted that *variation factors* are not relevant in the case of characterization since this purpose focuses exclusively on providing a snapshot of the development process and product, e.g., distributions of effort across phases or components across complexity levels.

8. Presentation and Discussion of Analysis Results

This section discusses recommended strategies for the dissemination, discussion, and interpretation of analysis results. These activities are part of Step 4 of the goal-oriented measurement process.

8.1. Objectives of Feedback Sessions

The major objective of the feedback sessions is to interpret the data analysis results with the help of the *viewpoints* and the project personnel who have the necessary expertise. Therefore, the results are presented to the session participants and possible interpretations are discussed. The presentation and discussion is structured according to the stated GQM goals. Depending on the *purpose* of the goals, the chosen models are assessed (i.e., do they make valid assumptions and do they fully capture the phenomenon). Improvement possibilities concerning the development process or changes of the project plan may be considered by the participants.

An important goal of the feedback sessions is the evaluation of the measurement program. If participants are not able to use the data, this may be explained in different ways:

- The results are not presented in an adequate or comprehensible form to the participants.
- The data may not fit the stated measurement goal, i.e., the defined measures do not adequately capture the attribute that one purports to measure.
- There may be some relevant information missing, i.e., some extraneous factors are not measured.

During the initial phases of a measurement program, these issues have to be considered carefully, because they ensure the completeness, consistency, and reliability of a measurement program. After a few projects, the measurement program should stabilize.

Subsequently to the feedback sessions, one should refine the analysis, and, if necessary, the GQM plan and the collection procedures, based on new insights gained. If alternative interpretations still exist after the feedback sessions, further analyses of the data may help select the most likely one. New problems may be identified during feedback sessions and may require further analysis to be addressed properly.

Summary Table: Objectives of Feedback Sessions
• Interpret trends identified by the data analysis • Take corrective actions concerning the project, process, or measurement program • Assess/refine the measurement plan

8.2. Organization of Feedback Sessions

Once the data collection has started, feedback sessions should be held periodically, e.g., should be a matter of weeks. They are prepared by analyzing the data. Preparation consists of the following activities:

- Statistical and/or inductive analysis of the data.
- Layout of results in comprehensible and intuitive ways.
- Identification of alternative interpretations.

The participants of the feedback sessions are the *viewpoints* of the GQM goals and the people who collected data. Both groups are important for the interpretation of the data and are likely to be affected by process and data collection changes that may be decided during the feedback sessions [GHW95].

The presentation material should be structured according to the GQM plans and contain at least all the discussion points identified in the analysis. The material should be distributed to the participants at least one week before the feedback session so that they have a chance to look at the results before the discussions.

Summary Table: Organization of Feedback Sessions
• They are held periodically • Participants are data collectors, viewpoints, and measurement analysts • Presentation material should be distributed well in advance

8.3. Interpretation of Results

The analyzed data are interpreted by the *viewpoints* and, in some cases, the data collectors. Measurement analysts check whether interpretations are fully consistent with the data analysis results. *Viewpoints* will know how to use the data for their purposes, and the data collectors know how well the data they provided were actually collected and whether they are suited for the purposes of the *viewpoints*. For example, the *viewpoint* may draw false conclusions from the small number of failures being reported, if the data collectors do not object that not every failure identified during test has been reported due to pressure of time.

The *viewpoints* (and only them) can draw conclusions from the data that are highly dependent on the context of the measurement program. The underlying rationale leading to conclusions

and all related explanations must be documented. This is necessary in order for those conclusions to be questioned and refined later on if inconsistent or complementary conclusions are drawn during subsequent feedback sessions.

Furthermore, only the *viewpoints* can realize that additional data may be needed to answer their questions. Such needs for new types of data should be used to update the GQM plan.

The interpretation of the data should lead to identifying weaknesses of the processes in place and to discussing possible improvement strategies. On one project, the authors experienced the following situation. A special kind of faults was frequently reported after delivery. According to the process, these faults should have been detected during testing using a particular commercial testing tool. Why did this not happen? There were many plausible reasons, but the right one(s) could only be identified by the people who performed the testing. The developers knew that the provided testing tool was important, but the tool was so user unfriendly that only one tester knew how to use it. On the project in question, this developer was not fully available so that the tool was not used. This appeared to be the most probable explanation during feedback sessions and could not have been deduced by the measurement analysts themselves. Corrective actions based on these conclusions were to provide training on the tool for all the testers and/or select a more user-friendly tool on the market.

9. Establishing a Process Improvement Action Plan

Various types of common improvement opportunities raised by measurement are outlined in the first subsection. Guidelines are then provided to identify these improvement opportunities. Last, different strategies to identify promising improvement solutions are discussed. Our goal in this section is to provide a structured overview of measurement-based improvement opportunities and how to proceed with them.

9.1. Different Types of Improvement Opportunities

In the context of a goal-driven measurement program, lessons learned based on a thorough data analysis and interpretation lead to various opportunities for improvement. A non-exhaustive list of typical recurring improvement opportunities is provided below:

- *Identification of unsuitable or low quality development artifacts*, e.g., systematic inconsistencies and incompleteness in requirement documents.
- *Identification of error-prone and/or inefficient activities*, e.g., inadequate quality assurance procedures, too low defect detection rates in inspections early in the process, too many inconsistencies are introduced in the specification documents, requirement analysis is too imprecise and therefore the requirements are unstable, etc.
- *Interfacing problems between phases*, e.g., inconsistent exit and entry criteria (what is provided from one phase to the other is not what is expected), unclear distribution of responsibilities between phases (functional units in the context of a matrix organization), inadequate intermediate products (not fully usable for the next phase of development). This is a type of problem most commonly encountered in

organizations having a functional or matrix structure where teams change from one phase to the other as well as the management hierarchy.

- *Management problems*, e.g., inaccurate resource and schedule planning, personnel management problems (high personnel turnover, lack of training, lack of motivation).

9.2. Identification of Improvement Opportunities

The identification of improvement opportunities is based on existing descriptive process models and on a careful analysis of the distribution of effort and defects across phases, activities, and artifacts. In general, one should look at the following aspects:

- differences in proportion between categories of defects according to their type, origin, cause, etc.
- associations between defect categories and
    - phases/activities and life cycle products where introduced
    - phases/activities where detected
    - various products' parts, e.g., subsystems
- activities' and phases' relative effort and duration

Unexpected distributions or associations may be indicators of problems. Examples would be:

- High coding effort during the specification phase may be due to intensive prototyping because of difficulties for the developers to understand the application domain (lack of training) and poor communication with the application domain specialists (management or logistic problem).
- High rework effort during integration test and validation test phases may be due to poor detection rates of inspections and unit testing.

9.3. Assessing Potential Solutions

Once problems have been clearly identified, the search for sound and economically viable solutions starts. New technologies and methods should be introduced with care in an organization. Any method, technique, and language should always be carefully evaluated before spreading its use across the organization. The fact that a technology has been successfully used in another organization is not a guarantee of success because organization goals, skills, application domain, and economic constraints may differ widely. Most likely, a new technology will need to be adapted and will have a negative impact until the learning phase is completed. New technologies may be assessed by different means but, in all cases, they need to be studied carefully and empirically on pilot projects and/or during training sessions.

Different types of empirical investigations may be used. The two main ones can be briefly and informally described as follows:

- *Case Studies* [Yin94]: One or a small number of pilot projects are usually monitored. The new technology is introduced on all pilot projects with little control on influencing factors. There is usually no "control" project where the new technology is not used and

against which results can be easily compared. Results are interpreted by relying heavily on interviews and a careful qualitative analysis of the process. When data are collected, which is recommended, comparisons to the baselines may be performed.

- *Controlled experiments*: The size of the sample (usually individuals) under study allows for the derivation of statistically significant results. The new technology is introduced on a part of the sample, the other part being used for comparison. These parts are selected randomly. The factors influencing the impact of the new technology are largely controlled for.

The descriptions above are a rough but relatively representative generalization. These two types of investigation represent the extreme points of a range of empirical research designs. Many intermediary strategies exist and may be better suited, e.g., quasi-experimental designs [JSK91].

The two types of investigation have different drawbacks, strengths, and therefore purposes. We will briefly discuss them in the following paragraphs:

Case studies:

- *Strengths:* low cost, can be easily performed in a real field setting, useful to identify new issues to be investigated, suited to understand the why and how of phenomena.
- *Weaknesses:* no statistically significant results can be obtained, many threats to the validity of the conclusions that can be drawn, more difficult to perform well (e.g., concerning data analysis) and requires high application domain expertise, difficult to ensure that the pilot project is representative (e.g., task's size and complexity).

Despite their important role in the process improvement process [BEM95a], case studies cannot provide high validity results, i.e., results that unambiguously prove the existence of causal relationships (e.g., between the use of technique and lower error density rates). Furthermore, these results are difficult to generalize to other experimental conditions. When possible, other kinds of empirical studies, such as quasi-experiments and controlled experiments, should be performed as well to supplement the results obtained with case studies.

Controlled Experiments:

- *Strengths:* statistically significant results, causal relationship may be demonstrated, effects of new technology may be more precisely estimated.
- *Weaknesses:* high cost, difficult to perform in field setting, only useful for (dis)confirming well stated hypotheses and theories.

Controlled experiments are in general more expensive because they require more extensive and rigorous data collection procedures, more subjects, and more time. However, only controlled experiments are likely to provide firm answers about software technology improvement. They

are usually perfectly suited to be used during training sessions when introducing the new technology in the organization.

One effective strategy is to combine the use of controlled experiments during training exercises and case studies on pilot projects. Because these two investigation strategies have complementary weaknesses and strengths, if consistent results are obtained, each investigation reinforces the other's results.

10. Conclusion

Setting up a successful measurement program for process improvement is a necessity but is a challenging undertaking. The reasons are multiple. Measurement needs to be performed from various points of views, encompasses numerous attributes, models, and interdependencies between them. Furthermore, many psychological issues have to be addressed to increase chances of success.

For this reason, goal-oriented measurement combined with explicit modeling (e.g., process, quality, etc.) can greatly help structure and provide rigor to the measurement plan. This in turn allows for completeness and consistency analysis of the plan. In addition, communication among the measurement program participants and users is improved, because supported by clear and explicit documentation.

In this paper, we have provided practical guidelines to all the steps required to address the issues mentioned above and to increase the chances of measurement to lead to actual process improvement. Additional guidelines concerning the implementation of the measurement plan (collection, analysis, interpretation) are given within the context of the GQM paradigm.

Future work includes formalizing better the structure and content of the measurement plan so that better automated support can be provided. Thus, the complexity will be easier to cope with for the measurement analysts and improved guidelines will be available for data collectors.

Acknowledgments We thank Frank Bomarius, Alfred Broeckers, Khaled El Emam, Christopher Lott, Sandro Morasca, Dietmar Pfahl, Carsten Tautz, and Isabella Wieczorek for their comments on the paper.

Datrix is a trademark of Bell Canada

GEN++ is a trademark of AT&T

SAS is a trademark of SAS Institute

Statistica is a trademark of StatSoft

References

[AK94] J.W. Armitage and M.I. Kellner. “A Conceptual Schema for Process Definitions and Models.” In Proceedings of *the Third International IEEE Conference on the Software Process*, pages 153-165, Reston, VA, USA, October 1994.

[Ash83] H.B. Asher. “Causal Modeling.” *Series: Quantitative Applications in the Social Sciences,* Sage Publications, 1983.

[Bas93] V. Basili “Applying the Goal/Question/Metric Paradigm in the Experience Factory.“ Presented at the 10th Annual CSR Workshop in Amsterdam, October 1993, to appear in a book entitled *Software Quality Assurance: A Worldwide Perspective*, by Chapman and Hall.

[Bas95] V. Basili “The Experience Factory and its Relationship to other Quality Approaches” , *Advances in Computers*, Vol.41, Academic Press, 1995. Edited by M. Zelkowitz.

[BBC+96] V. Basili, L. Briand, S. Condon, Y. Kim, W. Melo, and J. Valett. “Understanding and Predicting the Process of Software Maintenance Releases.” In Proceedings of *the 18th IEEE International Conference on Software Engineering*, pages 464-474, Berlin, Germany, March 1996.

[BCR94] V. R. Basili, G. Caldiera, and H. D. Rombach. “Measurement”, In John J. Marciniak, editor, *Encylopedia of Software Engineering*, volume 1, pages 528-532. John Wiley & Sons, 1994.

[BR88] V. Basili and H. D. Rombach, ”The TAME Project: Towards Improvement- Oriented Software Environments”, *IEEE Transactions on Software Engineering*, 14 (6), pages 758-773, June, 1988.

[BW84] V. Basili and D. Weiss, ”A methodology for collecting valid Software Engineering Data”, *IEEE Transactions on Software Engineering*, 10 (6), pages 728-738, November 1984.

[BK95a] Andreas Birk and Ralf Kempkens. “Introduction to Goal-Oriented Measurement: Tutorial Package.” ESPRIT Project #9090 “PERFECT“ and University of Kaiserslautern, Kaiserslautern, Germany 1995.

[BK95b] A. Birk and R. Kempkens. “Participation of Project Teams in Measurement Programs.” Tutorial Package. ESPRIT Project #9090 “PERFECT” and University of Kaiserslautern, Kaiserslautern, Germany 1995.

[Bri93] L. Briand. “Quantitative Empirical Modeling for Managing Software Development: Constraints, needs and solutions. In D. Rombach, V. Basili, R. Selby, editors, “*Experimental Software Engineering Issues: Critical Assessment and Future Directions.*” , pages 158-163, Springer-Verlag, 1993.

[BBH93] L. Briand, V. Basili and C. Hetmanski. "Developing Interpretable Models with Optimized Set Reduction for Identifying High Risk Software Components," *IEEE Trans. Software Eng.*, SE-19 (11):1028-1044.

[BBK+94] L. Briand, V. Basili, Y.M. Kim, and D. R. Squier. “A Change Analysis Process to Characterize Software Maintenance Projects.” In Proceedings of the *International Conference on Software Maintenance*, pages 38-49, Victoria, Canada,1994.

[BEM95a] L. Briand, K. El Emam, and W. Melo. “An Inductive Method for Process Improvement: Concrete Steps and Guidelines.” In proceedings of *the ESI-ISCN’95 conference*, Vienna, Austria, September 1995.

[BEM95b] L. Briand, K. El Emam, S. Morasca. “Theoretical and Empirical Validation of Software Product Measures.” ISERN technical report 95-03, 1995.

[BEM96] L. Briand, K. El Emam, S. Morasca. “On the Application of Measurement Theory in Software Engineering.” *Empirical Software Engineering: An International Journal*, 1 (1), 1996.

[BMS+95] L. Briand, W. L Melo, C. Seaman, and V. Basili. “Characterizing and assessing a large-scale software maintenance organization.” In *Proceedings of the 17th. International Conference on Software Engineering*, pages 133-143, Seattle, WA. 1995.

[BMB94a] L. Briand, S. Morasca, and V. Basili, “Defining and Validating High-Level Design Metrics”, CS-TR-3301, Version 2, University of Maryland, College Park, MD, 20742, April 1994. Submitted for publication.

[BMB96] L. Briand, S. Morasca, and V. Basili, “Property-Based Software Engineering Measurement”, *IEEE Transactions on Software Engineering*, 22 (1), pages 68-86, January, 1996.

[BTH93] L. Briand, W. Thomas, and C. Hetmanski, “Modeling and Managing Risk Early in Software Development”, in Proceedings of *the 15th International Conference on Software Engineering*, pages 55-65, Maryland, May 1993.

[BDT96] A. Broeckers, C. Differding, and G. Threin. “The Role of Software Process modeling in Planning Industrial Measurement Programs.” In Proceedings of *the Third International IEEE Software Metrics Symposium*, pages 31-40, Berlin, March 1996.

[CY79] L. Constantine, E. Yourdon, "Structured Design", Prentice Hall, 1979.

[CZ79] E. Carmines and R. Zellner. “Relability and Validity Assessment.” *Series: Quantitative Applications in the Social Sciences*, Sage Publications, 1979.

[DHL96] C. Differding, B. Hoisl, and C. Lott. “Technology Package for the Goal Question Metric Paradigm.” Internal Report 281-96, Department of Computer Science, University of Kaiserslautern, 67653, Kaiserslautern, Germany, April 1996.

[Fen91] N. Fenton. “Software Metrics: A rigorous approach.”, Chapman & Hall, 1991, London.

[Eri92] H. Eriksson. “A Survey of Knowledge Acquisition Techniques and Tools and their relationship to Software Engineering.” *The Journal of Systems and Software*, 19:97-107, 1992.

[GHW95] C. Gresse, B. Hoisl, and J. Wuest. “A Process Model for Planning GQM-based Measurement.” Technical Report STTI-95-04-E, Software Technology Transfer Initiative Kaiserslautern, University of Kaiserslautern, Germany, October 1995.

[Hoi94] B. Hoisl. “A Process Model for Planning GQM based measurement.” Technical Report STTI-94-06-E, Software Technology Transfer Initiative Kaiserslautern, University of Kaiserslautern, Germany, April 1994.

[JSK91] C. M. Judd, E. R. Smith, and L. H. Kidder. Research, “*Methods in Social Relations*.” Harcourt Brace Jovanovich College Publishers, 1991.

[KPM92] T.M. Khoshgoftaar, A.S. Panday, and H.B. More. “A Neural Network Approach for Predicting Software Development Faults.” In Proceedings of *the Third International IEEE Symposium on Software Reliability Engineering*, pages 83-89, North Carolina, October 1992.

[Nat94] National Aeronautics and Space Administration. “Software Measurement Guidebook.” Technical Report SEL-94-002, NASA Goddard Space Flight Center, Greenbelt MD 20771, July 1994.

[PCC+91] M. C. Paulk, W. Curtis, M. B. Chrissis, and C. V. Weber. “Capability Maturity Model, Version 1.1.”*IEEE Software*, 10(4), pages 18-27, July 1993.

[Rom91] H. D. Rombach. “Practical Benefits of Goal-Oriented Measurement.” In N. Fenton and B. Littlewood, editors, *Software Reliability and Metrics*, pages 217-235. Elsevier Applied Science, London, 1991.

[Yin94] Robert K. Yin. “Case Study Research.” *Applied Social Research Methods Series, Volume 5*. Sage Publications, 1994.

Document Information

Title:	Practical guidelines for measurement-based process improvement
Date:	Dec. 31,1996
Report:	IESE-004.96/E
Status:	Final
Distribution:	Public
	also published as ISERN-96-05

Copyright 1996, Fraunhofer IESE.
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means including, without limitation, photocopying, recording, or otherwise, without the prior written permission of the publisher. Written permission is not needed if this publication is distributed for non-commercial purposes.