

# MEASUREMENT FRAMEWORKS

## MEASUREMENT FRAMEWORKS

---

**Measurement is not** just the collection of data/metrics

calendar time

number of open problems

number of defects found in inspections

cyclomatic complexity

machine time

lines of code/module

total lines of code

severity of failures

total effort

total number of defects

lines of code/staff month

number of failures during system test

## MEASUREMENT FRAMEWORKS

**Measurement must** be driven by goals and models

for the processes and products

projects

organization

There are a variety of measurement frameworks that help you define the appropriate metrics and interpret the results of the measurement process

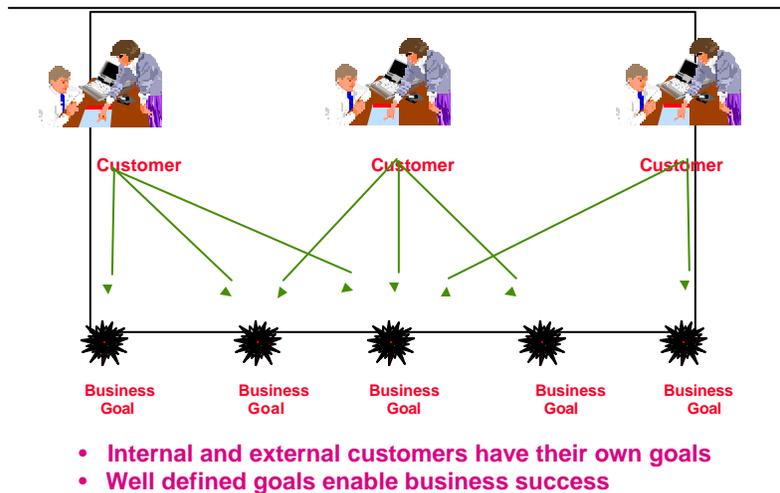
The most common frameworks in use are:

Goal/Question/Metric Paradigm (GQM)

Quality Function Deployment (QFD)

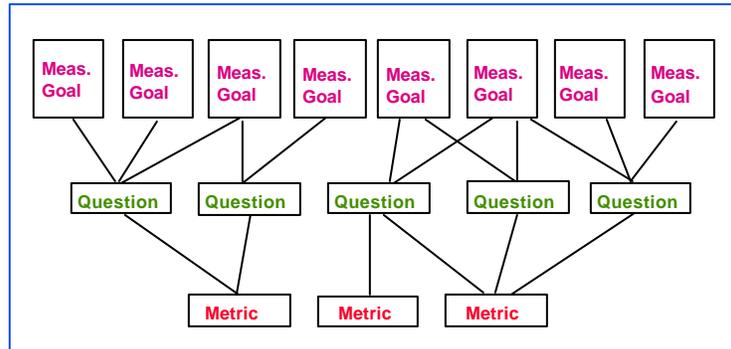
Software Quality Metrics (SQM)

## THE MEASUREMENT INFRASTRUCTURE



## THE MEASUREMENT INFRASTRUCTURE Goal Based Measurement

---



- Each metric supports multiple goals
- Questions focus metric selection and in-process analysis

## MEASUREMENT FRAMEWORKS

---

### Goal/Question/Metric Paradigm

A mechanism for defining and interpreting operational, measurable goals

It uses four parameters:

a model of an **object of study**,

e.g., a process, product, or any other experience model

a model of one or more **focuses**,

e.g., models that view the object of study for particular characteristics

a **point of view**,

e.g., the perspective of the person needing the information

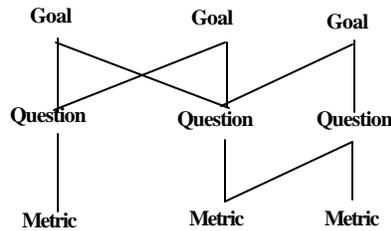
a **purpose**,

e.g., how the results will be used

to generate a **GQM model**

relative to a **particular context (environment)**

## GOAL/QUESTION/METRIC PARADIGM Goal and Model Based Measurement



A Goal links two models: a model of the **object of interest** and a model of the **focus** and develops an integrated GQM model

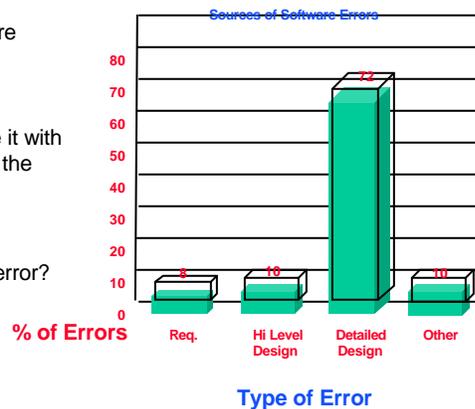
**Goal:** Analyze the **final product** to **characterize** it with respect to the **various defect classes** from the point of view of the **organization**

**Question:** What is the error distribution by phase of entry

**Metric:** Number of Requirements Errors, Number of Design Errors, ...

## DEFINING MEASUREMENT GOALS A GOAL/QUESTION/METRIC EXAMPLE

- **Business Goal**
  - Understand problem areas in the software business
- **A Measurement Goal**
  - Analyze the final product to characterize it with respect to the various defect classes from the point of view of the organization
- **Question**
  - What is the error distribution by type of error?
- **Metrics**
  - Number of Requirements Errors,
  - Number of Design Errors, ...



## GOAL/QUESTION/METRIC PARADIGM

### Overview of the GQM Approach

---

**Develop** a set of corporate, division and project **business goals and associated measurement goals** for productivity and quality.

**Generate questions** (based upon models) that define those goals as completely as possible in a quantifiable way.

**Specify** the **measures** needed to be collected to answer those questions and track process and product conformance to the goals.

**Develop mechanisms** for data collection.

**Collect, validate** and **analyze** the **data** in real time to provide feedback to projects for corrective action.

**Analyze** the **data** in a postmortem fashion to assess conformance to the goals and make recommendations for future improvements.

## GOAL/QUESTION/METRIC PARADIGM

### Goal Generation Template

---

Goals may be defined for any object, for a variety of reasons, with respect to various models of quality, from various points of view, relative to a particular environment.

Analyze some

(**object of study**: process, product, other experience model)

to

(**purpose**: characterize, evaluate, predict, motivate, improve)

with respect to

(**focus**: cost, correctness, defect removal, changes, reliability, user friendliness,...)

from the point of view of

(**customer**: user, customer, manager, developer, corporation,...)

in the following **context**

(problem factors, people factors, resource factors, process factors,...)

## GOAL/QUESTION/METRIC PARADIGM

### Template parameters

---

**Context:** defines all the factors in the environment that can effect this study

**Objects of study:** processes, products, other experience models

**Associated Models:**

- the process itself,
- how the performance of the process will be evaluated
- the domain
- how well the domain is understood
- how well the object on which the process is being performed is understood

**Indirect Models:**

- training, experience, ability of the people in the process, the domain

**Focus Models:** cost, correctness, defect removal, changes, reliability, user friendliness,...

## GOAL/QUESTION/METRIC PARADIGM

### Template parameters

---

**Purpose of study:**

**understanding or characterization** requires:

- model of the object and any data we collect helps provide us with some insight

**evaluation** requires:

- some basis for comparison (historical data base) or some absolute measure

**prediction** requires:

- a pattern in the data that allows us to develop a consistent model based upon factors that we have available or can reasonably estimate

**motivation** or **improvement** requires:

- an accurate model that represents a true understanding of the object or positive quality we are modeling

## GOAL/QUESTION/METRIC PARADIGM

### Template parameters

---

#### Point of view:

determines such things as  
when the information should be available  
who needs it  
its level of granularity  
its acceptable accuracy, etc.

#### Feedback:

What has been learned about the process, its application, the product domain, or any other process or product?  
Can we change what we are doing on this project in real time?  
How might we modify our models to represent this knowledge?

## GOAL/QUESTION/METRIC PARADIGM

### Characterizing Goals

---

Analyze the **software products**  
in order to **characterize** them with respect to  
**development error rates**  
**cost in staff months**  
**% of code reused**  
from the point of view of the **organization** relative to the **SEL environment**

Analyze the **software processes**  
in order to **characterize** them with respect to  
**effort distributions**  
**classes of errors**  
**source code growth**  
from the point of view of the **organization** relative to the **SEL environment**



## DEFINING GOAL RELATIONSHIPS

### Example

---

- **Business Goal:** Cycle Time Reduction
- **Measurement Goal:** Improve Software Delivery Schedule by 10%
- **Example Support Measurement Goals:**

**Goal 1:**  
Build a  
Schedule  
Baseline

Analyze  
**representative projects**  
in order to  
**characterize** them  
with respect to  
**calendar time & effort**  
**for each phase, activity**  
**and in total**  
from the point of view of  
**the Corporation**

**Goal 2:**  
Determine  
how good  
your prediction  
capabilities are

Analyze  
**representative projects** in  
order to  
**characterize** them  
with respect to  
**variation from predicted**  
**schedule for each phase,**  
**activity and in total**  
from the point of view of  
**the Corporation**

## GOAL/QUESTION/METRIC PARADIGM

### Process Goal: Question Guidelines

---

**Process Conformance:**

Characterize the process quantitatively and assess how well the process is performed. (How quantitatively can we characterize a process?)

**Domain Understanding:**

Characterize the object of the process and evaluate the knowledge of the object and its domain by the process performers. (How can we quantify this knowledge?)

**Focus:**

What is the aspect of the process which is of interest? Analyze the output of the process according to some quality model and some viewpoint.

**Feedback:**

What has been learned about the process, its application, the product domain, or any other process or product?

## GOAL/QUESTION/METRIC PARADIGM

### Process Goal: Example

---

Consider the following situation:

An organization has decided that its customers are reporting too many failures and that most of these problems should have been caught during system test.

It is considering adopting a new system test process (a risk and expense) and wants to try the new system test process on several projects to determine if it is doable and more effective than what it has been doing

The organization has data on the number of faults identified by the system test process and released to the field for various products. It uses a waterfall type life cycle process, ...

To make an informed decision it must define the new test process, determine if it is being followed, characterize how well the process is identifying faults, and compare it to what they were doing before

## GOAL/QUESTION/METRIC PARADIGM

### Process Goal: Example

---

#### **System Test Process Evaluation Goal:**

Analyze the system test process for the purpose of evaluation with respect to defect slippage from the point of view of the corporation.

#### **System Test Process Model:**

Goal: Generate a set of tests consistent with the complexity and importance of each requirement.

Procedure: (1) Enumerate the requirements, (2) Rate importance by marketing, (3) Rate complexity by system tester, (4) ...

#### **System Test Process Definition Goal:**

Analyze the software product requirements for the purpose of characterizing them with respect to a set of tests consistent with the complexity and importance of each requirement from the point of view of the tester and marketer respectively.

## GOAL/QUESTION/METRIC PARADIGM

### Defect Slippage Model

Analyze the system test process for the purpose of evaluation with respect to defect slippage from the point of view of the corporation.

#### Defect Slippage Model:

Let **Fc** = the ratio of faults found in system test to the faults found after system test on this project.

Let **Fs** = the ratio of faults found in system test to the faults found after system test in the set of projects used as a basis for comparison.

Let **QF = Fc/Fs** = the relationship of system test on this project to faults as compared to the average the appropriate basis set.

## GOAL/QUESTION/METRIC PARADIGM

### Simple Interpretation of Defect Slippage Model

**if QF > 1** then  
    method better than history  
    check process conformance  
    if process conformance poor  
        improve process or process conformance  
    check domain conformance  
    if domain conformance poor  
        improve object or domain training

**if QF = 1** then  
    method equivalent to history  
    if cost lower than normal then method cost effective  
    check process conformance

**if QF < 1** then  
    check process conformance  
    if process conformance good  
        check domain conformance  
        if domain conformance good  
            method poor for this class of project

**PROCESS GOAL EXAMPLE**  
**Test Method Process Conformance**

---

Characterize the test method experience of the test team.  
(subjective rating per person)

- 0 - none
- 1 - have read the manuals
- 2 - have had a training course
- 3 - have had experience in a laboratory environment
- 4 - have used on a project before
- 5 - have used on several projects before
- x - no response

How many requirements are there?  
(enumerate them)

What is the importance of testing each requirement?  
(Subjective rating 0 - 5 by marketing and testers)

- 0 - not important, could be left out
- 1 - not too important, may affect some users
- 2 - mildly important, will affect some users
- 3 - important, should affect most users
- 4 - extremely important, part of the essence of the system
- 5 - critical, without this the system is useless
- x - don't know

**PROCESS GOAL EXAMPLE**  
**Test Method Process Conformance**

---

We are defining the process as we define conformance

Models of importance and complexity will vary with the organization, product type, etc.

What is most important is that we articulate what it is and gain agreement

What is the complexity of testing each requirement?

- How many people are required to test?
- How many configurations are needed?
- How much time is predicted to test?
- The number of concepts and interactions among those concepts?
- Estimate of the number of inputs required
- Prediction of the number of execution paths to implement the requirement
- Number of tests in the test suite

## PROCESS GOAL EXAMPLE

### Test Method Process Conformance

What is the complexity of testing each requirement?

(subjective rating 0 - 5 by tester)

- 0 - doesn't need to be tested
- 1 - easy to test, one test should do it
- 2 - reasonably easy to test, only a few ad hoc tests are needed
- 3 - not easy to test, requires carefully made up test suite
- 4 - very difficult to test, requires a lot of thought to make up a good test suite
- 5 - extremely difficult to test, requires a large, complex test suite
- x - impossible to test

What is the distribution of tests over requirements?

(number of tests/requirement)

Is the number of tests/requirement consistent with the evaluation of its complexity and importance?

- 0 - there are no tests for this requirement
- 1 - there is at least one test
- 2 - there are several tests but not nearly enough
- 3 - the number of tests are reasonable but insufficient for the requirement importance/complexity
- 4 - the number of tests are sufficient for the complexity and importance of the requirement
- 5 - the number of tests are more than adequate for the requirement importance and complexity
- x - no response

or a formula,

e.g., If importance = 5 and complexity = 5  
then number of test cases = ...

## PROCESS GOAL EXAMPLE

### Test Object Domain Understanding

#### Understanding of the Domain and Requirements

How familiar is the domain?

(subject rating 0 - 5 for each tester)

- 0 - domain new to me
- 1 - have had a course in the subject domain
- 2 - have built or tested one system in this domain
- 3 - have built and tested at least one system in this product line
- 4 - have built and tested several systems in this domain
- 5 - have tested and built several systems in this product line

How understandable are the requirements for this project?

(subjective rating 0 - 5 for each requirement)

- 0 - not understandable at all
- 1 - requirement ambiguous or not sure what it means
- 2 - not sure of the full ramifications
- 3 - reasonably clear requirement
- 4 - requirement is perfectly clear
- 5 - have successfully tested this type of requirement before

**PROCESS GOAL EXAMPLE**  
**Test Object Domain Understanding**  
**Quality of the Tests**

How precisely are the tests (inputs, results) known in advance?

(subjective rating 0 - 5)

- 0 - there were no tests for this requirement
- 1 - will make the inputs up at terminal
- 2 - know the inputs but not the results
- 3 - know the inputs and the range of the results
- 4 - know the inputs and the results
- 5 - have simulation results for the test cases

How confident are you that the result is correct?

(subjective rating 0 - 5)

- 0 - there are no results
- 1 - the results are incorrect
- 2 - not sure the results are correct
- 3 - think they are correct
- 4 - reasonably sure they are correct
- 5 - positive they are correct

What is the evaluation of the domain conformance?

**PROCESS GOAL EXAMPLE**  
**Goal Focus: Cost of Use**

What is the total cost of testing?

OR

(depending upon the level of granularity required)

What is the staff time to make a test?

What is the staff time to run a test and check the result?

What is the staff time to isolate the fault?

What is the staff time to design and implement a fix?

What is the staff time to retest?

What is the machine time used?

## PROCESS GOAL EXAMPLE

### Goal Focus: Simple Defect Slippage Model

Simple defect slippage model questions:

What is the number of faults failures discovered during system test, acceptance test and one month, six months, one year after system release on this project?

What is the number of faults failures discovered during system test, acceptance test and one month, six months, one year after system release on the set of projects classified as similar?

What is the ratio of faults in system test on this project to faults found from system test on?

What is the ratio of faults in system test on the set of similar projects to faults found from system test on?

What is the ratio of system test performance on this project to system test performance on the set of similar projects?

## PROCESS GOAL EXAMPLE

### Goal Focus: General Defect Slippage Model

What is the number of errors, faults and failures on this project  
in total,  
per line of code,  
by various classification schemes, and  
by cost to isolate, fix and overall,  
discovered during each phase of development and one month, six months, one year after system release?

What is the number of errors, faults and failures on the set of similar projects  
in total,  
per line of code,  
by various classification schemes, and  
by cost to isolate, fix and overall,  
discovered during each phase of development and one month, six months, one year after system release?

What is the ratio of weighted faults in system test on this project to faults found from system test by various classification schemes?

What is the ratio of weighted faults in system test on the set of similar projects to faults found from system test by various classification schemes?

What is the ratio of system test performance based upon the various slippage models on this project to system test performance on the set of similar projects?

## PROCESS GOAL EXAMPLE

### Feedback

---

Does the system test method need to be refined or modified?

Is more or different training needed in the method or the technology?

Is more or different training needed in the application domain?

What should be automated?

What is the input to the requirements, specification, design and code techniques, methods, and tools, and the defect detection techniques, methods, and tools?

## PROCESS GOAL EXAMPLE

### Data Sources

---

**System test tables:**

System test table 1: **Nature of requirements**  
(Filled out after baselining of requirements)

Req. #	How understandable is the requirement	Importance of testing (marketing)	(systest)
R1	5	3	2

System test table 2: **Nature of tests**  
(Filled out after test plan)

Req. #	# of tests	How well are tests known?	Difficulty of testing	Evaluation subj. stat	Are # of tests consistent with diff. & import?	Rating
R1	5	3	2	3	3/50	4

System test table 3: **Results of the tests**  
(Filled out after tests run)

Test #	Failure? Yes : No	How confident are you in the results?	# of Faults found
T1	x	3	1

## PROCESS GOAL EXAMPLE

### Defect/Change Report Form

---

For each fault, record:

- Description of fault
- Date isolated
- Date fixed
- Time to fix
- # of components changed
- # of components that had to be examined
- # of components that were examined

Classifications:

- Error origin
- Error domain
- Detection time
- Omission/commission
- Software aspect
- Failure severity

## PROCESS GOAL EXAMPLE

### Data Presentations

---

Slippage model data:

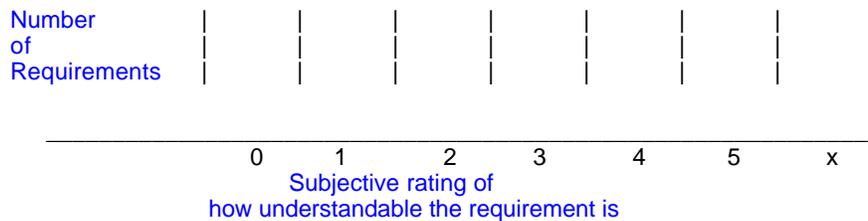
- QEs
- REs, RPEs
- Es, Ea, Eo

Histograms of:

- Number of faults found in each phase
- The number of requirements vs. subjective ratings of
  - how understandable the requirement is
  - importance of requirement
  - difficulty of testing the requirement

...

Example:



## GOAL/QUESTION/METRIC PARADIGM

### Product Goal: Question Guidelines

---

**Product Model/Definition:**

Characterize the product qualitatively independent of the perspective of interest. Aspects of interest include:

**Logical/Physical Attributes:**

Characterize the logical and physical attributes of the product e.g.,

logical attributes: application domain, function

physical attributes: size, complexity, interfaces

dynamic attributes: coverage, reliability

**Cost:**

Characterize the resources expended, e.g., effort, computer time

**Changes:**

Characterize the modifications associated with the product, e.g., enhancements, errors, faults, failure

**Context:**

Characterize the customer community and their operational profiles

## GOAL/QUESTION/METRIC PARADIGM

### Product Goal: Question Guidelines

---

**Perspective/Focus:**

Analyze the product models from each perspective of interest, e.g., reliability, user friendliness, specifying the following:

**Major model(s) used**

Specify some perspective model/definition and viewpoint

**Validity of the model for the project**

Evaluate the appropriateness of the model for the project environment

**Validity of the data collected**

Evaluate the quality of the data

**{Substantiation of the model**

Given any alternate perspectives that provide support for the quality of the results}

**Feedback:**

What has been learned about the product, the processes that produced it, or any other product that will improve this project and future projects?

## GOAL/QUESTION/METRIC PARADIGM

### Product Goal Example

---

Consider the following situation:

A project manager will need to decide if the design document is in sufficiently good shape to begin coding or whether a redesign is needed

The decision need to be made in real time, with whatever information is available

It is known that there is a well defined successful inspection process being used and it has been used before

He can make the assumption that a well defined process, if followed should be effective for purpose

## GOAL/QUESTION/METRIC PARADIGM

### Product Goal Example

---

Analyze the design document for the purpose of evaluation with respect to the design inspection defects uncovered from the point of view of the project manager.

#### Design Inspection Process Model:

Goal: Analyze the design document for the purpose of characterization with respect to its correct and complete implementation of the requirements from the point of views of the user, developer and tester.

- Procedure:
- (1) Disseminate the appropriate part of the requirements and design documents,
  - (2) Read the document by the appropriate set of readers from the appropriate points of view,
  - (3) Report defects by various classification schemes, including omission and commission defects,
  - (4) ...

## GOAL/QUESTION/METRIC PARADIGM

### Product Goal Example

---

#### Design Document Product Model/Definition:

##### Logical/Physical Attributes:

logical attributes: application domain, function

physical attributes: size: **lines of design language**, complexity, interfaces

##### Cost:

total effort, effort by activity (effort in design inspection)

##### Changes:

# of enhancements

**# faults found during design inspection**

##### Context:

Customer community: designers, coders, users, ...

## GOAL/QUESTION/METRIC PARADIGM

### ~~Simple Document/Defect Evaluation Model~~

---

---

**KLOD** = number of thousand lines of design language

**Fc** = number of faults/**KLOD** found in **design inspections** on this project

**Fs** = number of faults/**KLOD** found in **design inspections** in the set of projects used as a basis for comparison (same size, application, ...)

**QF = Fc/Fs** = the relationship of faults found on this project as compared to the average the appropriate basis set

if **QF** > 1 then QF = H (worse than history)

if **QF** <= 1 then QF = L (better than history)

**PC** = the process conformance rating on this project  
= C if inspections are performed to the definition, N otherwise

**DU** = the domain understanding rating on this project  
= S if domain understanding is satisfactory, U otherwise

## GOAL/QUESTION/METRIC PARADIGM Simple Document/Defect Evaluation Model

**QF** = H if more faults found when compared with history

**QF** = L if less faults found when compared with history

**PC** = C if inspections are performed to the definition  
N otherwise

**DU** = S if domain understanding is satisfactory  
U otherwise

<u>PC</u>	<u>DU</u>	<u>QF</u>	<u>Design-in</u>	<u>Design-out</u>	<u>Design Process</u>	<u>Inspection Process</u>
C	S	L	good	good	effective	effective
C	S	H	poor	fixed-up	not-effective	effective
N	X	X	?	?	?	?
X	U	X	?	?	?	?

## EXAMPLE G/Q/M GOALS

### Defining the System Test Process Goal:

Analyze the software product requirements for the purpose of characterizing them with respect to a set of tests consistent with the complexity and importance of each requirement from the point of view of the tester and marketer respectively.

### Evaluating the System Test Process:

Analyze the system test process for the purpose of evaluation with respect to defect slippage from the point of view of the corporation.

### Defining the Design Inspection Process Goal:

Analyze the design document for the purpose of characterization with respect to its correct and complete implementation of the requirements from the point of views of the user, developer, and tester.

### Evaluating the Design Document:

Analyze the design document for the purpose of evaluation with respect to the design inspection defects uncovered from the point of view of the project manager.

## GOAL/QUESTION/METRIC PARADIGM

### Product Goal: Example

---

Analyze the **final product** for the purpose of **evaluation** with respect to **reliability** from the point of view of the **various customer classes**.

We need a model of the

final product

reliability

the customer usage

## PRODUCT GOAL EXAMPLE

### ~~Final Product Model: Logical/Physical Attributes~~

Classify the application domain and solution domain of the final product?

What is the function of the final product?

What is the size of the final product with respect to:

pages or lines of user documentation,  
source lines with comments,  
executable statements,  
program units, e.g., functions, processes, components,  
the number of requirements, etc.?

What is the complexity of the product with respect to:

syntactic control (e.g., cyclomatic complexity, software science metrics),  
data use (e.g., span, data bindings), etc.  
for each appropriate unit?

What is the distribution of programming language features used?

What are the run-time time and space constraints?

## PRODUCT GOAL EXAMPLE

### Product Model: Cost

What is the effort by phase, activity, personnel type, used to develop the system?

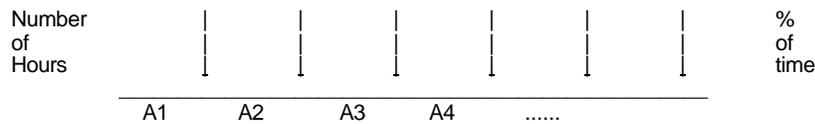
How much machine time was used to develop the system; documentation as well as other aspects?

How much calendar time was used to develop the system, each component?

#### **Presentation:**

Histogram of the number of hours and the percent of time vs. the time card accounting sub-codes.

Histogram of the # of hours and the percent of time vs. the various activities (from the beginning of data collection and during the last report period).



where A1, A2, A3, etc., are the different activities for which time is being accounted.

### Product Model: Changes

#### **A. Enhancements**

What is the number of enhancements  
(normalized by calendar time, phase of project)?

What are the number of enhancements categorized by  
type (requirements, specification, design, architecture, planned enhancements, insert/delete debug code, improve clarity, optimize: space or time, feature, enhancement, bug)?  
market/external and internal needs?  
size, e.g., number of lines of code, number of components affected, etc.?  
disposition, e.g., rejected as a change, not relevant, under consideration, being worked on, completed, saved for next enhancement?  
level of document changed?

How many customers are affected by the changes?

What is the Trouble Report history profile for each change?

## PRODUCT GOAL EXAMPLE

### Product Model: Changes

#### B. Defects

What are the number of errors in total and by type, e.g., error origin, error domain?

What are the faults in total and by type, e.g., fault entry time, fault detection time/phase, (omission, commission), software aspect, mechanism of discovery, product level?

**What are the failures in total and by type, e.g. severity, failure detection time/phase?**

What are the number of errors, faults and failures in total and by type, normalized by phase and calendar time?

#### **Presentations:**

Histograms of the number of changes/defects by various classes.

Graphs of # of changes/defects initiated and closed vs. calendar time  
Number of errors/component for each component

## PRODUCT GOAL EXAMPLE

### Product Model: Context

#### A. Customer community

**What classes of customers are expected to use the system?**

**What is the matrix of functional requirements vs. customer classes?**

**What is the matrix of functional requirements vs. components of the system?**

#### B. Operational profile

**What percent of the system is expected to be executed by each customer class?**

## PRODUCT GOAL EXAMPLE

### Product Model: Data - Various Matrices

Requirements X Customer Usage Matrix, e.g.,

#### CUSTOMER

REQUIREMENTS		percent use	
		by customer	

Components X Requirements Coverage Matrices, e.g.,

#### REQUIREMENTS

COMPONENTS		percent $R_i$ covered by $X_j$	= 100
		percent $X_i$ covers $R_j$	>= 100

Components X Customer Coverage Matrix

Test X Components Coverage Matrix

Test X Requirements Coverage Matrix

## PRODUCT GOAL EXAMPLE

### Focus Model: Reliability

**Perspective:** Examine the reliability of the system from the user's point of view.

**Major model used:** some MTTF model (e.g., Musa)

What is the MTTF for the overall system, during system test, acceptance test, and for each customer base during operation?

What is the estimated fault density for the model? target MTTF?

#### **Presentations:**

Graphs of the MTTF vs. execution time and calendar time containing actual mean-time to failure data and projected mean-time to failure data

## PRODUCT GOAL EXAMPLE

### Focus Model: Reliability

---

#### Validity of the model for the project:

How many requirements are there? What is the distribution of tests/requirement?

What is the customer/requirements matrix with probabilities (usage weightings) for each requirement?

How accurately does that describe each user operational profile?

Do the test cases reflect the customer/requirements probability matrix, i.e., is the test suite made up according to the different operational scenarios of the customer bases?

Is the probability distribution of tests run during system/acceptance test based on customer OP profile?

## PRODUCT GOAL EXAMPLE

### Focus Model: Reliability

---

#### Validity of the model for the project:

Were the test cases randomized and distribution based on the customer operational profile from the previous question when they were run?

Are corrections being made as failures are discovered?

Are new errors being introduced into the system during testing and debugging?

Is the model being run for each customer base with a different operational scenario?

Do the cumulative changes imply that the reliability model should be restarted?

## PRODUCT GOAL EXAMPLE

### Focus Model: Reliability

#### Validity of data collected:

How valid is the failure data and the associated failure times?

#### Substantiation of the model:

##### Use of error profiles:

What is number of faults detected per 1000 lines and per component (one-half system test, system test, one-half certification, end of certification)?

Does this agree with previous projects' fault history and reliability ratings?

##### Use of coverage data:

What is the requirement, component coverage of the test plan at various points in time?

Do the requirement and component coverage correspond to the distribution in the (requirement, component) and (component, customer) matrices?

## PRODUCT GOAL EXAMPLE

### Focus Model: User Satisfaction

Perspective: Examine the user satisfaction with the final system?

Major Model(s) used: User Response to System

How many failures are reported by the users?

How many clarifications are requested by the user?

Is the use of the system growing, shrinking, staying the same?

How many requests are there for functional enhancements?

How many functional change requests are real versus functions already in the system?

How many performance change requests are being submitted?

Validity of the Model for the Project: User Representation

Has a user committee been appointed that covers all user types?

Has the user community had any input into the requirements?

Are there prototype screens for the users to play with?

Is the user part of the test plan development?

Is the user part of the test team?

Validity of the Data Collected:

How valid is the data collected?

## **PRODUCT GOAL EXAMPLE**

### **Focus Model: User Satisfaction**

---

Substantiation of the Model: User Subjective Evaluation

How responsive is the system to user request for functionality?

How responsive is the system to user request for performance?

How does the user rate the system with respect to:

ease of use

functionality

performance

ease of understanding the documentation?

(a questionnaire should be made up)

Why did you buy it?

Did you get more, less, same as expected?

How many others have you used?

Would you recommend it to a friend? A competitor?

Would you reorder?

Which competitors did you consider?

How many requests are there for functional enhancements?

## **GOAL/QUESTION/METRIC Approach**

---

### **Practical Application**

Four phases:

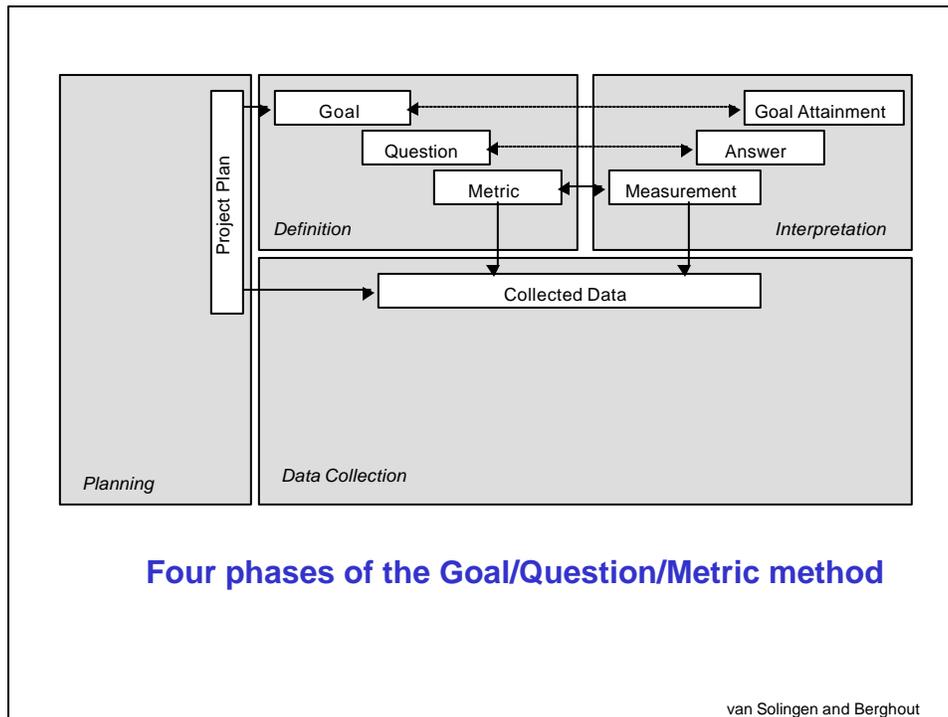
Planning

Definition

Data Collection

Interpretation

from: van Solingen and Berghout, The GQM Approach, McGraw Hill



## GOAL/QUESTION/METRIC Approach

To cost out a GQM program, we need to cost out the individual activities

Typical Activities:

1. *GQM program planning.* This includes the identification of available input, preconditions and constraints, the set up of an infrastructure, the selection of an improvement area, the selection of a project, initial planning of the measurement program, and the preparation and training of the Project team.
2. *Identify and define GQM goals.* This includes the characterization of the project and organization, identification and selection of improvement goals, definition of the measurement and GQM goals, modeling of the relevant software processes involved in the measurement program, and the identification of artifacts to be reused.
3. *Conduct GQM interviews.* This includes studying documentation, defining, scheduling and inviting interviewees, briefing of a project team, conducting GQM interviews and reporting them.

van Solingen and Berghout

## GOAL/QUESTION/METRIC Approach

---

Typical Activities (cont.):

4. *Develop GQM deliverables.* This includes definition, reviewing, and refining of a GQM plan, the definition of a measurement plan, identification and definition of data collection procedures, reviewing and refinement of a measurement plan, and development of an analysis plan.
5. *Data collection.* This includes a data collection trial to test data collection procedures and forms, briefing the project team and kick-off of the measurement program, collection, validation, coding and storage of measurement data.
6. *Data analysis and interpretation.* This includes analyses of the measurement data, preparation of the presentation material, planning, conducting and reporting of the feedback sessions.

van Solingen and Berghout

## GOAL/QUESTION/METRIC Approach

---

### Planning

Project Plan

1. Establish GQM Team
2. Select Improvement Area
  - possible improvement areas
  - management directives
3. Select Application Project and Establish Project Team
4. Create Project Plan
5. Training and Promotion
  - measurement training
  - promotion material

van Solingen and Berghout

## GOAL/QUESTION/METRIC Approach

---

### Planning

The main activities of the GQM team are:

- *Plan* measurement programs within development projects.
- Carry out measurement *definition* activities and develop GQM deliverables.
- Check *data collection* by the project team and process available data.
- Prepare *interpretation* of the measurement data by organizing feedback sessions.
- Moderate feedback sessions.
- Report progress to project team and management, and disseminate and package results.

van Solingen and Berghout

## GOAL/QUESTION/METRIC Approach

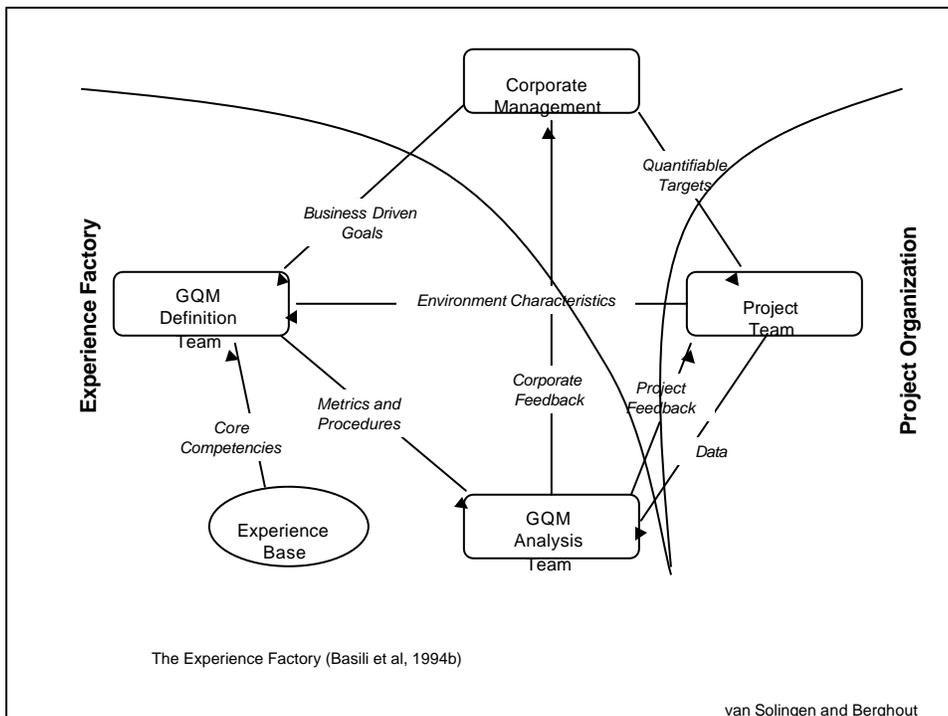
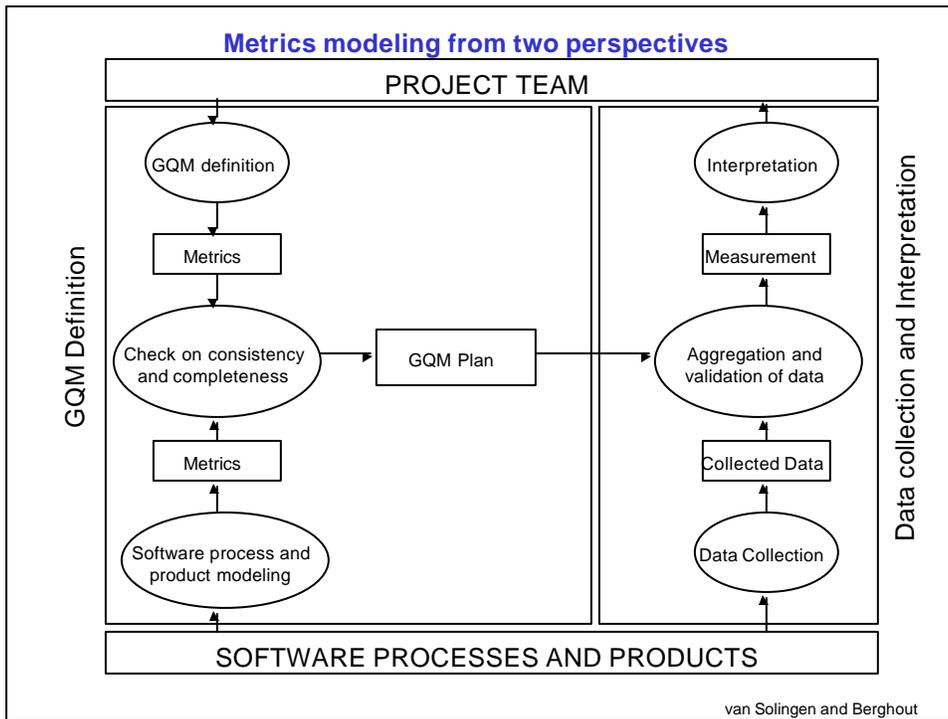
---

### Planning

A GQM team should:

- Be independent from project teams, and have no interest in measurement results.
- Possess sufficient background knowledge on the objects of measurement.
- Keep in mind that the project team 'owns' the improvement program, because a project team is most knowledgeable on a project.
- Be improvement oriented, which includes that it is willing to improve itself too.
- Be enthusiastic, in order to motivate the project team.

van Solingen and Berghout



## **GOAL/QUESTION/METRIC Approach**

---

### **Definition**

GQM Definition Phase:

1. Define measurement goals
2. Review or produce software process models
3. Conduct GQM interviews
4. Define questions and hypotheses
5. Review questions and hypotheses
6. Define metrics
7. Check metrics on consistency and completeness
8. Produce GQM plan
9. Produce measurement plan
10. Produce analysis plan
11. Review plans

van Solingen and Berghout

## **GOAL/QUESTION/METRIC Approach**

---

### **Definition**

Goal Definition Questions:

1. What are the strategic goals of your organization?
2. What forces have an impact on your strategic goals?
3. How can you improve your performance?
4. What are your major concerns (problems)?
5. What are your improvement goals?
6. How can you reach your improvement goals?
7. What are possible measurement goals, and what are their priorities?

van Solingen and Berghout

## GOAL/QUESTION/METRIC Approach

---

### Abstraction Sheets

To help define goals, one can use an abstraction sheet, which has four parts:

- *Quality focus*: what are possible metrics to measure an object of a goal, according to the project members?
- *Baseline hypothesis*: what is the project member's current knowledge with respect to these metrics? His or her expectations are documented as 'baseline hypotheses' of the metrics.
- *Variation factors*: which (environmental) factors does a project member expect to be of influence on the metrics?
- *Impact on baseline hypothesis*: how could these variation factors influence the actual measurements? What kind of dependencies between the metrics and influencing factors are assumed?

van Solingen and Berghout

Object	Purpose	Quality Focus	Viewpoint
Delivered Product	Understanding	Reliability and its causes	Project Team
Quality Focus Number of failures: <ul style="list-style-type: none"> <li>▪ by severity</li> <li>▪ by detection group</li> <li>▪ number of faults</li> <li>▪ by module</li> </ul>		Variation Factors  Level of reviewing	
<u>Baseline Hypotheses (estimates)</u> Distribution of failures: <ul style="list-style-type: none"> <li>▪ by severity:               <ul style="list-style-type: none"> <li>▪ Minor 60%</li> <li>▪ Major 30%</li> <li>▪ Fatal 10%</li> </ul> </li> </ul>		<u>Impact of Variation Factors</u>  The higher the level of reviewing, the less minor failures will be detected after release	

### Example of an abstraction sheet

van Solingen and Berghout

## **GOAL/QUESTION/METRIC Approach**

---

### **Data Collection**

Data Collection Procedures should specify

For a certain metric, which person should collect the data?

When should the data be collected?

How can the data be collected most efficiently and effectively?

To whom should the collected data be delivered?

The data can be collected via

manual forms, electronic forms, e-mail forms, spreadsheet forms

data base driven, web page driven

automated tools

van Solingen and Berghout

## **GOAL/QUESTION/METRIC Approach**

---

### **Data Collection**

Data Collection Start up

Trial period to test forms

Kick-off Session

Checking forms for correctness and completeness

Storing form data

Building a measurement support system

spreadsheets, statistical tools, database applications,

presentation tools

Aggregate the data in analysis sheets

raw data, e.g., data, fault id

processed data, e.g., date, # of faults

graphs and tables (offers the characterization data)

van Solingen and Berghout

## GOAL/QUESTION/METRIC Approach

---

### Interpretation

#### Feedback Session

Prepare feedback session

slides and handouts on answers to the questions and goals

Organize and hold session

run by a project team member

evaluate action points from earlier sessions

interpret measurement data

draw conclusions

take notes

Report Measurement Results (Feedback session report)

van Solingen and Berghout

## References

---

- V. Basili, *Software Quality Assurance and Measurement: A Worldwide perspective, Applying the Goal/question/Metric Paradigm in the Experience factory*, Chapter 2, pp 21- 44, International Thomson Computer Press, ITP An International Thomson Publishing Company, 1995
- V. Basili and D. Weiss, A Methodology for Collecting Valid Software Engineering Data, IEEE Transactions on Software Engineering, pp. 728-738, November 1984.
- D. Weiss and V. Basili, Evaluating Software Development by Analysis of Changes: The Data from the Software Engineering Laboratory, IEEE Transactions on Software Engineering, pp. 157-168. February 1985.
- V. Basili and H.D. Rombach, The TAME Project: Towards Improvement-Oriented Software Environments, IEEE Transactions on Software Eng., vol. 14, #6, June 1988.
- V. Basili, G. Caldiera, and H.D. Rombach, Goal Question Metric Paradigm, Encyclopedia of Software Engineering - 2 Volume Set, pp. 528-532, Copyright by John Wiley & Sons, Inc., 1994.
- R. van Solingen and E. Berghout, The Goal/Question/Metric Method, McGraw Hill, 1999