Establishing a Measurement Program

Victor R. Basili

University of Maryland
and
Fraunhofer Center - Maryland

Copyright University of Maryland 2006
Outline

– Measurement Overview

– Generating Business Goals

– Matching Business Goals to Measurement Goals

– Goal Oriented Measurement

– Guidelines for Establishing a Measurement Program

– Discussion
Measurement Overview
Importance of Measurement

Create a corporate memory - baselines/models of current practices  
e.g., how much will a project cost, where am I spending my money?

Plan, track and control project development and evolution  
e.g., what should happen, is it happening?

Determine strengths and weaknesses of the current processes  
e.g., are certain types of errors commonplace?

Develop a rationale for adopting/refining decisions  
e.g., what techniques will minimize the problems, change the baselines?

Assess the impact of decisions  
e.g., has an approach reduced cost and schedule, and improved quality,  
what is the return on investment for a new process?

Evaluate the quality of the process/product  
e.g., what is the reliability of the product before delivery, after delivery?
Measurement

What are the levels of sophistication of goals?
Measurement Capability Maturity

Characterize
Describe and differentiate processes and products
Build descriptive models and baselines

Understand
Explain associations/dependencies between processes and products
Discover causal relationships
Analyze models

Evaluate
Assess achievement of quality goals, impact of technology on products
Compare models

Predict
Estimate expected product quality and process resource consumption
Build predictive models

Motivate/Improve
Describe what we need to do to control and manage software
Build prescriptive models
Software Measurement

What can we measure?

**Resource Data:**
- Effort by activity, phase, type of personnel
- Computer time
- Calendar time

**Change/Defect Data:**
- Changes and defects by various classification schemes

**Process Data:**
- Process definition and conformance
- Domain understanding

**Product Data:**
- Product characteristics
  - logical, e.g., application domain, function
  - physical, e.g., size, structure
- Usage and context information, e.g., design method used
Generating Business Goals
## Business Goals to Measurement Goals through strategies

| Business Goals | Goals applicable to an organization or part of an organization that is concerned with the business or mission of that organization; also called organizational goals, strategic goals  
*Example: Improve customer satisfaction* |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategies for achieving goals</td>
<td>Write a set of strategies for each goal</td>
</tr>
<tr>
<td>Assumptions</td>
<td>Decision factors that tends to be unclear and that might change (could be wrong)</td>
</tr>
</tbody>
</table>
## Relationships among Goals

### Hierarchy
- Top goal, easily articulated
- Sub-goals for organization, inherited by the divisions, inherited by project and individuals

### Complementary goals
- Support the current goal

### Competing goals
- Conflict with the current goal

### Indifferent goals
- Don’t affect the current goal

### (Timeframe)
- Long-term vs. short-term, point in time vs. period, dynamic vs. static, permanent vs. temporary
Essential Business Goal Types

**Growth goals:** acquire new projects with current core competencies areas; expand existing projects; evolve new competencies

**Success goals:** deliver good products to customers; control costs; shrink schedule; Increase profits; getting corporate visibility [awards etc.], building core competency)

**Maintain principle (internal) goals:** (transparency, employee satisfaction, controlled risk, learning environment)

\[ \rightarrow \text{measure to assure no decrease} \]

**Specific focus goals:** (make helpdesk more efficient, predict if proposal effort has a good ROI)
Important Elements of Goals

- **What**: activity, focus, object, magnitude, scope, timeframe
- **How**: strategies (coming from decision template)
- **Limitations**: constraints (relations with other options), relations with other goals
### Business Goal Template

<table>
<thead>
<tr>
<th>Activity</th>
<th>Reduce, increase, achieve, pursue, provide, …</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>cost, profit, turnover, market share, prestige,</td>
</tr>
<tr>
<td></td>
<td>customer satisfaction,</td>
</tr>
<tr>
<td>Object</td>
<td>people, market, a project, collection of projects,</td>
</tr>
<tr>
<td></td>
<td>customer, services for the population</td>
</tr>
<tr>
<td>Magnitude (degree)</td>
<td>x%, 1000K, y% more than last year,</td>
</tr>
<tr>
<td>Timeframe</td>
<td>3 years, 1. January 2008, permanently, from … to …,</td>
</tr>
<tr>
<td>Scope (who, context)</td>
<td>whole organization, business unit A, a person,…</td>
</tr>
<tr>
<td>Constraints</td>
<td>limited influence on certain factors, laws, mission</td>
</tr>
<tr>
<td></td>
<td>statement &amp; basic principles, …</td>
</tr>
<tr>
<td>Relation with other goals</td>
<td>other business goals, tradeoffs, hierarchy, ordering,</td>
</tr>
<tr>
<td></td>
<td>…</td>
</tr>
</tbody>
</table>
## Example 1: Success Goal: Customer Satisfaction

<table>
<thead>
<tr>
<th>Activity</th>
<th>Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>Customer Satisfaction with respect to Product Quality</td>
</tr>
<tr>
<td>Object</td>
<td>Customer Satisfaction Index</td>
</tr>
<tr>
<td>Magnitude (degree)</td>
<td>by 10%</td>
</tr>
<tr>
<td>Timeframe</td>
<td>Per year for the next 5 years</td>
</tr>
<tr>
<td>Scope (who, context)</td>
<td>(5% by division A, 15% by division B)</td>
</tr>
</tbody>
</table>

**Constraints (limitations)**

- basic principles, infrastructure (e.g., staff size, training), investment

**Relations with other goals**

- Can conflict with development cost goals, schedule goals, …
<table>
<thead>
<tr>
<th>Activity</th>
<th>Reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>Time to Delivery</td>
</tr>
<tr>
<td>Object</td>
<td>Calendar Time</td>
</tr>
<tr>
<td>Magnitude (degree)</td>
<td>by 20%</td>
</tr>
<tr>
<td>Timeframe</td>
<td>next three years</td>
</tr>
<tr>
<td>Scope (who, context)</td>
<td>(10% by division A, 30% by division B)</td>
</tr>
</tbody>
</table>

**Constraints (limitations)**

- basic principles, infrastructure (e.g., staff size, training), investment

**Relations with other goals**

- Can conflict with development cost goals, quality goals, …
Mapping Business Goals on Strategies and Measurement Goals

- **success goal**
  - category mapping
  - **customer satisfaction**
  - Return customers
  - recognition

- **business goals**
- **Lower level goals**
- **measurement goals**

- **strategies**

- **decision space templates**
- **scenario templates**
An example strategy

Business Goals

one strategy

Strategy: Improve Reliability

Lower level goal:
Improve System Test
An example strategy

Low level Goals

Scenario activities

One scenario (informal description)

Identify Impact factors

Select Impact Factors

Change Single Factor

Analyze effects

measurement goal:
characterize, analyze

GQM goal templates

scenario template
**Business Goals to Software Goals to Measurement Goals**

### Business Goals
- **Increase customer satisfaction**
- **Stay market leader**

### Strategy Decisions
- Improve Estimates
- Improve software Development Process
- Improve Timeliness
- Save costs through reuse
- Improve Reliability
- Provide Added Functionality to Product

### Software Goals
- Improve effort prediction accuracy [0.99; 1.04]
- Improve System Test [68%; 82%]
- Maintain customer satisfaction index [4.5; 6]

### Scenarios
- **Experts available = True**
  - A1: Identify impact factors (expert-based)
  - B1: Select impact factors (expert-based)
  - C: Determine baseline
  - D: Change single factor
  - E: Analyze effects

- **Data available = True**
  - A2: Identify impact factors (data-based)
  - B2: Select impact factors (data-based)
  - C: Determine baseline
  - D: Change single factor
  - E: Analyze effects

- **Data available = True and Experts available = True**
  - A3: Identify impact factors (hybrid)
  - B3: Select impact factors (hybrid)
  - C: Determine baseline
  - D: Change single factor
  - E: Analyze effects

### Measurement Goals
- M1: Characterize
- M2: Analyze
- M3: Characterize
- M4: Analyze
- M5: Characterize
- M6: Analyze
- M7: Characterize
- M8: Monitor
- M9: Evaluate
Understanding Business Goals to Software Goal Alignment

**Higher-level Business Goal 1:**
- Grow the Company
  **Decisions:**
  - Increase Market Share
  - Expand existing projects
  - Acquire new business (projects)
  - Reduce turnover
  - Go to Market Faster
  - Enter New Markets
  - Evolve new competencies

**Higher-level Business Goal 2:**
- Grow Successes
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Higher-level Business Goal 3:**
- Improve Profits
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Business Goal 1:**
- Increase Quality
  **Decisions:**
  - Reduce Defects
  - Increase review effectiveness
  - Increase testing effectiveness (probably additional effort)
  - Increase fault tolerance
  - Increase recoverability
  **Scenarios:**
  - QIP-based improvement of related processes
  - Identify influence factors
  - Build & apply model
  - Analyze & improve

**Business Goal 2:**
- Reduce Time
  **Decisions:**
  - Reduce functionality
  - Reduce quality
  - Increase productivity
  - Increase planning & control
  - Increase estimation accuracy
  - Reduce rework
  **Scenarios:**
  - QIP-based improvement of effort estimation & productivity-related processes

**Business Goal 3:**
- Increase Customer Satisfaction
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Software Goal 1:**
- Increase Productivity
  +20% in 2 years
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Software Goal 2:**
- Increase Quality
  **Decisions:**
  - Reduce Defects
  - Increase review effectiveness
  - Increase testing effectiveness (probably additional effort)
  - Increase fault tolerance
  - Increase recoverability
  **Scenarios:**
  - QIP-based improvement of related processes
  - Identify influence factors
  - Build & apply model
  - Analyze & improve

**Measurement Goal 1:**
- Review Defects

**Measurement Goal 2:**
- Productivity/Effort

**Higher-level Business Goal 1:**
- Grow the Company
  **Decisions:**
  - Increase Market Share
  - Expand existing projects
  - Acquire new business (projects)
  - Reduce turnover
  - Go to Market Faster
  - Enter New Markets
  - Evolve new competencies

**Higher-level Business Goal 2:**
- Grow Successes
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Higher-level Business Goal 3:**
- Improve Profits
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Business Goal 1:**
- Increase Quality
  **Decisions:**
  - Reduce Defects
  - Increase review effectiveness
  - Increase testing effectiveness (probably additional effort)
  - Increase fault tolerance
  - Increase recoverability
  **Scenarios:**
  - QIP-based improvement of related processes
  - Identify influence factors
  - Build & apply model
  - Analyze & improve

**Business Goal 2:**
- Reduce Time
  **Decisions:**
  - Reduce functionality
  - Reduce quality
  - Increase productivity
  - Increase planning & control
  - Increase estimation accuracy
  - Reduce rework
  **Scenarios:**
  - QIP-based improvement of effort estimation & productivity-related processes

**Business Goal 3:**
- Increase Customer Satisfaction
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Software Goal 1:**
- Increase Productivity
  +20% in 2 years
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Software Goal 2:**
- Increase Quality
  **Decisions:**
  - Reduce Defects
  - Increase review effectiveness
  - Increase testing effectiveness (probably additional effort)
  - Increase fault tolerance
  - Increase recoverability
  **Scenarios:**
  - QIP-based improvement of related processes
  - Identify influence factors
  - Build & apply model
  - Analyze & improve

**Measurement Goal 1:**
- Review Defects

**Measurement Goal 2:**
- Productivity/Effort

**Higher-level Business Goal 1:**
- Grow the Company
  **Decisions:**
  - Increase Market Share
  - Expand existing projects
  - Acquire new business (projects)
  - Reduce turnover
  - Go to Market Faster
  - Enter New Markets
  - Evolve new competencies

**Higher-level Business Goal 2:**
- Grow Successes
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Higher-level Business Goal 3:**
- Improve Profits
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Business Goal 1:**
- Increase Quality
  **Decisions:**
  - Reduce Defects
  - Increase review effectiveness
  - Increase testing effectiveness (probably additional effort)
  - Increase fault tolerance
  - Increase recoverability
  **Scenarios:**
  - QIP-based improvement of related processes
  - Identify influence factors
  - Build & apply model
  - Analyze & improve

**Business Goal 2:**
- Reduce Time
  **Decisions:**
  - Reduce functionality
  - Reduce quality
  - Increase productivity
  - Increase planning & control
  - Increase estimation accuracy
  - Reduce rework
  **Scenarios:**
  - QIP-based improvement of effort estimation & productivity-related processes

**Business Goal 3:**
- Increase Customer Satisfaction
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Software Goal 1:**
- Increase Productivity
  +20% in 2 years
  **Decisions:**
  - Deliver good products
  - Control costs
  - Shrink schedule
  - Increase profits
  - Get Corporate awards

**Software Goal 2:**
- Increase Quality
  **Decisions:**
  - Reduce Defects
  - Increase review effectiveness
  - Increase testing effectiveness (probably additional effort)
  - Increase fault tolerance
  - Increase recoverability
  **Scenarios:**
  - QIP-based improvement of related processes
  - Identify influence factors
  - Build & apply model
  - Analyze & improve

**Measurement Goal 1:**
- Review Defects

**Measurement Goal 2:**
- Productivity/Effort

**Analyzing Business Goals to Software Goal Alignment**

- Analyze Object
- Purpose
- Focus
- Viewpoint
- Context
Matching Business Goals to Software Goals
Software Measurement Needs

What is needed to support and sustain the activity?

Where is goal and data definition and analysis support needed?
  - Definition of corporate goals
  - Mapping of corporate goals onto software goals
  - Mapping of software goals onto measurement goals
  - Mapping of goals onto models
  - Mapping of models onto existing data
  - Interpretation of data based upon goals
  - Presentation of data to various stakeholders

How is the measurement process embedded in the organization?
  - Organization Structure
    - Integrate many projects into a single measurement framework
e.g., Experience Factory, SEPG/Measurement group.
  - Data Collection
    - By project aggregate up to central group
    - By central group
Problems Establishing a Software Measurement Program

Defining the right goals
- Tying corporate goals to software goals
- Inheriting software goals from corporate goals
- Identifying the context and temporal aspects of goal definition and achievement

Collecting the right data
- The tension between individual project needs and corporate needs with respect to measures taken
- Maximizing benefits while minimizing costs of data collection and analysis
- Taking maximum advantage of existing data

Defining and Sustaining the measurement process
- Creating the right organizational structure
- Getting feedback to projects in a timely fashion
- Maintaining commitment within all organizational levels
Attacking the Problems

Building an Effective Software Measurement Program is difficult

It requires support for

- Defining and integrating an organization’s top level corporate goals with its software goals
- Mapping the software goals onto data, maximizing the use of existing data where possible
- Evolving the goals and data collected as the organization matures
- Storing and retrieving goals, data, and interpretations in context from an experience base

It involves decision making, observation, experience, facilitation, collaboration, analysis and synthesis about goals, contexts, and assumptions

Most organizations fall short of putting together a successful program
Defining The Right Goals
Goal Derivation Concepts

Goals
- Needs of a particular stakeholder set for prescribing information
- Business Goals – goals the organization wishes to accomplish in general,
- Software Goals – goals related to the software process or product directly
- Measurement Goals – goals that can be made operational

Strategies
- A hierarchical set of possible approaches for achieving the goal

Scenarios Templates
- A set of steps that achieve the selected strategy

Assumptions
- Estimated unknowns that can affect the interpretation of the data

Context Factors
- Environment variables that change the kind of models and data that can be used

Scenario Instances
- Scenario templates defined for specific assumptions and context variables
Building Measurement Scenarios

Select the right goals:
- Identify the relationship between corporate and software goals, the scenarios, assumptions, context factors, …

Business goal: Reduce product time to market
- Assumption: Software is on the critical path to product delivery - thus shortening software development time is a reasonable approach

Related Software Strategies: Reduce software functionality, shorten cycle time of software activities, trade-off software characteristics for schedule,…

Context Factor: Is this a one time/project goal or a long term corporate goal, meant to be sustained? This affects the resources to be allocated, …

Select Software Sub-strategy: perform activities more efficiently, overlap activities, identify new time saving activities, …

Selected Software Scenario: Perform activities more efficiently: what is the current calendar time of activities, which are using up the most schedule (where are the biggest opportunities for improvement), …
Building Measurement Scenarios: Example 1

Choose the right *scenarios instances*:
Select the right measurement goals based upon what can you assume about the environment’s maturity with respect to measurement

**Perform activities more efficiently**: what is the current calendar time of activities, which are using up the most schedule (where are the biggest opportunities for improvement), ...

**Context Factor A**: Baseline data exists at the activity level
**Assumption**: The selected set of projects that form the baseline is relevant to the current situation

**Scenario Instance 1**: 
1. Build a schedule baseline by activity,
2. Identify activities that use a major calendar time,
3. Identify opportunities for improvement
4. Apply (Test) out the opportunities for improvement and record the effect on schedule
Building Measurement Scenarios: Example 2

Choose the right scenarios:
   Select the right measurement goals based upon what can you assume about the environment’s maturity with respect to measurement

Perform activities more efficiently: what is the current calendar time of activities, which are using up the most schedule (where are the biggest opportunities for improvement), ...

Context Factor B: No data exists

Scenario Instance 2:
1. Propose explicit hypotheses about baselines, problems, and opportunities for improvement based upon available expertise
   Assumption: The guesses at the baselines are reasonable and will be updated with real baselines when available
2. Apply (Test) out the opportunities for improvement and record the effect on schedule
Building Measurement Scenarios

The results from the previous steps provide the information needed for measurement goals (GQM structure)

Analyze representative projects in order to characterize them with respect to time & effort for each activity from the point of view of the Corporation

Analyze baseline data in order to evaluate them with respect to schedule shrinkage opportunities from the point of view of the Corporation

Analyze pilot project in order to evaluate it with respect to the effect of a schedule shrinkage activity from the point of view of the Corporation

Select the right models, metrics, data given the data available
What data exists? What is the basis for normalizing? Can the data be mapped onto the goals being generated
NASA Metrics Selection & Analysis Project

NASA Program Goal

“Advance software engineering practices to effectively meet the scientific and technological objectives of NASA”

Key objective - establish an agency-wide metrics program (for HQ, Centers, and Projects)

Measurement Challenges and Opportunities

Build experience base for hierarchy of project goals, aggregated to headquarters
Develop link for project goals to headquarters goals
Develop scenarios that work effectively in different project contexts

Recognize the variation in project contexts and provide consistent goals using different models and data, e.g., variation in CMM levels among projects

Make intelligent aggregations
Provide guidance and support in deploying the methodology across the projects
Develop analysis technique for grouping projects into common “types” based on project characteristics
Decision tree “Execute the project within budget, “low maturity context”

NASA GOAL:

NASA's Strategic Enterprises and their Centers “to deliver products and services to our customers more effectively and efficiently”

PROJECT GOALS:

Execute the project within budget
Execute the project within schedule
Reduce Rework
Train People etc.

Context Factors:

High maturity organization
Low maturity organization

Metrics to use:

Time
Amount billed
SLOC
Deliverable Status

Updated projections?
Productivity?
Decision tree “Execute the project within budget, “High maturity context”:

**NASA GOAL:**

NASA's Strategic Enterprises and their Centers to deliver products and services to our customers more effectively and efficiently

**PROJECT GOALS:**

- Execute the project within budget
- Execute the project within schedule
- Reduce Rework
- Train People etc.

**Context Factors:**

- High maturity organization
- Low maturity organization

**Metrics to use:**

- Planned Budget
- Actual Budget
- Planned % of activity completeness
- Actual % of activity completeness
- Updated projections?
Summary of Key Components for building a software measurement program

An **experience base** of goals and scenarios that allow for the measurement program to be tailored to specific context variables and assumptions and is based upon experiences with various organizations.

A **method** that takes into account the need for:
- a goal hierarchy that allows goal choices for the needs of a particular organization and stakeholders,
- dependency of goals on one another, i.e., temporal relationships,
- scenarios for identifying clusters, recognizing which types of clusters are needed depending upon environmental constraints,
- mapping goals into existing data sets to maximize information while minimizing data collection,
- the inheritance of data across multiple goals, i.e., mapping the data required from one set of goals onto others.

An **expert** to help set up the measurement program in a the particular organization, including the generation of the goals, measures, data, and analysis.
The Goal Oriented Measurement
Software Measurement

Who are the stakeholders?

There are a variety of stakeholders at multiple levels
e.g., Manager, Customer, User, Organization, Developer

What does each want to know?
Determines what we measure
But these points of view need to be integrated and linked and interpreted for each viewpoint based on common data

How are the appropriate metrics determined?
There are measurement methods to support metric definition and interpretation
e.g., Goal/Question/Metric Paradigm (GQM), Practical Software Measurement (PSM), Balanced Scorecard (BSC), ...
Measurement Infrastructure

**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 must** be driven by goals and models

for the processes and products, projects, organization
Measurement Infrastructure

- Internal and external stakeholders have their own goals
- Well defined goals enable business success
Measurement Infrastructure
Goal Based Measurement

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

Goal/Question/Metric Approach

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 stakeholder 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 Approach
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
   (stakeholder: user, customer, manager, developer, corporation,...)
   in the following context
   (problem factors, people factors, resource factors, process factors,...)
Goal/Question/Metric Approach
Relating goals to Metrics

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, ...
Goal/Question/Metric Approach
Relating goals to Metrics

• **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, ...

Sources of Software Errors

% of Errors

Type of Error
Goal/Question/Metric Approach
Relating goals to Metrics

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 Approach
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 Approach
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 Approach
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 $F_c = \text{the ratio of faults found in system test to the faults found after system test on this project.}$

Let $F_s = \text{the ratio of faults found in system test to the faults found after system test in the set of projects used as a basis for comparison.}$

Let $Q_F = F_c/F_s = \text{the relationship of system test on this project to faults as compared to the average the appropriate basis set.}$
Goal/Question/Metric Approach
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
Guidelines for Building a Measurement Program
Establishing A Measurement Program
Guidelines from the SEL

The most important rule is to
Understand that software measurement is a means to an end, not an end in itself

Three key reasons for Measurement

Understanding the Business
Baseline models and relationships
Key characteristics

Managing Software Projects
Planning and estimating
Tracking actuals versus estimates
Validating models

Guiding Improvement
Understanding
Assessing
Packaging
Establishing A Measurement Program
Guidelines from the SEL

Understanding the Business

The most important reason for measurement is to understand your business

How much are we spending on software development?
Where do we allocate and use resources throughout the life cycle?
How much effort do we expend specifically on testing software?
What types of errors and changes are typical on our projects?
How long will it take me to finish testing this software?
Is reliability a function of testing time?
Should I impose stronger testing standards?
...
So we need to build baseline models and relationships as a basis for all forms of understanding
Establishing A Measurement Program
Guidelines from the SEL

Managing Software Projects

Planning and estimating
  Build models of relationships for key variables

Tracking actuals versus estimates
  Track your progress in real time and compare to your baselines

Validating models
  Learn how and when your models are changing so you can modify them

Focus on applying results rather than collecting data
ESTABLISHING A MEASUREMENT PROGRAM
Guidelines from the SEL

Guiding Process Improvement

The three basic steps are:
- Understanding
- Assessing
- Packaging

Understanding and characterizing helps you understand where you are.

Assessing involves learning what works and what doesn’t.

Packaging involves making what you have learned a part of your business.
ESTABLISHING A MEASUREMENT PROGRAM
Guidelines from the SEL

Key Issue for Setting Up a Program

Understand the goals
prioritize

Understand how to apply measurement
multiple customers for the results

Set expectations for change
measurement introduces change

Plan to achieve an early success
show the investment is worth while

Focus locally
gain should be to local organization

Start small
let the scope evolve based upon success
ESTABLISHING A MEASUREMENT PROGRAM
Guidelines from the SEL

Key Issue for Setting Up a Program

Organize the analysts separately from the developer
their goals and processes are different

Make sure the measures apply to the goals
don’t collect data for data’s sake

Keep the number of measures to a minimum
there is a real cost associated with measurement

Avoid over-reporting measurement data
make the results as crisp and clear as possible

Budget for the cost of the measurement program
include all costs in planning and tailor it to fit the goals and budget

Plan to spend at least 3X as much on data analysis and use as on data collection
the real payoff is in the analysis and use
ESTABLISHING A MEASUREMENT PROGRAM
Guidelines from the SEL

Costs in a Mature Program

The cost of **data collection** should not add more than 1 to 2 percent to the software development or maintenance budget

- includes completing forms, participating in interviews, attending training sessions and helping characterize project development

The **data processing** element of the measurement program may cost 3 percent of the total development budget

- includes collecting, archiving, validating, and maintaining the measurement data

The cost of the **analysis** element of the measurement program may cost 5 percent of the total project budget

- includes design of studies, information analysis, project interaction, packaging
ESTABLISHING A MEASUREMENT PROGRAM
Guidelines from the SEL

Experience-Based Guidelines

Data collection should not be the dominant element of process improvement; application of measures is the goal

The focus of a measurement program must be self improvement, not external comparison

Measurement data are fallible, inconsistent, and incomplete

The capability to quantify a process or product with measurement data is limited by the abilities of the analysis

Personnel treat measurement as an annoyance, not a significant threat

Automation of measurement has limits
Questions
Back-Up Slides
Defining the measurements to evaluate the new test method
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

How many requirements are there? (enumerate them)

What is the importance of 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
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

What is the distribution of tests over requirements?
(number of tests/requirement)

Is the number of tests/requirement consistent with its complexity and importance?
0 - there are no tests for this requirement
1 - there are several tests but not nearly enough
2 - # of tests reasonable but insufficient for the importance/complexity
3 - # tests are sufficient for the complexity and importance
Process Goal Example
Object of Test 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
Object of Test **Domain Understanding**
Quality of the Test Cases

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**

**Data Sources: System test tables**

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

<table>
<thead>
<tr>
<th>Req. #</th>
<th>How understandable is requirement?</th>
<th>How important?</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Req.</th>
<th># of tests</th>
<th>How well are tests known?</th>
<th>Difficulty of testing</th>
<th>Evaluation subj. stat</th>
<th>Are # of tests consistent with diff. &amp; import?</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>5</td>
<td></td>
<td>3</td>
<td>3</td>
<td>3/50</td>
<td>4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Test #</th>
<th>Failure?</th>
<th>How confident are you in the results?</th>
<th># of Faults found</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>x</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Number of Requirements</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Subjective rating of how understandable the requirement is</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>