Technical Report
CMU/SEI-93-TR-16
ESC-TR-93-193

Establishing a Software Measurement Process

Donald R. McAndrews

July 1993

Technical Report
CMU/SEI-93-TR-16
ESC-TR-93-193
July 1993

Establishing a Software Measurement Process

Donald R. McAndrews
Software Process Measurement Project

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/AXS
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

(signature on file)

Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

This document is available through SAIC/ASSET: 1350 Earl L. Core Road; PO Box 3305; Morgantown, West Virginia 26505 / Phone: (304) 284-9000 / FAX: (304) 284-9001 / World Wide Web: http://www.asset.com/SEI.html / e-mail: sei@asset.com.

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center / Attn: BRR / 8725 John J. Kingman Road / Suite 0944 / Ft. Belvoir, VA 22060-6218. Phone: (703) 767-8274 or toll-free in the U.S. — 1-800 225-3842).

Table of Contents

List of Figures

Acknowledgments

This report presents methods founded on successful experiences and lessons learned as well as methods documented throughout industry. Special thanks are owed to Anita D. Carleton and James Rozum, of the SEI Software Process Measurement Project, for their ideas, original thoughts, and experiences.

This technical report was completed as part of a working relationship between the SEI Software Process Measurement Project and the software engineering process group (SEPG) at the Air Force's Standard Systems Center (SSC) at Maxwell Air Force Base, Gunter Annex, Alabama. I am especially thankful to Colonel John Barnhart who sponsored our activities at SSC, and also to the members of the SSC SEPG (XPEP branch): Mr. Ron Obranovich, Captain Gloria Trabue, Captain Ken Anthonis, Captain Al Braaten, Captain Mark Minkler, Major Don McCanless, and Mr. Dub Jones. In addition, I would like to thank Gene Miluk, SEI's account manager at SSC, who actively participated in this work and whose valuable experiences contributed to this report.

This report could not have been assembled without the active participation and many contributions from other members of the SEI Software Process Measurement Project: John Baumert (Computer Sciences Corporation), Mary Busby (IBM Corporation), William Florac (SEI), Wolfhart Goethert (SEI), Mark McWhinney (IBM Corporation), Robert Park (SEI), Daniel Paulish (Siemens Corporation), and Patricia B. Van Verth (Canisius College).

In addition, I would like to thank the following members of the Software Engineering Institute who reviewed and contributed to this report: James Hart, Nancy Mead, Timothy Olson, and David Zubrow.

I also want to thank Suzanne Couturiaux of SEI Information Management group who reviewed early versions of this report as well as provided the final edits. And finally, I want to thank Kathy Cauley for her administrative support.

Establishing a Software Measurement Process

Abstract. This report presents guidelines for establishing a measurement process as part of an organization's overall software process. Methods are suggested that can be used to design a repeatable measurement process that is focused on goal setting, data analysis, and decision making rather than on just data collection and numbers. Examples are included to illustrate these methods in the context of some common software process management issues involving basic measures of size, effort, and schedule. This report also suggests some steps for starting a measurement program. These steps focus on identifying an organizational strategy for improvement and designing a process for measurement to support this strategy.

1. Introduction

The primary purpose of measurement is to provide insight into software processes and products so that an organization is better able to make decisions and manage the achievement of goals. This report proposes some guidelines that can help organizations integrate a measurement process with their overall software process.

1.1. Purpose/Objectives

In software organizations, measurement is often equated with collecting and reporting data and focuses on presenting the numbers. The primary purpose of this report is to focus measurement more on setting goals, analyzing data with respect to software issues, and using the data to make decisions.

The objectives of this report are to:

- Provide some guidelines that can be used to design and implement a process for measurement that:
  - ties measurement to organizational goals and objectives;
  - defines measurement consistently, clearly, and accurately;
  - collects and analyzes data to measure progress towards goals; and
  - evolves and improves as the process matures.
- Demonstrate the guidelines described with examples and illustrations.
- Outline some steps to help an organization start and sustain a measurement program.

1.2. Audience

This report is intended to be used by a software engineering process group (SEPG) or a working group tasked to implement a measurement program in their organization. But, because measurement is used to help quantify the software process for making decisions, virtually anyone involved with software can benefit from the use of measurement as described in this report. Some additional examples of functional areas where people will want to use good measurement practice include:

- managers involved in making decisions for planning and controlling software projects,
- project staff focused on improving their work,
- software configuration management groups focused on product integrity,
- software quality assurance groups focused on process assurance,
- customers focused on the end use of software products, and
- other functional areas involved with making decisions regarding software products and/or processes.

1.3. Overview

This report is divided into four chapters. This introductory material comprises Chapter 1.

Chapter 2 proposes methods for designing and implementing a measurement process. It describes an architecture consisting of five activities. Each activity consists of tasks to help a project and/or organization develop, plan, implement, and improve a measurement process that can evolve as measurement needs change.

Chapter 3 describes uses of measurement. It illustrates some applications of the methods described in Chapter 2.

Chapter 4 suggests steps an organization could take to establish a software measurement program. Designing the process as described in Chapter 2 is only one of the steps involved in starting or improving a measurement program. Chapter 4 includes steps to help ensure the organization has a strategy in place and recognizes the need to design a measurement process to support that strategy.

2. Designing a Software Measurement Process

This chapter describes an architecture for designing a software measurement process. The architecture is pictured in Figure 1. The activities required to design a measurement process using this architecture will be described in the following four sections:

- Section 2.1 - *Developing a measurement process* to be made available as part of the organization's standard software process;
- Section 2.2 - *Planning the process* on projects and documenting procedures by tailoring and adapting the process asset;
- Section 2.3 - *Implementing the process* on projects by executing the plans and procedures; and
- Section 2.4 - *Improving the process* by evolving plans and procedures as the projects mature and their measurement needs change.

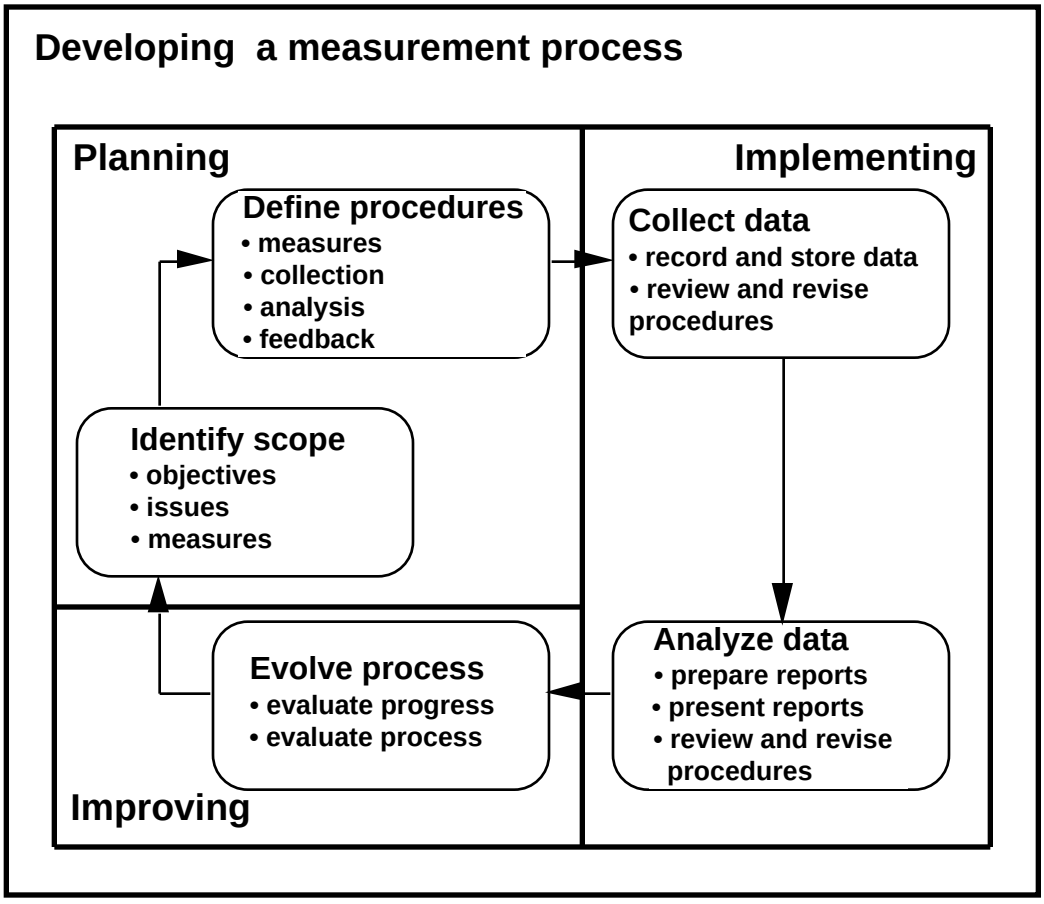


Figure 1. Software Measurement Process Architecture

The architecture is generic in that it can be instantiated at different levels, e.g., project level, divisional level, or organizational level. Chapter 3 illustrates some examples of these instantiations.

The measurement process described in this report is based on the goal-question-metric (GQM) approach pioneered by Basili and Rombach at the University of Maryland [Basili 84]. This approach links the measurement activities to the quantifying of products, processes, and resources to make decisions to meet project goals. Similar approaches are described in [ami 92], [IEEE 89], [Paulish 92], and others. The key principle shared by all is that projects must assess their environments so that they can link measurements with business objectives. Then projects can identify suitable measures and define measurement procedures that address these business objectives. Once the measurement procedures are implemented, the process can continuously evolve and improve as the projects and organizations mature. The architecture in Figure 1 addresses these concepts and principles.

The process represented in Figure 1 is analogous to the Shewhart/Deming Cycle which pertains to the fundamental tasks in quality control [Shewhart 86][Deming 86]. The Shewhart/Deming cycle has evolved and is currently described as the plan-do-check-act (PDCA) cycle. The software measurement process relates to this cycle as shown in Figure 2:

Shewhart/ Deming Cycle	Activity in the Software Measurement Process	Purpose
• Plan	• Identify scope/ Define procedures	• Establish a purpose, plan, and procedure for measurement tied to organizational goals
• Do	• Collect data	• Execute the plan to collect data
• Check	• Analyze data	• Analyze data to determine progress towards measurement goals
• Act	• Evolve process	• Make decisions based on the progress towards goals and recommend improvements on the measurement process being used

Figure 2. Relationship of the Measurement Process to PDCA

By aligning the measurement process with the overall software process as shown in Figure 2, projects and organizations are able to simultaneously collect and analyze data to help make decisions with respect to project goals, and obtain feedback to improve the measurement process itself. A working definition for a software measurement process is given here for reference.

Software measurement process – That portion of the software process that provides for the identification, definition, collection, and analysis of measures that are used to understand, evaluate, predict, or control software processes or products.

2.1. Developing a Measurement Process

Figure 1 illustrated a process architecture that can be used to develop an organizational process for measurement. This measurement process becomes a process asset to be made available for use by the projects in developing, maintaining, and implementing the organization's standard software process [Paulk 93]. Some examples of process assets related to measurement include:

- the organization's standard software measurement process element (described below);
- organizational databases and associated user documentation;
- cost models and associated user documentation;
- tools and methods for defining measures (e.g., the reports referenced in [Park 92], [Goethert 92], [Florac 92]); and
- guidelines and criteria for tailoring the software measurement process element.

The asset that this report focuses on is the software measurement process element. Process elements are constituent parts of the overall software process, e.g., software estimating, software design, code, unit test, peer reviews, and measurement. Each process element covers a well-defined, bounded, closely related set of tasks [Paulk 93]. This report suggests some methods to design measurement process elements. Organizations can use this report to:

- develop measurement templates to be filled in, provide abstractions to be refined, or provide complete measurement descriptions to be used or modified;
- describe the software measurement process architecture and how the elements of measurement fit in with the organization's standard software process;
- identify how software measurement relates to other elements of the software process, e.g.,:
  - various life-cycle models;
  - the ordering, interfaces, and interdependencies within the software process; and
  - interfaces and dependencies outside the software process (e.g., systems engineering, hardware engineering, contract management).

The remaining sections of this chapter suggest methods and templates that can be used to help design a software measurement process element as a constituent part of the organization's overall software process, based on the architecture pictured in Figure 1. The architecture consists of five activities that correspond with planning, implementing, and improving the software measurement process. These activities are described in Sections 2.2

through 2.4 using entry-task-validation-exit (ETVX) diagrams [Radice 85] to describe the entry conditions, tasks to be performed, validation of the tasks, and the exit conditions (see Figure 3). Included in Section 2.2 are task-by-task descriptions of the activities, and templates that can be helpful in planning and documenting the process, including examples. Sections 2.3 and 2.4 use ETVX diagrams and discuss some of the common issues associated with implementing and improving measurement plans and procedures. Figure 3 illustrates the format of an ETVX diagram. Note that some information, such as roles and responsibilities, should be added to the templates in Sections 2.2-2.4, as these are unique to the organization implementing the process.

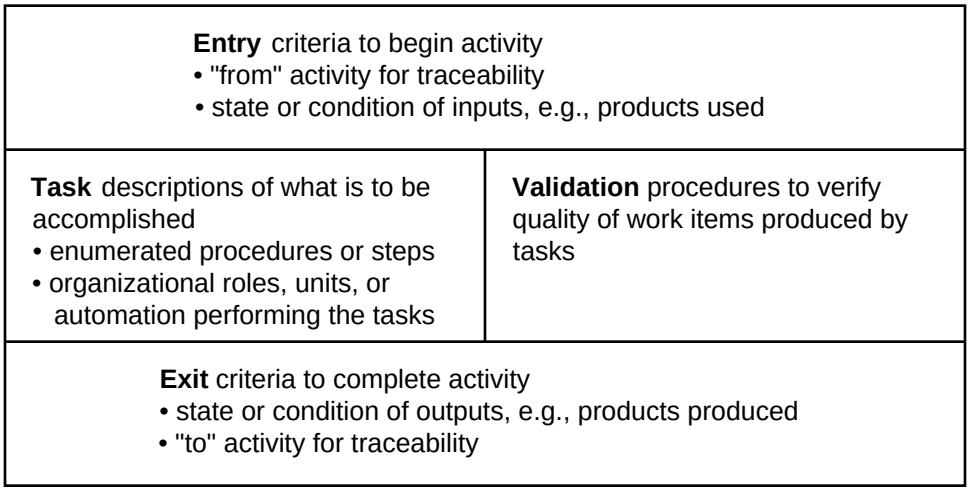


Figure 3. ETVX Diagram

The purpose of using a process definition method such as ETVX is to develop an operational definition of the measurement process. These operational definitions can then be adapted or tailored for use on all projects. To be operational, the documented process must put communicable meaning into the concepts and express the measurement process in operational terms to:

- identify what data are to be collected,
- define how the data are to be collected and reported,
- define how the data are to be used to make decisions,
- define how the process is to evolve and improve,
- collect and analyze the data,
- make decisions, and
- start over by continuing and/or adjusting the process.

To be operational, the process must achieve a result nearest to what is needed for a particular end [Deming 86]. With measurement, projects collect only those measures that are needed to provide enough data to make decisions. Because data collection can be expensive, the measurement process must be feasible with respect to the time and costs associated with collecting data.

Documentation of this process may vary considerably from project to project. It may result in plans or procedures that organizations present in their own ways. Regardless of how the process is documented (i.e., whether as a department instruction, a standard practice, or a formal procedure), the basic tasks described in the following sections should be included or at least addressed.

2.2. Planning the Process

Planning the measurement process involves the first two activities of the architecture, as shown in Figure 4.

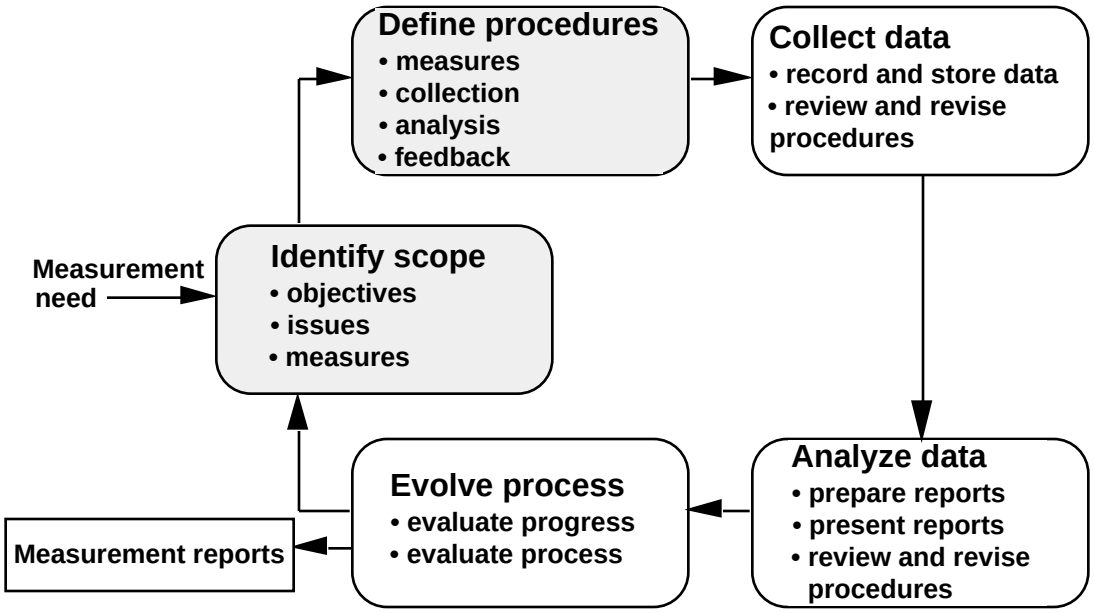


Figure 4. Measurement Architecture - Planning

2.2.1. Identify Scope

The measurement process originates with a need for measurement. The need could be large or small. For example: an organization may need to measure corporate-wide project performance; a small project may need to assess and evaluate their design and code inspections; or SEPGs may need to focus and evaluate process improvement activities. The need for measurement typically will involve different audience perspectives; that is, various functional groups will use or give inputs to the measurement process. Before designing a measurement process, it is important to understand what the needs for measurement are, and who the audiences are.

Once the need for measurement has been established, the first activity of the measurement process that must be addressed is the Identify Scope Activity. This activity helps to establish the purpose for measurement. The Identify Scope Activity includes identification of

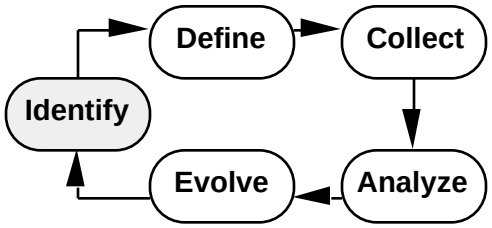
objectives that measurement is to support, issues involved to meet these objectives, and measures that provide insight into these issues.

The tasks shown in Figure 5 determine the scope and purpose of the measurement effort. Detailed descriptions of each task are included following Figure 5. In some situations, some of these tasks may overlap. In others, they need not be emphasized. The organization should tailor its tasks to suit its needs, and clearly align measurement to support its goals and objectives.

During the Identify Scope Activity, measurement is tied to the goals and objectives of the organization. Only measures that support decision making with respect to goals and objectives are identified. Without this activity, projects frequently collect data that are not meaningful for making decisions within their overall process.

Entry:

Audience needs insight into software processes and/or products to make decisions and manage the achievement of objectives. This may be a new need, or the continuation or modification of an existing need for measurement.



Tasks:	**Validation:**
Identify:	
1. measurement need	• audiences and needs are identified
2. objectives	• objectives are relevant to organization
3. methods	• objectives are feasible
4. measurement issues	• issues describe the products, processes, and resources attributable to the methods
5. measurement goals	• measurement goals are quantifiable and unambiguous
6. questions	• answers to questions give insight into achievement of goals/issues
7. measures	• measures answer and are traceable to questions

Exit:

A set of measures or things to be measured that provide insight into the software processes and/or products for making decisions. These measures are approved, and the process continues with the Define Procedures Activity.

Figure 5. ETVX for Identify Scope Activity

In Figure 5, the sequence of tasks to complete the Identify Scope Activity are:

1. *Identify the needs for measurement.* A measurement need could be as broad as providing insight into the achievement of the organizational mission; or it could be as focused as providing insight into a change in project performance due to the implementation of a new software tool. For any measurement need, it is important to recognize that there may be several audiences or users of the measurement data, each with different perspectives and needs for measurement (e.g., SEPGs, project management, quality assurance). All audiences or users should be identified, including those who will receive the measurement reports, those who will make decisions based on the data, and those who will provide the data.
2. *Identify the objectives of the organization that each measurement need is going to address.* This task ensures that measurement is aligned with and will support decision making with respect to the business objectives of the organization.
3. *Identify the methods that will be used to achieve the objectives stated in task 2 above.* This task will help to ensure that there are methods to achieve the objectives, and that the objectives are feasible. Also, these are the methods that need to be measured in order to give the measurement users insight to make decisions.
4. *Identify the issues that need to be managed, controlled, or observed.* These issues should refer to products, processes, and/or resources that are traceable, and relate to the methods identified in task 3. Some common issues are size, cost, and schedule.
5. *Translate each measurement issue into precise, quantifiable, and unambiguous measurement goals.* These goals may involve measurement to help individuals understand an issue (i.e., to evaluate, predict, or monitor) or to address a need for improvement (i.e., to increase, reduce, achieve, or stabilize). [ami 92] provides several examples of measurement goals.

   As a template, a goal could identify the purpose of the measurement in the following format [Basili 87]:

   > To [characterize, evaluate, predict, monitor, motivate, increase, reduce, achieve, stabilize] the [process, product, model, measure] in order to [plan, control, improve, understand, assess, manage, engineer, learn] it.
   >
   > For example: To evaluate the system testing methodology in order to improve it.
6. *Identify a list of questions for each goal that, when answered, will determine if the goal is being achieved.* These questions could involve products, processes, improvements, etc. Answers to the questions should indicate the status of progress toward goals, which will in turn provide insight into the issues.
7. *For each question, identify things to be measured which could determine the answers to the questions.*

This sequence of tasks produces a set of measures to address the need for measurement. However, as the software process evolves, so will the measurement process that is supporting it, and additional measures may be identified. Each iteration or cycle through the measurement process may re-identify the scope.

These tasks can be used as a template for documenting the Identify Scope Activity of the measurement process. Figure 6 provides an example that illustrates this activity. [ami 92] provides several additional examples of how measures are identified to support the decision-making process.

Task 1. Measurement need:

Project managers need to have insight into project progress

Task 2. Organizational objective(s):

Produce products on time, within budget

Task 3. Methods:

Below are some examples of methods for achieving the objective:

- Document a standard process for estimating effort and schedule and use a commercial cost model to improve the estimating and tracking process.
- Use prior project performance as a check of project feasibility, use realistic estimates, perform periodic reviews of actual versus plan, replan when necessary, etc.

Task 4. Measurement issues: (Actual versus estimated size, effort, and schedule)

- What do you do when costs and schedule overrun the plan?
- How do deviations in size affect schedule?

Task 5. Measurement goal:

To monitor actual versus estimated size, effort, and schedule in order to make decisions with respect to project plans, progress, and need for replanning

Task 6. Questions:

What was the planned size, effort, and schedule for the project?

How much have we done?

How much is left to do?

When will it be complete?

Task 7. Things to be measured:

- size
- effort
- schedule

Figure 6. Example of Identify Scope Activity

The Identify Scope Activity identifies questions and measures that indicate progress towards the organization's objectives. The scope of the measurement effort is limited to measures necessary for making decisions with respect to the objectives, i.e., useful data. The process continues on to the Define Procedures Activity with a set of measures.

2.2.2. Define Procedures

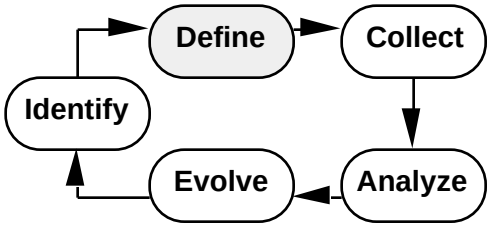
The complete set of identified measures is obtained from the Identify Scope Activity. Once measures have been identified, operational definitions and procedures of the measurement process should be constructed and documented. The tasks for the Define Procedures Activity include:

- define each of the identified measures completely, consistently, and operationally;
- define the collection methods, including how the data are to be recorded and stored; and
- define the analysis techniques to be used to determine status relative to the identified measurement goals. These analysis techniques will most likely evolve and expand through each reporting cycle as the user gains more insight into the data.

The procedures documented during this activity should be reviewed and revised if necessary during each reporting cycle. Figure 7 summarizes the tasks to define procedures in the ETVX format. These tasks are then detailed, along with examples to illustrate the tasks included in Figure 8.

Entry:

From the Identify Scope Activity, a set of measures or things to be measured is obtained. These measures have been approved, and provide insight into the software processes and/or products for making decisions.



Tasks:	**Validation:**
Define:	
1. measures	• measures are described in clear, unambiguous, and operational terms
2. counting procedures	• counting procedures are repeatable; they include source of data, frequency of collection, special tools identified, and verification procedures
3. recording	• forms are developed for consistent recording of data
4. storing	• databases exist to reliably store historical data
5. analyzing	• the methods for analyzing the data answer the goal related questions
6. reporting	• tables, charts, and graphs adequately summarize progress towards goals
7. feedback	• mechanisms exist to incorporate feedback for updating operational definitions

Exit:

An approved procedure for measurement that is operationally defined. To be operational, the documented procedures must put communicable meaning into the concepts, and express the measurement process in operational terms of:

- identify what data are to be collected,
- define how the data are to be collected and reported,
- define how the data are to be used to make decisions,
- define how the process is to evolve and improve, and
- identify who will collect, enter, store, maintain, verify, and have access to the data.

Figure 7. ETVX for Define Procedures Activity

To define the measurement activity as indicated in Figure 7, document the following sequence of tasks:

1. *Define the measures.* [Florac 92], [Goethert 92], and [Park 92] provide advice for a checklist approach to defining measures related to size, effort, schedule, and problem reporting. These definitions follow structured rules that state what is included and excluded in each measure. The checklists describe the measurement data so that receivers of the information will not be misled by unstated assumptions or local variations in measurement practices [Carleton 92]. Defining measures could also involve the design of questionnaires, interviewing techniques, etc. This type of information should be documented and kept as part of the plans and procedures for the measurement process. [ami 92] also provides templates and advice for defining measures.
2. *Define counting procedures.* Once the measures are defined in operational terms, procedures for collecting each measure should be documented. The procedures should be clear and repeatable, and include the source of the data, collection responsibilities, frequency of collection, and any special tools required. There should also be some vehicle for verifying that the data are correct, i.e., that the procedure can be duplicated to get the same result.
3. *Construct forms or procedures to record the measures.* The forms should include administrative data such as the project's name, date, configuration management data, data collector, etc. It is helpful to work with project staff and let them take part in designing and buying into the format.
4. *Select and design a database or spreadsheet to store the information.* The use of a spreadsheet or database will help to organize the data and make it easier to use. As the database grows, the collected data can help indicate trends and long-term performance of the measured projects.
5. *Define analysis methods and show how the measures can answer the goal-related questions.* After some study and evaluation, preliminary assumptions can be made about key process events and their relationships, and basic analysis methods become apparent. As projects collect data and gain knowledge about the process, these analysis techniques will be refined as additional issues will become part of the process. Some initial considerations should include:
   - Identification of any potential or anticipated interactions between various measures. Analytical methods can involve the correlation of related measures using a scatter diagram to plot one measure versus another.
   - Identification of a basic set of warning signals that should arise when the collected data are significantly different from the plan or estimate. This could involve trend analysis of actual versus estimated measures and/or control charts indicating thresholds and warning limits.
   - Illustration or examples of conclusions that can be made from the data report with respect to the goals.

6. *Use the traceability between the issues/goals and the measures to define potential relationships among the measures and construct measurement reporting formats.* These formats should include charts and graphs to present the measures and answer questions about each issue/goal.
7. *Define mechanisms for feedback that:*
    - show how the reports are to be distributed to the different audiences, including a distribution list;
    - identify the needs for any regular meetings to present measurement reports, including typical agendas, attendees, purposes, outcomes, etc.;
    - specify how reports can be used and affect decisions; and
    - define the process for receiving feedback from the audiences, e.g., marked-up hard copies or formal change requests.

These tasks are illustrated in Figure 8. Because defining the measurement process will be different for each organization, Figure 8 includes some examples and general comments to illustrate this activity. The examples are a continuation of the example which began in Figure 6 on page 11. This example is referred to in the remainder of this report to illustrate how an organization or project can implement and evolve a measurement process.

Task 1. Define measures

size: Size represents the magnitude of software being developed, to be used as an indication of the amount of work being done and the amount of resources needed to do the work [Rozum 92]. One useful measure for size is lines of code. The figure below illustrates a partial checklist for defining source lines of code. Refer to [Park 92] for more details on defining line of code measures. Other useful measures for size are number of requirements, computer software modules, and function points.

effort: Effort can be measured in staff hours, i.e., counts of staff hours planned and expended on a project. Staff hour counts can be partitioned by labor and support staff categories, experience levels, functional areas (e.g., quality assurance, configuration management, testing), or by life-cycle activity (e.g., requirements analysis, code and unit test). Refer to [Goethert 92] for more details on defining staff-hour measures.

schedule: Schedule is shown as a graphical portrayal of delivery dates, activities, and milestone progress. [Goethert 92] includes a checklist for defining schedule events that can be used to clarify these schedule measures.

Partial Definition Checklist for Source Statement Counts [Park 92]

Definition name: ***Physical Source Lines of Code (basic definition)*** Date: ***8/7/92***

Originator: ***SEI***

Measurement unit: **Physical source lines** [4] **Logical source statements** [ ]

Statement type	**Definition** [4] **Data array** [ ]		**Includes**	**Excludes**
	When a line or statement contains more than one type, classify it as the type with the highest precedence.			
1	Executable	**Order of precedence ->** 1	4	
2	Nonexecutable			
3	Declarations	2	4	
4	Compiler directives	3	4	
5	Comments			
6	On their own lines	4		4
7	On lines with source code	5		4
8	Banners and nonblank spacers	6		4
9	Blank (empty) comments	7		4
10	Blank lines	8		4
11				
12				

How produced	**Definition** [4] **Data array** [ ]	**Includes**	**Excludes**
1	Programmed	4	
2	Generated with source code generators	4	
3	Converted with automated translators	4	
4	Copied or reused without change	4	
5	Modified	4	
6	Removed		4
7			
8			

Figure 8. Example of Define Procedures Activity (1 of 5)

Task 2. Define counting methods

Planned (estimated) data for size, effort, and schedule may be taken from contract documentation (e.g., statement of work, proposals, software development plans).

Actual data may be collected and reported periodically (e.g., monthly) using collection tools to facilitate timeliness, accuracy, and efficiency of the process. Procedures for counting the measures will vary significantly among software environments. Some general comments and/or suggestions on counting methods for the measures in this example are:

- Organizations may already have adopted an automated, standard line of code counter to measure *size*. Consider using the checklist in [Park 92] to clearly document how the existing tool counts lines of code. Tools can be developed to access the source listings in the configuration management library, count the lines of code per source file, and aggregate the totals for the project, each product, or however the data are to be tracked. For example, if plans were developed at the CSCI level, then the data should be collected, reported, and analyzed at that level.
- *Effort* data are typically obtained from the accounting department through the existing cost accounting system. If not, staff-hour data can be obtained from employee timecards. Consider using the checklist in [Goethert 92] to help document existing measures for effort being used.
- *Schedule* data could be obtained from the project status reports, with the software supervisors updating the activity status at least monthly. Objective entry and exit criteria must be defined for each activity and agreed upon in order to track actual versus plan. [Goethert 92] discusses this aspect of defining milestones and schedule measures. One way to automate this collection is to enter activity plan and actual data in the developer's software development files, on line. Then tools can be developed to search the software development files for planned and actual activity status.

Task 3. Define the recording formats

Data should be recorded in a format that includes: name of project, date collected, configuration management reference, and data collector. In this example, the recording format should include actual size, effort, and scheduled activity progress, and any revised estimates for size, effort, or scheduled activities.

Figure 8. Example of Define Procedures Activity (2 of 5)

Task 4. Define storage mechanisms

The use of a database tool or spreadsheet such as Excel or Lotus 123 is suggested. The tool used should have import/export capabilities with the related tools and/or databases being used for collection. For example, the automated line of code counter can export data directly into the spreadsheet.

Task 5. Define analysis methods [adapted from Rozum 92]

Based on the scope and purpose of measurement in this example, the measures should provide insight into progress of the software project, and give early warning of cost overruns and/or the need to replan.

Look for actual size, effort, and schedule to closely track planned values. Deviations could signify problems.

Variations in size, effort, or schedule could mean:

- developers may be sacrificing effort and/or schedule to produce the product;
- developers may be sacrificing functionality to meet effort and schedule;
- estimating process may have been inadequate; and
- original requirements may be incomplete, inaccurate, or not understood.

In either case, the original plans may no longer be realistic, and the project requires re-planning.

However, by correlating measures, other process problems could be apparent. For example, if there are deviations in size, but not effort and schedule, it could mean several things, for example:

- The project may be spending its time closing open problems, and not developing new code.
- Or, if actual size is much larger than planned, but effort and schedule are on track, it may mean that the product is being developed too hastily, at the expense of quality.

In either case, the project manager will need to look at some measures of product quality in subsequent reporting cycles.

Examples of charts that can be used as indicators of these described conditions are included in Task 6.

Figure 8. Example of Define Procedures Activity (3 of 5)

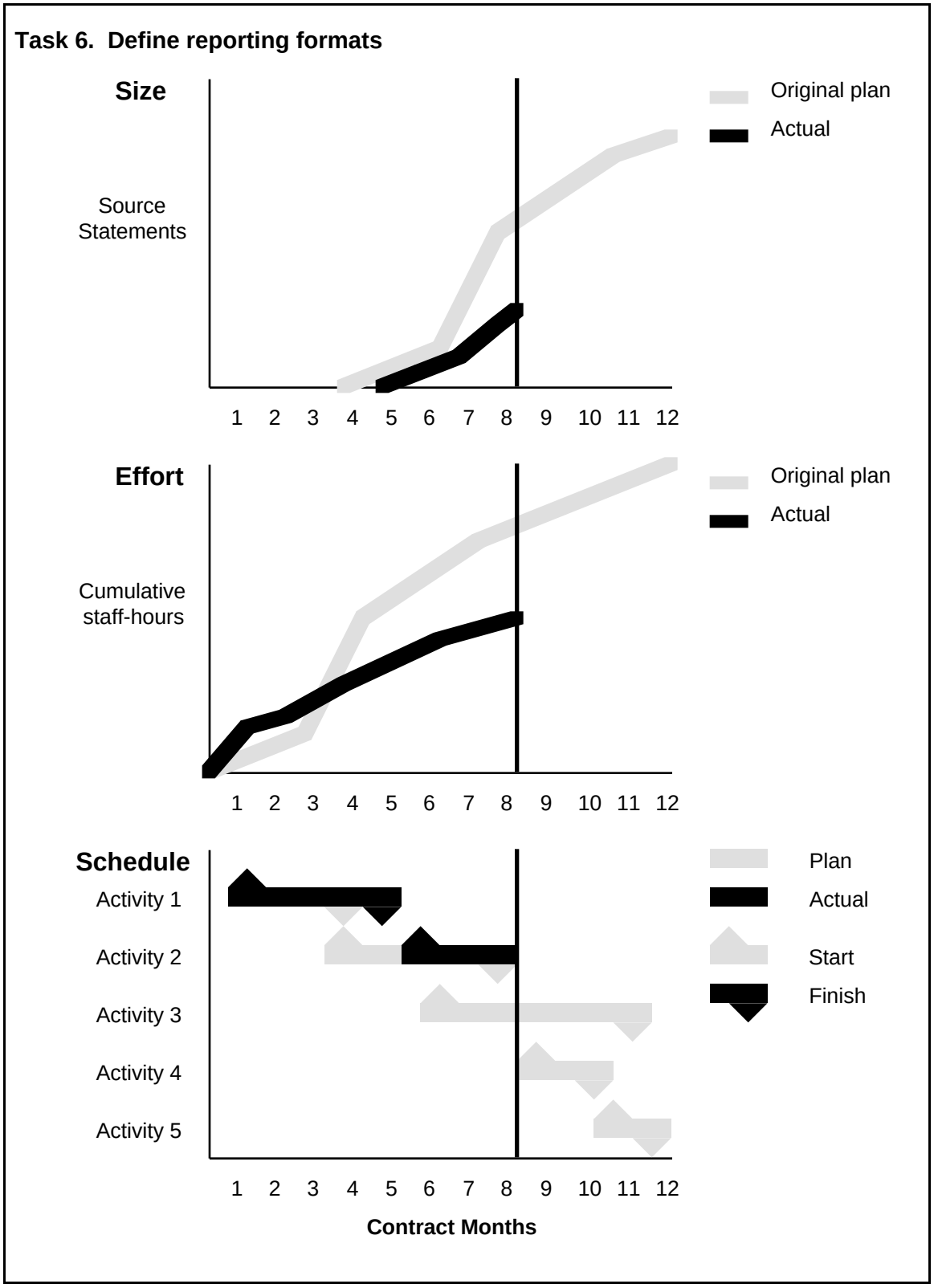


Figure 8. Example of Define Procedures Activity (4 of 5)

Task 7. Define mechanisms for feedback

Feedback mechanisms will be tailored to the organization. Some general comments pertaining to this example:

- Monthly project meetings could be held to review project status among the managers and software staff, and the functional areas (i.e., quality assurance, configuration management, test) that interface with the developers.
- Measurement reports should be distributed prior to review sessions so that the issues reflected in the reports can be examined for resolution or discussion at the meetings.
- Much feedback can be obtained from the scheduled review sessions. However, not all issues will be raised in a meeting. Red-lined or written comments after the meeting should also be accepted.
- The absence of feedback may be bad feedback. If reports are not generating feedback, it is likely that the reports are not being used.

Figure 8. Example of Define Procedures Activity (5 of 5)

2.2.3. Summary of Identify Scope and Define Procedures Activities

Execution of the Identify Scope and Define Procedures Activities should result in a documented, operational plan and procedures for measurement. The plan should identify the scope and purpose of the measurement effort, together with the roles, responsibilities, resources, and schedules for the measurement process. The procedures will be tailored and adapted to each project that implements the measurement plan. These documented plans and procedures will serve as a baseline that will evolve as necessary with each iteration through the process (i.e., each reporting cycle). Because it is not possible to know beforehand all of the analysis techniques or the information that will be necessary, the baseline measurement procedures will evolve as data are collected and further insight is obtained from successive cycles through the process. As each reporting cycle provides additional feedback, the baseline measurement procedure should be reexamined and updated accordingly. With a measurement plan in hand, the organization is ready to collect data.

2.3. Implementing the Process

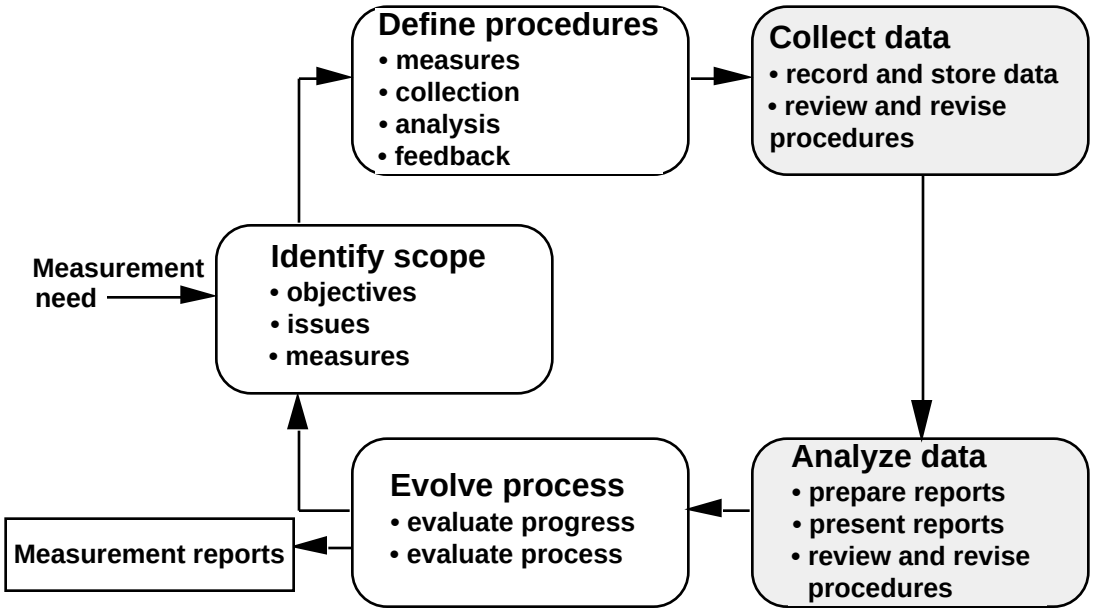


Figure 9. Measurement Architecture - Implementing

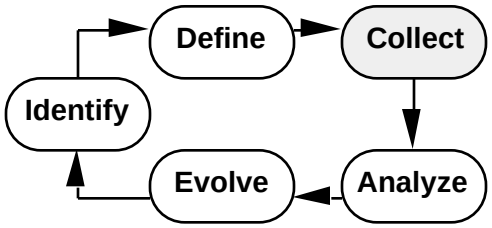
Figure 9 illustrates the activities that are the focus of implementing the measurement process on actual projects. The first two activities of the measurement process identified the scope and purpose for measurement and documented repeatable procedures to operationally define the measurement process. The next two activities implement the documented measurement procedures. These activities are described using ETVX diagrams as in Section 2.2. Based on the execution of these activities, the documented procedures for measurement should be reviewed and revised to meet the evolving needs of the projects or organization. Evolving the process is discussed in Section 2.4.

2.3.1. Collect Data

This activity implements the collection procedures defined in the Define Procedures Activity. Using the operational definition of the measurement procedures, data are collected, recorded, and stored. In addition, the data are validated and the procedures are reviewed for adequacy. Procedures may need to be tailored or adapted to suit each project that implements the measurement procedure. Figure 10 outlines the entry and exit criteria, tasks, and validation required for this activity.

Entry:

Approved, operational definition of the measurement procedures from the Define Procedures Activity



Tasks:	**Validation:**
1. collect data	
2. record data	
3. store data	• data are verified (list specific criteria, e.g., data are timely, consistent, repeatable)
4. review and revise procedures	• collection procedures are adequate (list specific criteria, e.g., adaptable to project environment)

Exit:

Approved, verified data are provided as input to the Analyze Data Activity

Revisions to the Collect Data Activity procedures are provided as input to the Evolve Process Activity

Figure 10. ETVX for Collect Data Activity

Using the example charts shown in Figure 8 on page 19, some general comments could be made regarding problems with the data collection procedures:

- Are those individuals who are collecting data implementing the procedures as defined?
- Are definitions of procedures valid for a particular project?
- Are actual size data lagging the plan? This could occur because the automated line of code counter is not finding all of the source listings. Developers and/or the

configuration management staff should ensure that the procedures to count lines of code are adequate.

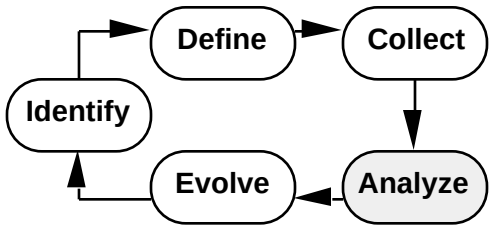
- Are actual size and effort data lagging the plan? It may be that not all effort data are being reported as planned. For example, are indirect support staff, and overtime included in the actual data? Were these figured into the plans? Here the definitions of the effort data must be reviewed to ensure consistency. Compare the definitions of planned size with the definition of actual size being implemented by the line of code counter to ensure that these definitions are consistent. [Goethert 92] discusses these issues.
- If status of activities is being obtained from the developer's software development files or from project status reports, the developers must ensure that the source of the data accurately reflects actual status. Here again, there may be a problem with the collection procedure.

After implementing the collection procedures, the process continues with the Analyze Data Activity.

2.3.2. Analyze Data

Entry:

Approved, verified data from the Collect Data Activity



Tasks:	**Validation:**
• analyze data	
• prepare report	• report provides insight into issues, status of goals obtained from report
• present report	• report is understood and acceptable (list criteria, e.g., approved by audience)
• review and revise procedures	• analysis procedures are adequate (list criteria, e.g., additional issues captured)

Exit:

Draft measurement reports distributed to the audience to the Evolve Process Activity

Revisions to the Analyze Data Activity procedures are provided as input to the Evolve Process Activity

Figure 11. ETVX for Analyze Data Activity

This activity consists of analyzing the data to prepare the reports, presenting the reports to the audience, and reviewing procedures for accuracy and adequacy. The procedures may need to be updated if the report is not providing insight into the issues, or the report is not understood. In either case, feedback is collected to update the measurement procedures to analyze data.

Referring to the example in Figure 8 on page 19, the overall trend is that actual size, effort, and schedule are lagging the plan. This could indicate that :

- there are not adequate staff working on the project;
- adequate computer resources (e.g., government furnished equipment, contractor furnished equipment) are not available;
- the requirements are not stable or not defined; or
- the project needs to get more data.

The project manager must investigate these issues to determine what the cause of the slips are. The project manager may find that the original plans are no longer realistic. If data are not already available, the measurement process may then be updated to include these additional issues and measures for the next reporting cycle.

Collect Data and Analyze Data Activities are intended to implement the current measurement plans and procedures. At the same time, these two activities collect feedback to update the plans and procedures for the next reporting cycle. Issues may come up that require the procedures to be updated. Section 2.4 more specifically addresses the evolution of the measurement process.

2.4. Evolving the Process

Figures 12 and 13 illustrate the final activity of the measurement process architecture. The Evolve Process Activity is used to improve the process and ensure that there is a structured way of incorporating issues and concerns into improvement of the process.

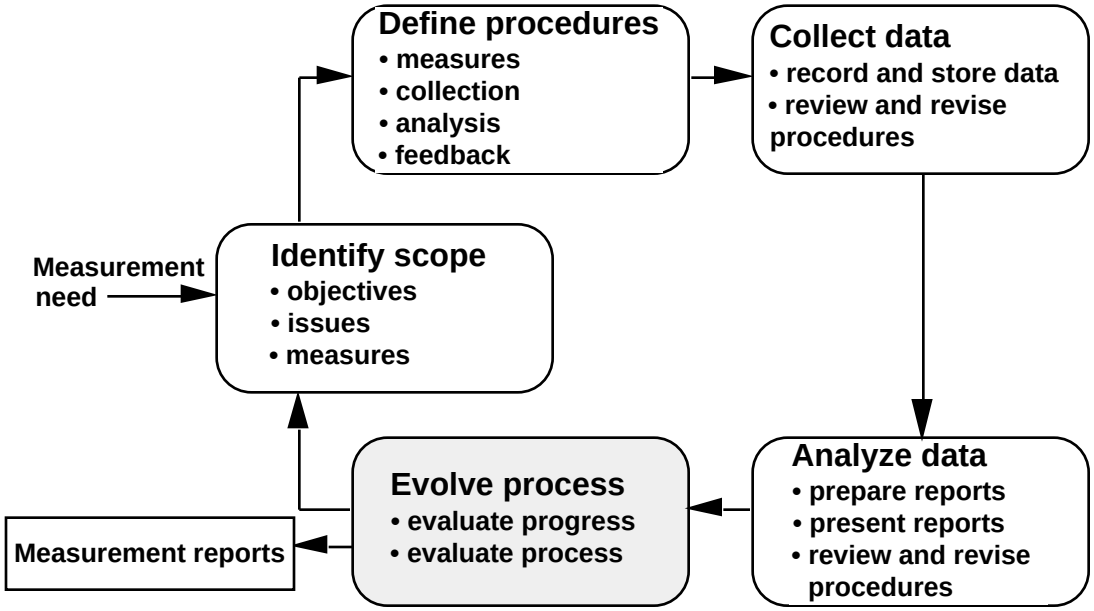


Figure 12. Measurement Architecture - Evolving

At this point, the audience receives the measurement reports, and makes decisions based on the data. These decisions may involve replanning, corrective action, or simply moving on without change.

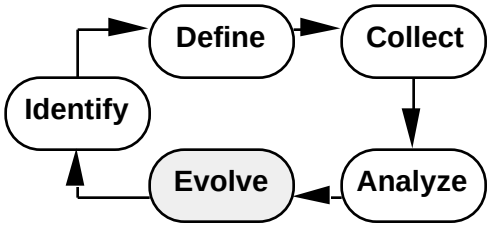
This activity also assesses the adequacy of the measurement process itself. As decisions are made, the audience provides feedback on the adequacy of the data available. They may raise additional issues, or indicate that certain data are no longer available. This feedback, in addition to the feedback from the other activities, is addressed during the next reporting cycle, by reidentifying and redefining the measurement process. This activity overlaps the entire process, and it culminates in the review of the current reporting cycle. Typically, the measurement focal group (discussed in Chapter 4) will collect any feedback or changes necessary to the measurement procedures and evaluate what kind of changes should be made for the next reporting cycle.

Entry:

Draft measurement reports distributed to the audience from the Analyze Data Activity

Revisions to the collection procedures from the Collect Data Activity

Revisions to the analyze data procedures from the Analyze Data Activity



Tasks:

- evaluate insight into progress
- evaluate measurement process

Validation:

- audience reviews and accepts measurement report
- inconsistencies or concerns to be addressed are identified

Exit:

Final measurement reports are reviewed and accepted by audience

Feedback is obtained to evolve the process to the Identify Scope Activity, including adjustments to:

- identified scope (objectives, issues, measures)
- defined procedures (measures, collection, analysis, feedback)

Figure 13. ETVX for Evolve Process Activity

Managers need to determine if they have adequate information to evaluate progress with respect to their goals (i.e., could they use the data to make decisions?). Also, they need to evaluate whether the procedures defined are adequate. The end result of this activity is:

- final measurement report is accepted by the audience for the current reporting period; and
- feedback is obtained to evolve the measurement plans and procedures.

In the example in Figure 8 (on pages 16-20), there were several inconsistencies or concerns raised. Any feedback such as the following issues should be addressed during the next reporting cycle:

- The defined measurement procedures may need to be updated as a result of reviewing the collection methods.
- The measurement plans may need to be updated to reflect additional issues raised during analysis.
- There may have been no feedback at all from the measurement report, indicating that the audience may not have needed the report.

3. Illustrations of Use

Chapter 2 outlined some methods for designing a measurement process which an organization can apply at various levels. This chapter illustrates some of the different levels or scenarios where the organization could apply the measurement process described in Chapter 2. The purpose of a measurement process is to provide insight into software processes and products so that an organization is better able to manage the achievement of goals. Figure 14 illustrates some of the uses for measurement.

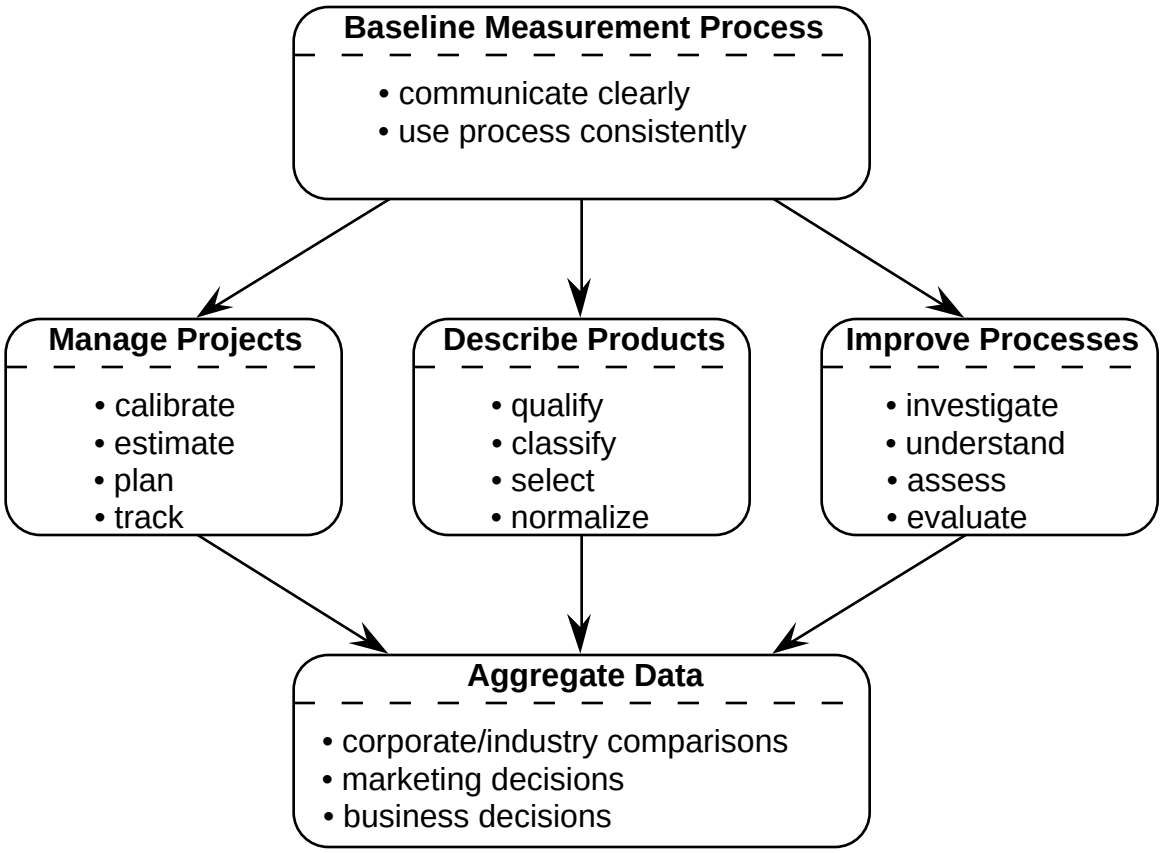


Figure 14. Uses of Measurement

3.1. Baseline Measurement Process

An effective measurement program starts with a baselined process that can be used consistently by all decision makers across the organization. Without this baseline, the use of measurement is ad hoc, limiting an organization's ability to manage projects, describe products, improve the process, and aggregate data across projects.

The initial instantiation of the methods in Chapter 2 could be to document a process that all projects can use to communicate consistently in terms of the size, effort, schedule, and problems. These are the basic measures described in [Carleton 92]. They address fundamental, important product and process characteristics central to planning, tracking, and improving. The methods in Chapter 2 can be used to design a process that includes:

- common management objectives and issues;
- basic things to be measured (e.g., size, effort, schedule, and problem counts);
- unambiguous definitions;
- collection procedures;
- analysis techniques;
- mechanisms to evolve (e.g., monthly progress reviews); and
- roles and responsibilities.

With a baseline measurement process defined, a solid foundation for collecting measures is established. As the basic measures evolve, their definitions and uses often expand. For example:

- problem reports can be expanded to track status, type, severity, and priority;
- size attributes can be tracked by language, platform, development status, origin, and production method;
- effort attributes can be added to track by labor class, phase, and activities performed; and
- schedule can be tracked by dates and completion criteria.

These basic measures can be built upon so that they become even more effective in managing projects, describing products, and improving processes. These uses of measurement are discussed below.

3.2. Manage Projects

With a repeatable measurement process, managers can begin to rely on the basic measures they use for managing their projects. Using the methods described in Chapter 2, a measurement process element can be defined to:

- collect basic measures on completed projects, so that a historical database can be established to help understand the capability of the current software process;
- use historical data to calibrate software estimating models;
- develop realistic estimates and plan the size, effort, and schedule for future projects based on historical data;
- use actual versus planned data to evaluate how much of the project is complete, how much is left to do, and when the project can be complete, in order to make decisions regarding resources and progress; and
- replan projects based on deviations in status, progress, or renegotiation of requirements.

3.3. Describe Products

Once the manager understands the basic measures to manage project performance and capability, the next level involves describing the products in terms of these basic measures. As an organization defines the processes to be used across all projects, it can begin to focus measurement on quality and product descriptions. Product descriptions are used to qualify how good a product is, classify product characteristics, and select products to meet user specifications. Some examples of product descriptions include maintainability, reliability, and problem densities. These are all ways of describing products in terms of the relationships between the basic measures (e.g., defect density = defects/size). The methods in Chapter 2 can be used to identify some of these product description issues and relationships that exist. Some examples of how different users might want to use the measurement process to describe products include:

- managers can understand the level of quality of their products and the quality that their existing process is capable of achieving;
- developers can use product descriptions to help them understand the quality of their work and identify potential strengths and weaknesses in the process; and
- customers can describe the products in their requirements specifications to indicate the desired level of quality.

3.4. Improve Processes

So far, the illustrations have involved describing the products and the existing capability of the software process. Another common objective of managers is to improve the capability of the organization [Humphrey 89]. Once managers understand the basic measures they use for managing their existing processes and products, they can begin to focus on process improvement. Here, measurement is used to investigate and understand, and to focus on problem areas or targets for improvement. For example, if projects are finding a large percent of their problems from system-level or field-level testing, the organization may need to implement a more rigorous design and code inspection process, or provide training on unit testing (i.e., focus improvement on earlier phases of the development process). Once process improvements are implemented, measurement helps to assess and evaluate the success of process improvement activities.

3.5. Aggregate Data

With measurement used consistently on projects, managers can aggregate data across projects. The methods from Chapter 2 can be used to help senior management identify and define measures that will help them make decisions with respect to organizational goals and objectives. With measurement they can better understand the software process and organizational capabilities, and get involved with the business aspects of software. This typically involves aggregating data from projects within the organization to assemble data on productivity (size/effort), quality (before and after release), and predictability [Grady 87].

Often, software organizations are required to report measures as part of a mandate or policy. This upwards communication of project status is an example of how important a consistent measurement process could be. In order for projects to report and reflect their current progress accurately, they should have documented plans and procedures that communicate exactly what the data represent. Without this consistency, the use of aggregations of data is questionable.

3.6. Dynamic Change

One final illustration involves the concept of dynamic change of the measurement process. This concept builds on the original need for measurement and an evolving process for measurement. As organizations or projects mature, objectives will be met or modified. With changes in the organization (and the process), the needs for insight into the process will also change. For example, if the organization establishes a formal software quality assurance group or software configuration management group, there immediately exist new audiences and users of measurement data. These groups need to become part of the measurement process. Also, as tools and technology are introduced into the organization, new insights into the process become necessary to determine the effectiveness of these tools. For example, if

the organization adopts a new cost estimating tool, it will need to identify and define the measures that are necessary for the cost estimating model, i.e., the parametric inputs that describe the software environment, application complexity, developer experience, etc.

3.7. Summary of Illustrations

Figure 14 focused on distinct uses for measurement. However, it is clear that these uses and the measures overlap, and that the uses are built on foundational, basic measures for size, effort, schedule, and problems. The definitions and uses of these measures expand as the measurement process matures. An organization benefits from measurement by gaining the insight necessary to:

- establish baseline performance parameters for estimating and project planning;
- communicate status and make decisions based on actual measurements versus plan;
- understand a process or product, its characteristics, and/or problem areas; and
- assess improvement activities to justify, manage, and evaluate quality and productivity improvement programs.

4. Starting a Software Measurement Program

Chapter 2 addressed several issues regarding the design of a software measurement process. However, besides the technical design of the software measurement process, there are also other organizational considerations and boundaries involved in implementing a new process or improving an existing process. Whether an organization is defining a software measurement process or a configuration management process, changes should be implemented in a structured, well thought-out manner.

This chapter provides some guidelines for integrating a measurement process into the overall software process of an organization. Figure 15 suggests a series of steps that can help an organization initiate or improve a measurement program. These steps are adapted from [Grady 87] and [QSM 92]. The degree of emphasis on individual steps depends on the maturity of the organization and on existing measurement capabilities. Each of these steps is discussed individually in the following sections. Note that the second step, *Design the Process*, is addressed by Chapter 2 of this report.

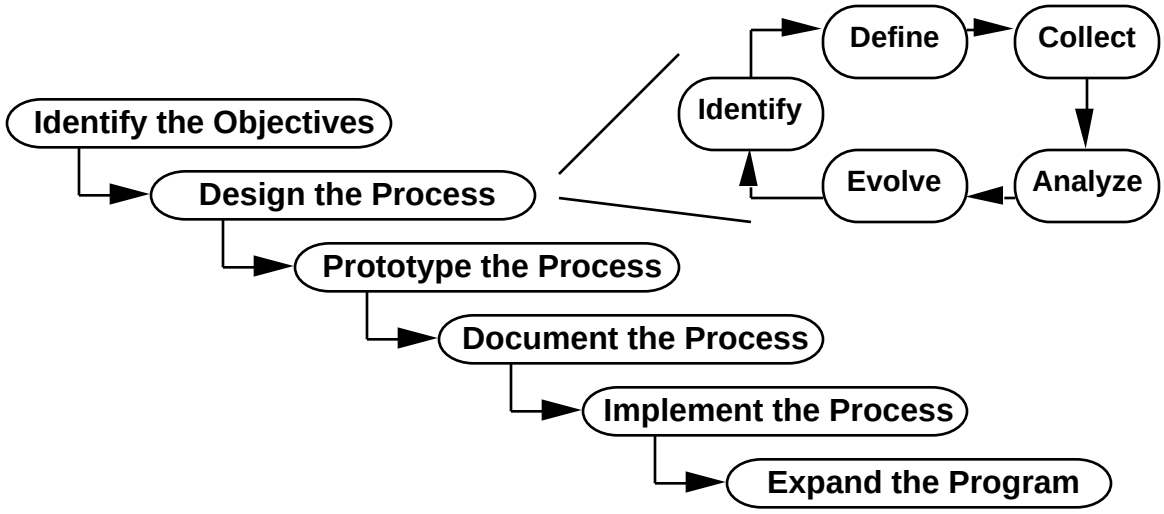


Figure 15. Starting a Software Measurement Program

4.1. Establish a Measurement Focal Group

Once an organization recognizes the need to start or improve its measurement program, it should identify and allocate resources for a measurement focal group. This focal group may be part of an existing SEPG, or a separate working group or process action team assigned to measurement. Some responsibilities for this focal group might include:

- assessing organizational sponsorship, commitment, and involvement;
- implementing the steps described in Sections 4.2 through 4.7;
- communicating and strengthening the network of sponsors;
- documenting successes and lessons learned;
- evolving the process based on successes and lessons learned; and
- establishing a historical, organizational database of project data.

As the role of measurement expands, more resources may become necessary. Often the focal group is augmented with project representatives [Dion 92]. For example, as the need for measurement grows, the focal group may perform measurement functions full time, while project representatives provide part-time support. This composition has several benefits. For example: additional project representatives help grow the group's capabilities; measurement obtains wider visibility among projects; and projects develop a sense of ownership of the measurement program. Projects are more likely to support measurement if they are part of the process and see the benefits of using measurement to make decisions.

4.2. Identify the Objectives

Any change or improvement in the software process, including the use of measurement, must start with clearly identified, realistic objectives. Management must set challenging objectives, monitor progress, and insist on performance [Humphrey 89]. These objectives may be the result of process assessment findings and recommendations or some other process improvement activity. The measurement focal group then works with management to translate high-level objectives into measurable goals for improvement. Ultimately, everyone must be involved in the measurement activity. Decision makers, including all levels of management, should be able to use measurement data to make decisions regarding the issues related to organizational objectives. This task will frame the methods, priorities, and management support, and will align the measurement activity with the overall goals of the organization.

4.3. Design the Process

Projects all too often begin their measurement efforts by collecting data. However, even though many projects may be collecting data, few use this data to make decisions regarding software products and processes. This happens because the projects are focused on collecting numbers rather than on working through a process to identify and define what measures they want to collect and analyze with respect to their goals. An organization should assess and understand its current measurement capability, and then design a process that uses its existing capability for leverage.

4.3.1. Assess and Understand Current Capability

Effective improvement of any software process requires an understanding of the current capability [Humphrey 89]. Therefore, the focal group needs to concentrate initially on assessing and understanding the organization's current software process capability and existing measurement and/or data collection methods. This involves surveying the projects, and identifying existing process(es) being used and existing data collection procedures. The focal group should determine how the projects are currently using and benefiting from measurement, and what the projects might like to see in a measurement program. By assessing and understanding the organization's existing measurement capability, the focal group is able to design a measurement process that uses existing capabilities for leverage.

4.3.2. Design a Measurement Process

A software measurement process needs to be designed and documented. Otherwise, it will merely be adjusted to each successive crisis, and the overall performance will remain unchanged [Humphrey 89]. Once the focal group understands the organization's existing capability and objectives, it can design a software measurement process to meet the organization's needs. There are several sources in the literature to use as a foundation for designing this process. The process in Chapter 2 uses an adaptation of Basili's goal/question/metric paradigm [Basili 84] and ties measurement to the organizational objectives. This process involves identifying and defining measures that directly relate to the project's or organization's goals for improvement. The process continuously evolves based on feedback from each collection and analysis cycle.

4.4. Prototype the Process

A measurement process should be tested on actual projects and adjusted before it is implemented throughout the organization [Humphrey 89]. The measurement focal group should work with each project to do this. The focal group helps projects understand what the organization needs, and the projects help with the implementation details.

As a result of prototyping, the focal group should develop an understanding of the:

- current project performance with respect to the organizational objectives,
- benefits and lessons learned from the existing measurement process, and
- scope of the effort and resources necessary to initiate and maintain the measurement process on projects.

4.5. Document the Process

The focal group should discuss its prototyping results with management, addressing the benefits and lessons learned and how measurements can support the organization. During this step, the focal group and management should review the goals and objectives that were stated during step 1 (see Section 4.2: Identify the Objectives). Management should examine the organizational objectives for the future and determine how they can use measurement to monitor the achievement of these objectives. Based on the prototype and the level of effort involved, management should decide how to proceed with the measurement activity.

The focal group can now formally document the measurement process as part of the organization's standard software process. This may involve a policy statement, a standard procedure, templates for defining measures, or some other form of documenting software processes. An organization-wide process helps clarify activities, roles, and responsibilities, and integrates measurement with related policies and procedures.

4.6. Implement the Process

Once measurement is formally documented within the organization, the focal group can begin to implement the measurement process across the organization. This involves working with projects to integrate the measurement process with their software processes.

A good way to work with projects consistently is to develop a measurement workshop. This workshop can be used for training projects with a need for measurement. It will transfer the basic measurement capabilities and train the projects on the use of organizational policies, standards, and procedures for measurement. The workshop can be a vehicle for setting up working relationships between the measurement focal group and projects.

4.7. Expand the Program

The focal group can now proceed to look for opportunities to integrate measurement with other process improvement activities. To expand the measurement effort, projects must perceive that there is a benefit to be obtained [QSM 92]. It is much easier to sell the measurement process based on the benefits to the project rather than by enforcing the policy as a requirement. Once useful results have been produced, projects will be more willing to support the work [Humphrey 89]. With plans for the measurement process in place, selling the process involves continuous implementation of the measurement program.

The measurement group should:

- publicize the successes, benefits, and organizational trends observed from projects already involved with the program;
- collect measures and build on the historical repository; and
- develop tools, standards, and procedures to make the process more usable and adaptable to project needs.

5. Summary

This report has presented some basic concepts for establishing software measurement as part of the overall software process in an organization. To be successful, the measurement program entails more than defining and establishing a process. It may require a shift in culture and be accompanied by a corresponding change in attitude. This kind of cultural shift will be slow and will rely heavily on building success through implementation of successful measurement procedures. Human-intensive processes, such as the software process, are constantly changing. The measurement focal group should continuously discuss and review all aspects of the measurement program with the projects and ensure that there are mechanisms for change built into the software measurement process [Rifkin 91].

Appendix A: Acronyms

CMM	capability maturity model
CMU	Carnegie Mellon University
GQM	goal-question-metric
ETVX	entry-task-validation-exit
IEEE	Institute of Electrical and Electronic Engineers
PDCA	plan-do-check-act
QSM	Quantitative Systems Management
SEI	Software Engineering Institute
SEPG	software engineering process group
SLOC	source lines of code

Appendix B: Terms Used

Activity – An organizational unit or stage that performs a specific function of a process (e.g., the identify scope activity of the software measurement process). Part of the measurement process architecture with a predefined: (1) list of entry criteria that should be satisfied before beginning the tasks, (2) set of task descriptions that indicate what is to be accomplished, (3) validation procedure to verify the quality of work items produced by the tasks, and (4) checklist of exit criteria that should be satisfied before the activity is viewed as complete (based on [Radice 85]).

Goals – Target set by an organization.

Goal/Question/Metric (GQM) – An architecture for stating goals and refining them into specific questions about the characteristics that need to be measured. These questions provide a specification for the data needed to help address the goal. This architecture was developed by V. Basili from the Software Engineering Laboratory at the University of Maryland when working in collaboration with the NASA Goddard Space Center. He has also applied it within various U.S. companies [Basili 84].

Baseline – A specification or product that has been formally reviewed and agreed upon, which thereafter serves as the basis for future development, and which can be changed only through formal change control procedures.

Data – Something collected, given, or admitted, especially as a basis for reasoning or inference; factual material used as a basis especially for discussion or decision; something used as a basis for calculating or measuring.

Measure – *n*. A standard or unit of measurement; the extent, dimensions, capacity, etc. of anything, especially as determined by a standard; an act or process of measuring; a result of measurement. *v*. To ascertain the quantity, mass, extent, or degree of something in terms of a standard unit or fixed amount, usually by means of an instrument or process; to compute the size of something from dimensional measurements; to estimate the extent, strength, worth, or character of something; to take measurements.

Measurement – The act or process of measuring something. Also a result, such as a figure expressing the extent or value that is obtained by measuring.

Measurement goals – Specific objectives or targets which relate to the implementation and execution of methods to achieve objectives, and are stated in quantifiable, unambiguous terms.

Measurement plan – A document that details how the activities of applying and using the measurement process are approached and applied in a particular working environment.

Method – Systematic procedures and techniques used to accomplish a task [Radice 85].

Objective – Concrete, measurable performance targets that specify achievements and support the overall mission of the organization.

Operational measurement process – To be operational, the documented process must put communicable meaning into the concepts, and express the measurement process in operational terms to:

- identify what data are to be collected,
- define how the data are to be collected and reported,
- define how the data are to be used to make decisions,
- define how the process is to evolve and improve,
- collect and analyze the data,
- make decisions, and
- start over by continuing and/or adjusting the process.

It is important to remember that to be operational, the process must achieve a result nearest to what is needed for a particular end, i.e., it must be feasible with respect to the time and costs associated with collecting data for decision making (based on [Deming 86]).

Problem report – A document or set of documents (electronic or hard copy) used to recognize, record, track, and close problems. (Sometimes referred to as trouble reports, discrepancy reports, anomaly reports, etc.).

Process asset – Documented methods available for use by the projects in developing, maintaining, and implementing the organization's standard software process [Paulk 93].

Process element – Constituent part of the overall software process, e.g., software estimating, software design, code, unit test, peer reviews, and measurement. Each process element covers a well-defined, bounded, closely related set of tasks [Paulk 93].

Software engineering process group - A group of people who facilitate the definition and improvement of the software process used by the organization.

Software measurement process – That portion of the software process that provides for the identification, definition, collection, and analysis of measures that are used to understand, evaluate, predict, or control software processes or products.

Software measurement process architecture – An abstraction of the software measurement process describing required activities for an operational process that can be used to integrate measurement with the overall software process.

Software process – That set of actions (or activities) required to transform a user's need into an effective software solution [Humphrey 89]. In this report, process encompasses the system of all tasks and the supporting tools, standards, methods, and practices involved in the production and evolution of a software product throughout the software life cycle.

Task – A piece of work associated with an activity (e.g., identifying the need for measurement is a task that is part of the identify scope activity of the software measurement process).

References

ami 92 ami: Kuntzmann-Combelles, Annie; Comer, Peter; Holdsworth, Jacqueline; Shirlaw, Stephen, ed. *A Quantitative Approach to Software Management*. Europe: Application of Metrics in Industry Consortium Esprit Project, 1992.

Basili 84 Basili, Victor R.; Weiss, David M. "A Methodology for Collecting Valid Software Engineering Data." *IEEE Transactions on Software Engineering* Vol. SE-10, No. 6 (November 1984), pp. 728-738.

Basili 87 Basili, Victor R.; Rombach, H. Dieter. *TAME: Integrating Measurement Into Software Environments* (TR-1764, TAME-TR-1-1987). College Park, MD: Dept. of Computer Science, University of Maryland, 1987.

Carleton 92 Carleton, Anita D.; Park, Robert E.; Goethert, Wolfhart B.; Florac, William A.; Bailey, Elizabeth; Pfleeger, Shari Lawrence. *Software Measurement for DoD Systems: Recommendations for Initial Implementation* (CMU/SEI-92-TR-19, ADA258305). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, September 1992.

Deming 86 Deming, W. Edwards. *Out of the Crisis*. Cambridge, Mass.: MIT, Center for Advanced Engineering Study, 1986.

Dion 92 Dion, Raymond. "Elements of a Process-Improvement Program." *IEEE Software*, Vol. 9, No. 4 (July 1992), pp. 83-85.

Florac 92 Florac, William A. *Software Quality Measurement: An Architecture for Counting Problems and Defects* (CMU/SEI-92-TR-22, ADA258556). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, September 1992.

Goethert 92 Goethert, Wolfhart B.; Bailey, Elizabeth K.; Busby, Mary B. *Software Effort Measurement: An Architecture for Counting Staff-Hours and Reporting Schedule Information* (CMU/SEI-92-TR-21, ADA258279). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, September 1992.

Grady 87 Grady, Robert B.; Caswell, Deborah L. *Software Metrics: Establishing a Company-Wide Program.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

Humphrey 89 Humphrey, Watts, S. *Managing the Software Process*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1989.

IEEE 89 *IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software*. New York, NY: 1989.

Park 92 Park, Robert E. *Software Size Measurement: An Architecture for Counting Source Statements* (CMU/SEI-92-TR-20, ADA258304). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, September 1992.

Paulish 92 Paulish, Daniel J.; Moeller, Karl-Heinrich. *Software Metrics: A Practitioner's Guide to Improved Product Development*. Los Alamitos, CA: IEEE Computer Society Press, 1992.

Paulk 93 Paulk, Mark C.; Weber, Charles V.; Garcia, Suzanne M.; Chrissis, Mary Beth; Bush, Marilyn. *Key Practices of the Capability Maturity Model, Version 1.1* (CMU/SEI-93-TR-25). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.

QSM 92 QSM. "Creating a Corporate Metrics Program." QSM Perspectives, Spring 1992.

Radice 85 Radice, R.A.; Roth, N.K.; O'Hara, A.C., Jr.; Ciarfella, W.A. "A Programming Process Architecture." *IBM Systems Journal* Vol. 24, No 2 (1985), pp. 79-90.

Rifkin 91 Rifkin, Stan; Cox, Charles. *Measurement in Practice* (CMU/SEI-91-TR-16, ADA241781). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, July 1991.

Rozum 92 Rozum, James A. *Software Measurement Concepts for Acquisition Program Managers* (CMU/SEI-92-TR-11, ADA254177). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, June 1992.

Shewhart 86 Shewhart, Walter A. *Statistical Method from the Viewpoint of Quality Control*, 2d ed. New York, NY: Dover Publications, Inc., 1986.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. **AGENCY USE ONLY** (leave blank)	2. **REPORT DATE** July 1993	3. **REPORT TYPE AND DATES COVERED** Final
4. **TITLE AND SUBTITLE** Establishing a Software Measurement Process		5. **FUNDING NUMBERS** C — F19628-95-C-0003
6. **AUTHOR(S)** McAndrews, D.		
7. **PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. **PERFORMING ORGANIZATION REPORT NUMBER** CMU/SEI-93-TR-016
9. **SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)** HQ ESC/AXS 5 Eglin Street Hanscom AFB, MA 01731-2116		10. **SPONSORING/MONITORING AGENCY REPORT NUMBER** ESC-TR-93-193
11. **SUPPLEMENTARY NOTES**		
12.a **DISTRIBUTION/AVAILABILITY STATEMENT** Unclassified/Unlimited, DTIC, NTIS		12.b **DISTRIBUTION CODE**

13. **ABSTRACT** (maximum 200 words)

This report presents guidelines for establishing a measurement process as part of an organization's overall software process. Methods are suggested that can be used to design a repeatable measurement process that is focused on goal setting, data analysis, and decision making rather than on just data collection and numbers. Examples are included to illustrate these methods in the context of some common software process management issues involving basic measures of size, effort, and schedule. This report also suggests some steps for starting a measurement program. These steps focus on identifying an organizational strategy for improvement and designing a process for measurement to support this strategy.

14. **SUBJECT TERMS**			15. **NUMBER OF PAGES** 57 16. **PRICE CODE**
17. **SECURITY CLASSIFICATION OF REPORT** UNCLASSIFIED	18. **SECURITY CLASSIFICATION OF THIS PAGE** UNCLASSIFIED	19. **SECURITY CLASSIFICATION OF ABSTRACT** UNCLASSIFIED	20. **LIMITATION OF ABSTRACT** UL