Applications of the Indicator Template for Measurement and Analysis

Wolfhart Goethert
Jeannine Siviy

September 2004

Software Engineering Measurement and Analysis Initiative

Technical Note
CMU/SEI-2004-TN-024

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

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Contents

List of Figures

Abstract

Organizations often do not achieve the potential benefits of a sound measurement program due to the inconsistent construction and interpretation of indicators derived from measurement data. This technical note presents guidance for adapting and completing an *indicator template*—a tool the Software Engineering Institute has developed to precisely describe an indicator—including its construction, correct interpretation, and how it can be utilized to direct data collection and presentation and measurement and analysis processes. An indicator template can help an organization to define *indicators*, or graphical representations of measurement data, which describe the who, what, where, when, why, and how for analyzing and collecting measures. This technical note defines each field of the indicator template, provides example inputs, and shows how the template may be used in the context of a process improvement effort that uses the Capability Maturity Model® Integration framework and/or Goal-Driven Software Measurement.

1 Introduction

The Software Engineering Measurement and Analysis (SEMA) team at the Carnegie Mellon® Software Engineering Institute (SEI℠) promotes the use of measurement for improving the management and work processes of software development and acquisition. SEMA works with representatives from industry, government, and academia to develop basic measurement techniques and measurement processes that can be used to systematically and repeatedly measure software development organizations, projects, products, and processes.

The SEI has found that an indicator template can help an organization to improve its software measurement processes and infrastructure. In turn, it serves as a tactical aid in the execution of the measurement process. Just as completing an indicator template helps to define or improve a measurement process; its contents can enrich and further define what the measurement process means and guide improvements in performance at the project and organizational levels.

The SEI defines an ***indicator*** as a representation of measurement data that provides insight into software development processes and/or software process improvement activities. A *measure* quantifies a characteristic of an item whereas an indicator may use one or more measures. For example, an indicator may be the trend of a measure over time or the ratio of two measures.

An ***indicator template*** is used to precisely document an indicator, its construction, correct interpretation, as well as to direct its data collection, presentation, and measurement and analysis processes. It helps to ensure the consistent collection of measures for constructing the indicator and provides a set of criteria for ensuring the consistent interpretation of the measures collected. This technical note describes a template that can be used to precisely describe, document, and report the who, what, when, where, why, and how of an organization's indicators. It also describes the use of the indicator template within the context of the Goal-Question-Indicator-Measurement (GQ[I]M) methodology and the Capability Maturity Model® Integration (CMMI®) framework.

[SM] SEI is a service mark of Carnegie Mellon University.

[®] Capability Maturity Model, Carnegie Mellon, and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

1.1 Challenges in Software Measurement

When beginning to leverage the measurement and analysis processes they have established, organizations frequently encounter problems, such as

- analysis misses the “big picture”
- charts are colorful but meaningless
- charts/indicators are misinterpreted
- inconsistent definitions of measures and data elements are used
- context of the indicators is not understood
- data set includes inaccurate information
- no baseline or benchmark exists for comparing current performance
- infrequent or ineffective data integrity activities
- comparing or predicting process results without ensuring stability of processes

The consequences of these problems can be significant. For example, graphical charts of data that “look pretty” may lack substance and not support real-time or post-project decision making. Worse, they can skew perceptions of trends and predictions and create misrepresentations that don’t reflect or allow comparisons to historical performance.

Further, invalid data must be removed from a data set. Often this leaves the practitioner with a very small subset of data upon which to draw conclusions. The SEI has worked with organizations that have discarded as much as 70% of their data set. Depending on the organization and the duration of the project life cycle, this can result in delays (sometimes more than one year) to obtain a new, valid data set for analysis.

Underlying these problems and unintended consequences may be a measurement infrastructure that has been only partially defined and designed. Indications of poor definition and design include

- unclear goals
- inconsistent data collection practices across the organization
- data elements lacking well defined operational definitions
- inconsistent construction and interpretation of indicators
- data presentations using “easy” instead of effective Excel graphing option

There is much guidance available to address these issues. For instance, the Raytheon Corporation has provided guidance for developing and documenting metrics in *A Management Guide for the Development and Deployment of Strategic Metrics* [Raytheon 98].

The following guidance and questions are offered to assist an organization in developing "good" indicators:

- Establish clear objectives.
- Know the purpose of having the indicator.
- What do you want to know when you receive this information?
- Who is responsible? Who is the owner of the indicator? Who is accountable for the measurement information? Who is the customer or user of the indicator?
- How do you get the data? What are the key drivers of the data? How good (accurate) is the data?
- Is your indicator clearly defined?
- How do your calculate your indicator?
- How often do you report your indicator? When must the user receive it to be of value or use to him/her?

Questions such as these are the underpinnings of the indicator template. Using the template as guidance to address the details of measurement infrastructure enables an organization to more fully realize the benefits of successful measurement and analysis [Baumert & McWhinney 92]. These include the following results:

- insight into product development
- capability to quantify tradeoff decisions
- better planning, control, and monitoring of projects
- better understanding of both the software development process and the development environment
- identification of areas of potential process improvement as well as an objective measure of the improvement efforts
- improved communication

Additional guidance on implementing measurement programs can be found in *Experiences in Implementing Measurement Programs* [Goethert & Hayes 01]. This technical note describes lessons learned from several organizations that have implemented measurement programs using the Goal-Driven Software Measurement methodology. It contains a description of the methodology, a discussion of challenges, obstacles, and solutions, an initial set of indicators and measures, as well as some artifacts (such as templates and checklists) that have been found to enable successful implementations.

1.2 Purpose

This document describes a comprehensive template for developing and precisely documenting an organization’s indicators. By completing the indicator template, an organization is given a consistent and precise method to follow with documented results. The fields of the template address the information needed to specify, implement, and interpret an indicator. This template has been used by many organizations for several years and has met with much success. Many organizations tailored the indicator template to fit their unique environment. Through the guidance and examples provided in this document it is hoped that many more organizations will find value in using the indicator template.

The objectives of this document are to

- Provide a template that can be used to document the construction and use of indicators.
- Provide examples of how this indicator may be modified to adapt to unique information needs.
- Provide information on how the indicator template relates to the CMMI models.
- Provide examples of how this template has benefited several organizations’ measurement programs.

2 Indicator Template

Many organizations have recognized the importance of using precise communication and collecting measurements based on need rather than capability. This realization has led many organizations to develop their own site-specific measurement and indicator templates.

The indicator template that accompanies Goal-Driven Software Measurement, and which is described in this section, reflects the thinking and practices of multiple organizations over time. It has been shown to reduce cycle time by enabling organizations to leverage aggregate experience and to quickly focus on measurement content rather than form.

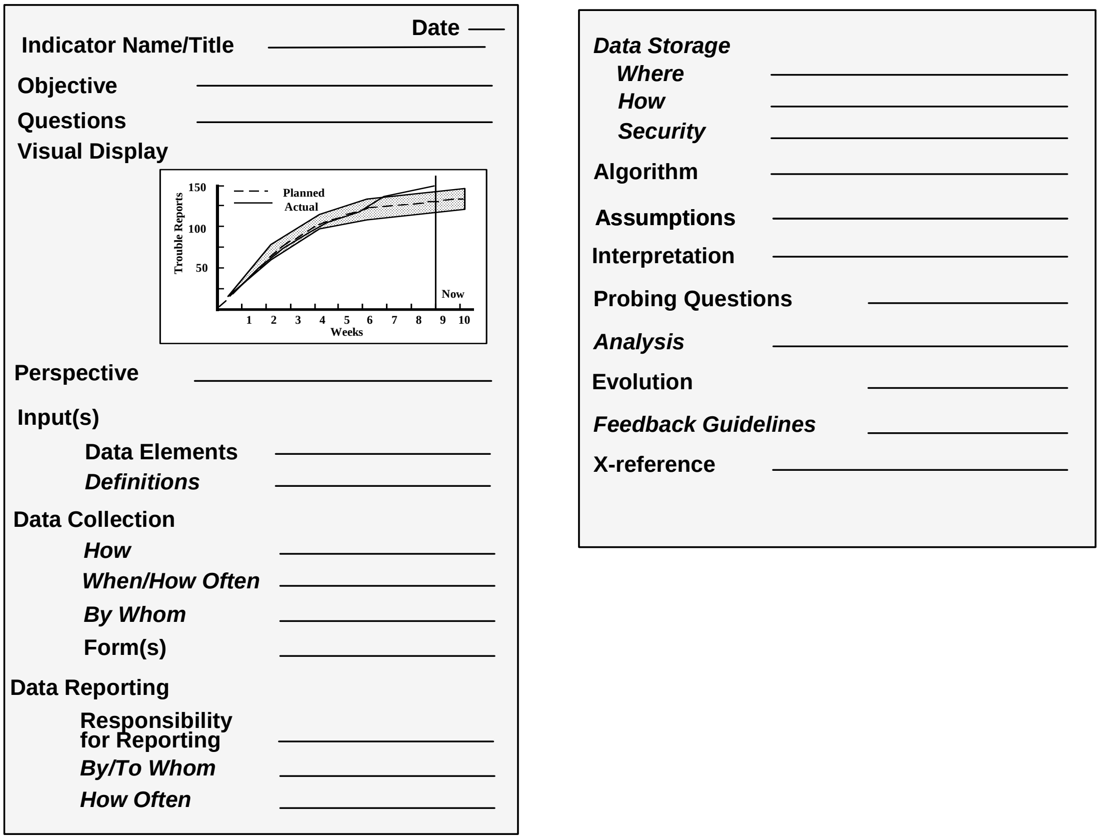
For example, one organization has used the indicator templates to define success criteria and an associated measurement infrastructure for each measurement goal. Previously, the organization had been tracking and reporting an overwhelming number of measures throughout all levels of management. However, after a goal-setting workshop led by the SEI, the organization was able to develop a vertically aligned goal structure and common operational definition of success.

Another organization, which had a well defined, enterprise-wide measurement infrastructure, modified the indicator template to focus and align its highest-priority indicators and measures at the organization and project levels. This allowed the organization to pursue higher levels of maturity and to develop quantitatively managed processes.

At EDS, the organization used the indicator template to get control of its financial and contractual requirements. Before using the template, the organization’s business development goals were articulated but rarely acted upon and had little measurement support. After trying a number of approaches such as one-on-one interviews and using “guru” experiences that resulted in weak measures, EDS adopted the indicator template and integrated it into its processes. EDS now describes the indicator template as the “cornerstone” of its successful measurement and process improvement efforts [Crawford & Stephens 04].

2.1 General Indicator Template Structure

The indicator template contains fields to precisely document the construction, interpretation, and use of indicators. The template has evolved over time as organizations tailored it to fit their unique environment. Figure 1 depicts the template structure. Fields that have been added since its initial creation are indicated in italics.



Date

Indicator Name/Title

Objective

Questions

Visual Display

150
100
50
Trouble Reports
Planned
Actual
Now
1
2
3
4
5
6
7
8
9
10
Weeks

Perspective

Input(s)

Data Elements

Definitions

Data Collection

How

When/How Often

By Whom

Form(s)

Data Reporting

Responsibility for Reporting

By/To Whom

How Often

Data Storage

Where

How

Security

Algorithm

Assumptions

Interpretation

Probing Questions

Analysis

Evolution

Feedback Guidelines

X-reference

Figure 1: Indicator Template

The fields of the indicator template are briefly defined below. Comprehensive descriptions of each template field appear in Appendix A.

- indicator objective: the objective or purpose of the indicator
- questions: the questions that the user of the indicator is trying to answer
- visual display: a graphical view of the indicator
- perspective or viewpoint: the description of the audience for whom the indicator is intended
- inputs: the list of the measures required to construct the indicator and its definitions
- algorithms: the description of the algorithm used to construct the indicator from the measures
- assumptions: the list of assumptions about the organization, its processes, life-cycle model, and so on that are important conditions for collecting and using the indicator
- data collection information: information pertaining to how, when, how often, by whom, etc. the data elements required to construct the indicator are collected
- data reporting information: information on who is responsible for reporting the data, to whom, and how often

- data storage: information on storage, retrieval, and security of the data
- analysis and interpretation of results: information on how to analyze and interpret as well as to not misinterpret the indicator

Beyond this standard structure, organizations can and should further tailor the template to fit their unique environment. Adding, modifying, or deleting fields, in advance of documenting a set of indicators, can help to ensure the template will be accepted and implemented by the organization. The template in this document provides a good starting point and incorporates the fields that have been found to be useful to many organizations.

Some examples of modifications and uses of the indicator template by organizations to fit their unique environment include the following:

- replacing the tables for data collection and reporting with “swimlane” diagrams that showed both the data flow and the responsible parties
- adding a field for goals and subprocess selection to ensure clarity in the indicator’s purpose and its relationship to process improvement
- adding a section for “corrective action guidelines.” This section, the concept of which was borrowed from manufacturing, guides the user to the steps beyond interpretation and probing questions. It provides guidance for the appropriate response to special cause variation or other data patterns that require action, whether project-specific or systemic.
- adding descriptive information that clearly ties measures to processes which they describe and/or control

2.2 Example Indicator Templates

To illustrate its use in practice, several completed templates are included in the appendices of this document:

- Appendices B and C address cycle time and illustrate how two organizations have tailored the template to document the attributes of the cycle time indicator in their unique requirements.
- Appendix D contains an example of earned value management (cost and schedule).
- Appendix E contains an example of an indicator that documents the status of software engineering processes.

3 Integrating the Indicator Template

The indicator template may be used to develop indicators one-at-a-time on an as-needed basis or as a component of the measurement infrastructure development process within a process improvement effort. This section briefly highlights the GQ(I)M approach to measurement infrastructure development in which the indicator template plays a key role. It briefly highlights the relationship of the indicator template to CMMI.

3.1 The GQ(I)M Methodology

The indicator template is a key artifact of the GQ(I)M methodology. In order to fully appreciate the template it is important to have a working understanding of the principles and key steps of the methodology.

Many organizations use the GQ(I)M methodology when deciding what to measure to achieve its business goals. The "I" in parentheses distinguishes the GQ(I)M methodology from the closely related GQM methodology introduced and described by Basili and Rombach [Basili & Rombach 88] [Basili & Weiss 84] [Basili 89] [Rombach 89]. The GQ(I)M methodology identifies and defines software measures that directly support an organization's business, process improvement, and project goals, ensuring relevance and traceability from their goals to the data collected. The indicator template, as a key artifact, is used to precisely describe and document the who, what, where, when, why, and how of an indicator and to document its alignment with the goals of an organization. It ensures consistent collection of the measures used to construct an indicator and provides additional fields to ensure a consistent interpretation of an indicator.

The Goal-Driven Software Measurement approach is described in the SEI's *Goal-Driven Software Measurement Guidebook* [Park et al. 96]. The steps of the GQ(I)M approach, as implemented by the SEI, are organized into three general sets of activities [Zubrow 98]:

1. Goal identification

In measurement, the question should not be "What indicators should I use?" but "What do I want to know or learn?" and "What decision do I want to make?"

As a starting point to address these questions, an organization may find it useful to develop a goal structure, as shown in Figure 1. Such a diagram clarifies an organization's strategic mission and provides a structure against which to elaborate on the above questions.

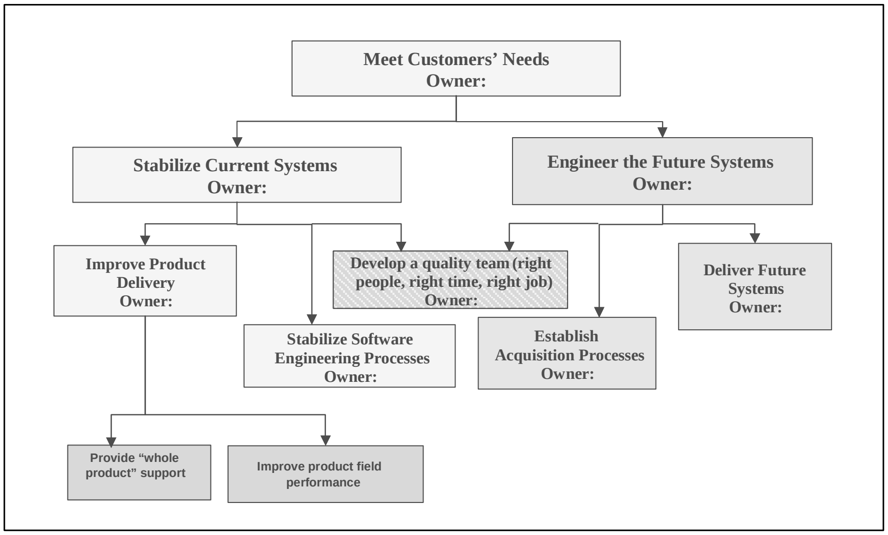

Meet Customers' Needs
Owner:
Stabilize Current Systems
Owner:
Engineer the Future Systems
Owner:
Improve Product Delivery
Owner:
Develop a quality team (right people, right time, right job)
Owner:
Deliver Future Systems
Owner:
Stabilize Software Engineering Processes
Owner:
Establish Acquisition Processes
Owner:
Provide "whole product" support
Improve product field performance

Figure 2: Goal Structure Illustration Example

Frequently, the questions of interest are "How will I know if I have achieved the goal?" and "How will I know if I am making progress toward the goal?"

Many organizations have difficulty deciding if or when their business goals have been achieved. While many organizations easily articulate a strategy and define tasks for achieving their goals, they have difficulty understanding the difference between **success indicators** (indicators used to determine if the goals have been met) and **progress indicators** (indicators used for tracking the execution of tasks). These indicators are show in Figure 2. The execution of defined tasks is a necessary but insufficient condition for meeting an organization's business goals. Analyses of the outcome of these tasks are part of the decision-making process for determining if the goals have been met successfully.

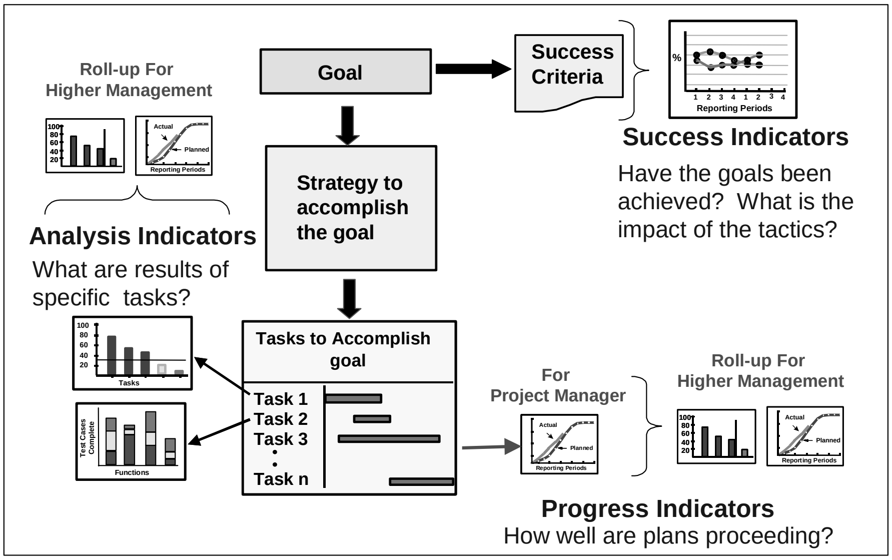


Figure 3: Indicator Classification

Figure 2 illustrates three types of indicators, defined as follows:

1. **success indicators**: These indicators are constructed from the defined success criteria and are used to determine if the goals have been met. An example of a success indicator is a trend chart with a target goal on it. The goal may be a quality, market share, or revenue target. Balanced scorecards contain these types of indicators.
2. **progress indicators**: These indicators are used to track the progress or execution of the defined tasks. Earned value and a Gantt chart are good examples of this type of indicator. Note, as mentioned above, that the successful execution of all the defined tasks does not necessarily guarantee that the goal has been successfully met. However, failure to execute the plan should be of great concern.
3. **analysis indicators**: These indicators are used to assist in analyzing the output of each task. An indicator that plots the number and type of defect detected in each phase of a development is an example of this type of indicator. The analyses help test our assumptions about the data we are using to judge progress and success.

Goal-Driven Software Measurement, which does not rely on a list of predefined indicators, provides a roadmap for identifying measures required to construct the different types of indicators. Additionally, it can provide a framework for selecting indicators and measures from a previously defined set. Practical Software and Systems Measurement (PSM) provides a catalog of indicators and measures on its Web site (http://www.psmsc.com). In both cases, it is critical to understand the measurement goals of the organization and the needed decisions or information needs which the indicators will help to solve.

2. Indicator identification and data specification

In our elaboration of Basili's methodology we have added an intermediate step to assist in linking the questions to the measurement data that will be collected. Our experience suggests that identifying questions and measures without visualizing an indicator is insufficient for starting a successful measurement program. The indicators or reports used to communicate the data (called indicators in our variation of the GQM methodology) are a key link for determining the success or failure of a measurement program. These indicators serve as a requirement specification for the data that must be gathered, the processing and analysis that must take place, and the schedule by which these activities occur.

Figures 4 and 5 depict, at a high level, the GQ(I)M methodology and its ten-step process as taught in workshops at the SEI.[1] As shown in Figure 6, the output or results of each step in the methodology is documented in the indicator template.

[1] Another excellent reference and comprehensive synthesis of the GQM concept is available in *The Goal/Question/Metric Method, A Practical Guide for Quality Improvement of Software Development* by van Solingen and Berghout [van Solingen and Berghout 99].

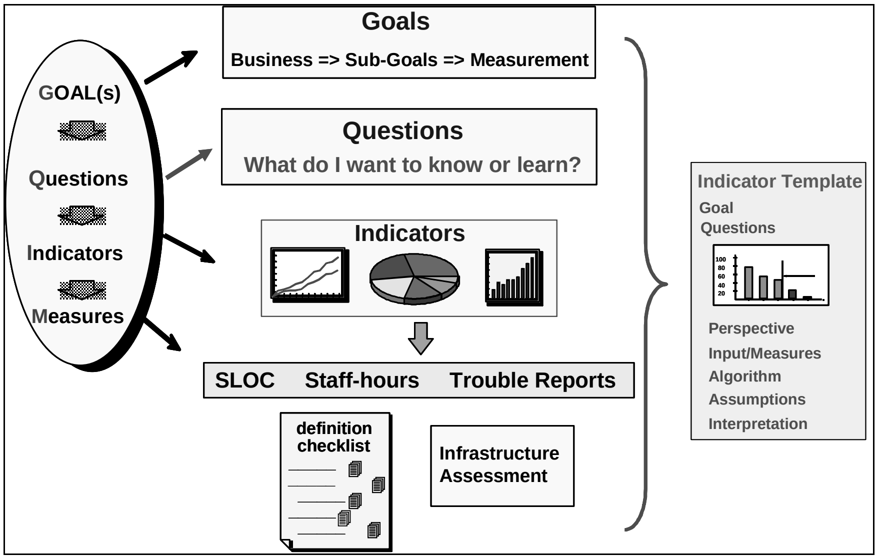


Figure 4: Goal-Driven Measurement Methodology

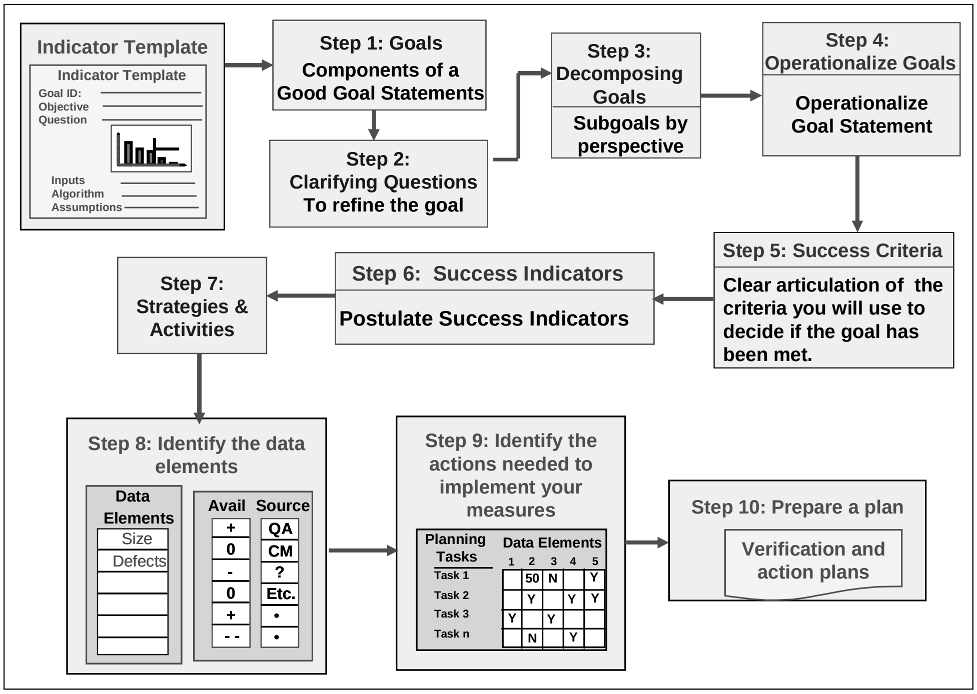


Figure 5: Ten Steps of Goal-Driven Software Measurement

Many organizations collect completed indicator templates in a measurement handbook to ensure consistent construction and interpretation of its indicators. They compile the completed indicator templates to create a catalog of approved and proven indicators for their organization. The following figure illustrates the contents of one such handbook and maps outputs obtained from a GQ(I)M workshop conducted with the SEI.

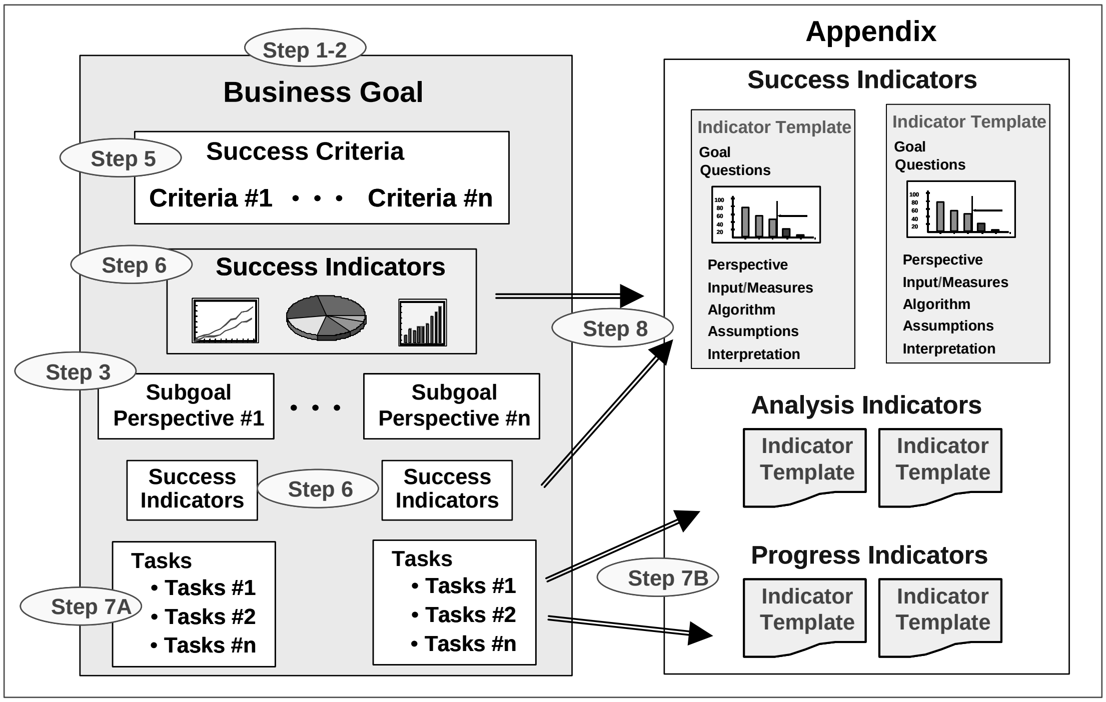


Figure 6: Contents of a Measurement and Analysis Handbook

3. **Infrastructure assessment and action planning to guide the implementation**

   Existing data collection and measurement activities within an organization are analyzed to avoid duplication and to identify gaps. Priorities, in terms of data needed to produce the indicators, are assigned. Tasks are defined to take advantage of existing activities and to address the gaps.

3.2 CMMI and the Indicator Template

The Measurement and Analysis process area (PA) of CMMI calls for the establishment of a measurement and analysis infrastructure and explains what the measurement infrastructure should include. Methodologies such as GQ(I)M (and others) provide tactical approaches for identifying the indicators and defining and implementing the required infrastructure. As an aid to CMMI adopters who would like to use GQ(I)M, this section maps the indicator template as a key artifact of GQ(I)M to the Measurement and Analysis practices.

3.2.1 The CMMI Measurement and Analysis Process Area

The purpose of the Measurement and Analysis PA is to develop and sustain a measurement capability that is used to support management information needs [CMMI 02]. In support of its Measurement and Analysis PA implementation, many organizations have found the indicator template to be a practical and useful guide for specifying and/or documenting data collection, storage, analysis, and reporting of measurement data via indicators. The Measurement and Analysis PA is linked to the generic practices used to institutionalize each of the other CMMI PAs. As such, the indicator template also serves to support these generic practices.

The Measurement and Analysis PA involves the following:

- Specifying the objectives of measurement and analysis such that they are aligned with identified information needs and objectives
- Specifying the measures, data collection and storage mechanisms, analysis techniques, and reporting and feedback mechanisms
- Implementing the collection, storage, analysis, and reporting of the data
- Providing objective results that can be used in making informed decisions, and taking appropriate corrective actions

The initial focus for measurement activities is at the project level. However, a measurement capability can be useful for addressing organization- and/or enterprise-wide information needs [CMMI 02].

The Measurement and Analysis PA goals are to align measurement and analysis activities and to provide measurement results. Figure 7 shows the specific practices associated with the goals (Specific Goal 1: Align Measurement and Analysis Activities; Specific Goal 2: Provide Measurement Results).[2] Figure 8 shows how these specific practices align with the fields of the indicator template. For organizations using the indicator template, its contents provide the needed documentation for all the practices of the goal to “Align Measurement Activities” and point to the processes, roles, responsibilities, and organizational process assets associated with the specific practices of the goal to “Provide Results.”

[2] Figure 7 is derived from CMMI training materials. These materials are not publicly available.

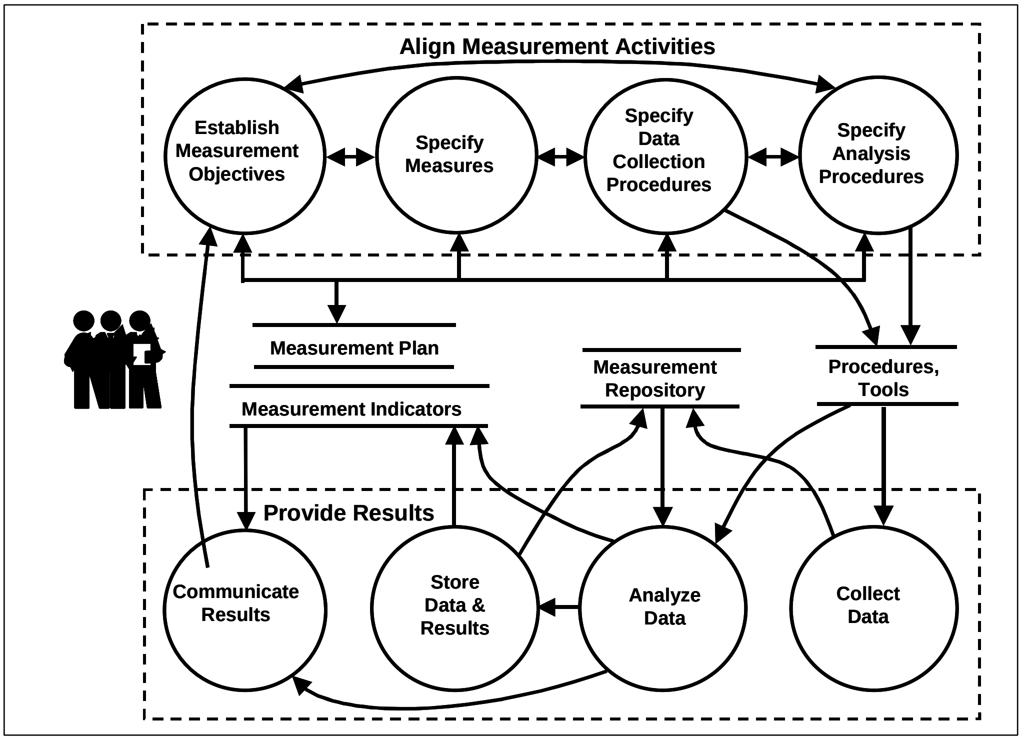


Figure 7: CMMI Measurement and Analysis PA Measurement Activities

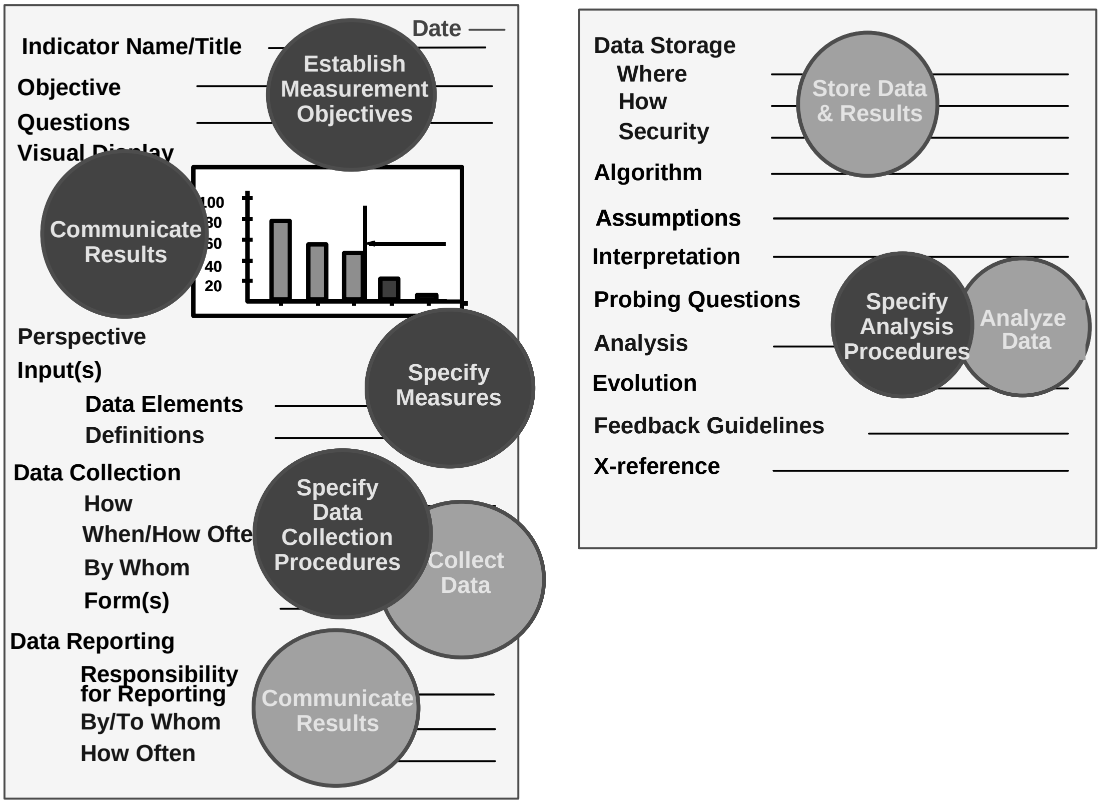


Figure 8: CMMI Measurement Practices Mapped to the Indicator Template

4 Summary

The indicator template described in this technical note is a tool an organization can use to precisely describe and document an indicator or graphical representation of measurement data, its construction, and correct interpretation. It can also serve to direct an organization's data collection, presentation, and measurement and analysis processes. The indicator template provides a comprehensive template to document the who, what, where, when, why, and how of an indicator to ensure consistent construction and interpretation. Guidance for adapting and completing the indicator template ensures that the potential benefits an organization can derive from a sound measurement program are realized.

We would appreciate hearing from you and learning about your experience using the indicator template. An electronic version of the template is available by contacting SEI Customer Relations (customer-relations@sei.cmu.edu).

Appendix A Indicator Template

The current version of the indicator template for describing the indicators is shown below. Fields which have been added based on repeated use of the template are shown in italics.

INDICATOR TEMPLATE

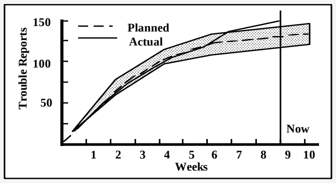
Date

Indicator Name/Title

Objective

Questions

Visual Display



Perspective

Input(s)

Data Elements

Definitions

Data Collection

How

When/How Often

By Whom

Form(s)

Data Reporting

Responsibility for Reporting

By/To Whom

How Often

Data Storage

Where

How

Security

Algorithm

Assumptions

Interpretation

Probing Questions

Analysis

Evolution

Feedback Guidelines

X-reference

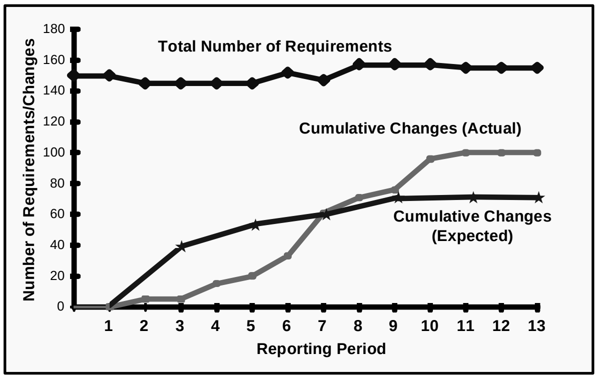
Date __________

Indicator Name/Title: ___________________________

OBJECTIVE Describe the objective or purpose of the indicator.

QUESTIONS List the question(s) the indicator user is trying to answer. Examples: Is the project on schedule? Is the product ready to ship? Should we invest in moving more software organizations to CMM maturity level 3?

VISUAL DISPLAY Provide a graphical view of the indicator.



PERSPECTIVE Describe the audience (for whom is this display intended) for the visual display.

INPUTS

Data Elements

List all the data elements in the production of the indicator.

____

____

____

____

____

Definition

Precisely define the data element used or point to where the definition can be found.

____

____

____

____

____

DATA COLLECTON

How Describe how the data will be collected.

When/How Often Describe when the data will be collected and how often.

By Whom Specify who will collect the data (an individual, office, etc.)

Forms

Reference any standard forms for data collection (if applicable) and provide information about where to obtain them.

DATA REPORTING

Responsibility for Reporting

Indicate who has responsibility for reporting the data.

By/To Whom

Indicate who will do the reporting and to whom the report is going to. This may be an individual or an organizational entity.

How Often

Specify how often the data will be reported (daily, weekly, monthly, as required, etc.)

DATA STORAGE

Where

Indicate where the data is to be stored.

How

Indicate the storage media, procedures, and tools for configuration control.

Security

Specify access to this data will be controlled.

ALGORITHM

Specify the algorithm or formula required to combine data elements to create input values for the indicator. It may be very simple, such as Input1/Input2, or it may be much more complex. It should also include how the data is plotted on the graph.

ASSUMPTION

Identify any assumptions about the organization, its processes, life cycle models, and so on that are important conditions for collecting and using this indicator.

ANALYSIS

Specify what type of analysis can be done with the information.

INTERPRETATION

Describe what different values of the indicator mean. Make it clear how the indicator answers the “Questions” section above. Provide any important cautions about how the data could be misinterpreted and measures to take to avoid misinterpretation.

PROBING QUESTIONS List questions that delve into the possible reasons for the value of an indicator, whether performance is meeting expectations or whether appropriate action is being taken.

EVOLUTION Specify how the indicator can be improved over time, especially as more historical data accumulates e.g., by comparison of projects using new processes, tools, environments with a baseline; using baseline data to establish control limits around some anticipated value based on project characteristics.

FEEDBACK GUIDELINES

A description of the procedure to use when recommending modification to the indicator template.

X-REFERENCES If the values of other defined indicators influence the appropriate interpretation of the current indicator, refer to them here

Appendix B Cycle Time Example from Company A

Cycle Time

A. Purpose:

To assess the impact of our investment in Software Process Improvement on the length of time to deliver software products from the perspective of the Chief Executive Officer.

Type: Schedule

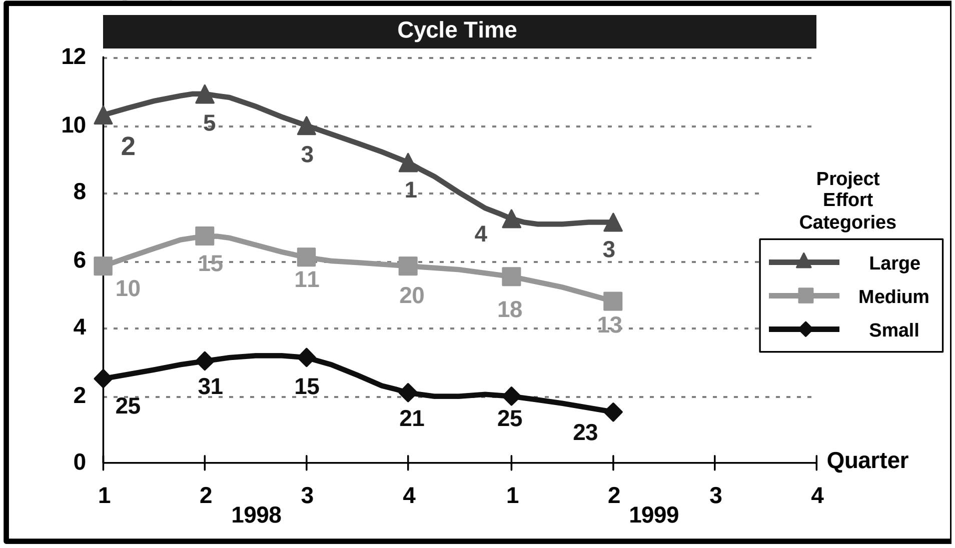
B. Definition:

Cycle Time illustrates the trend for <u>schedule duration</u> for all software projects within the organization completing in each quarter. This indicator represents the average number of calendar days per feature delivered by each project. Completed projects are grouped into three categories of similar levels of effort.

C. Questions:

1. How does the cycle time of recently completed projects compare to that of those completed earlier in our implementation of Software Process Improvement?
2. Has Software Process Improvement affected the cycle time of our small, medium, and large projects similarly over time?
3. What is the typical cycle time for our small, medium, and large projects?

D. Visual Display:



E. Calculations:

Algorithm:

Data is to be segregated into three Project Effort Categories (i.e. small, medium, and large) and submitted only for projects completing in the quarter.

For each project which completes during the quarter:

$$\text{Project Duration} = \text{Actual End Date} – \text{Actual Start Date}$$

$$\text{Project Cycle Time} = \frac{\text{Project Duration}}{\text{Delivered Features}}$$

For all project completing in a quarter within each Project Effort Category

$$\text{Category Cycle Time} = \frac{(\text{Project Duration}_1 + \text{Project Duration}_2 + \ldots + \text{Project Duration}_n)}{\text{Delivered Features}_1 + \text{Delivered Features}_2 + \ldots + \text{Delivered Features}_n)}$$

Category Cycle Time is charted in each quarter for each Project Effort Category. No data point is charted in a given category for those quarters in which no project completes. In addition, the number of projects (i.e.) associated with a data point is indicated in the graph if that count is greater than one.

Data Elements: (refer to Section 3 for detailed Data Element Definition Checklists)

For each project completed during the quarter:

Element	Definition
Actual Start Date	Phase 3.3 Entrance
Actual End Date	Phase 3.4 Exit
Delivered Features	The number of allocated requirements that are in the launched software
Total Actual Effort	The number of hours of effort applied by the software team to the project between the actual start and actual end date.
Project Name	Unique project name to which this information applies.
Data Source	Name of the individual who supplied this information.

Organization reference information:

Element	Definition
Project Effort Categories	Each completed project is included in one of three groups based upon a comparisons of its Total Actual Effort to the ranges for the following categories: Large > 30 staff years Medium > 10 to 30 staff years Small < 10 staff years

Example:

Calculation for each project:

As of the end of Phase 3.2 for project “A”, Phases 3.3 through 3.4 were planned to span January through June but actually lasted through August. This project delivered 17 features. The calculation is illustrated below:

Key Data Elements

Actual Start Date = 1 January
Actual End Date = 31 August
Delivered Features = 17 features

Calculations

Project Duration = Actual End Date – Actual Start Date
= 31 August – 1 January = 243 Calendar Days

$$\text{Project Cycle Time} = \frac{\text{Project Duration}}{\text{Delivered Features}} = \frac{\text{243 calendar days}}{\text{17 features}}$$

$$= \text{14.29 calendar days per feature}$$

Calculation for each category:

The following table will be used to illustrate the calculations for one data point to be plotted on the Cycle Time indicator. Assume that the following projects were completed in the quarter and that all projects are ‘medium’ effort projects.

Project	Project Duration	Delivered Features
A	243	17
B	375	35
C	400	44
D	350	15
E	385	42
F	250	25
G	210	33
H	211	25

The calculation is illustrated below.

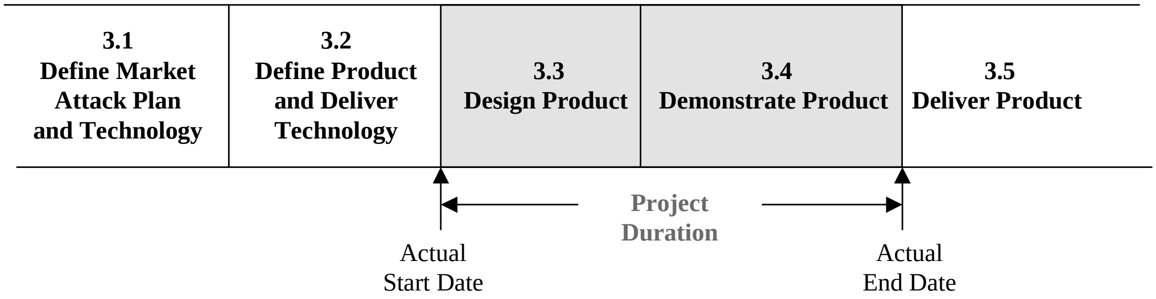
$$\text{Category Cycle Time} = \frac{(\text{Project Duration}_1 + \text{Project Duration}_2 + \ldots + \text{Project Duration}_n)}{(\text{Delivered Features}_1 + \text{Delivered Features}_2 + \ldots + \text{Delivered Features}_n)}$$

$$= \frac{243 + 375 + 400 + 350 + 385 + 250 + 210 + 211}{17 + 35 + 44 + 15 + 42 + 25 + 33 + 25}$$

$$= 10.3 \text{ calendar days per feature}$$

F. Assumptions:

1. Only projects completed in the indicated quarter are included in the information represented.
2. Use of the Core Process to manage the project and establish key milestones:



G. Interpretation:

Probing Questions:

1. Is the long-term trend for this indicator consistent with expectations?
2. Is the short-term trend for this indicator consistent with that of related indicators?

Evolution:

Additional indicators may supplement this indicator over time according to experience and relevance. Anticipated potential backup indicators include:

- Cycle Time vs. Quarter (% change over time)

This indicator may be revised over time as additional software process capability is established across the organization. Anticipated potential modifications include:

- Tuning of Project Effort Category ranges according to experience
- Refinements to the unit of size against which schedule is normalized (other than ‘Delivered Features’)

Cross-References:

- Cycle Time Slip Rate
- Feature Slip Rate

H. Reporting:

Status:	Core measure
Organizational Scope:	All Business Groups and Support Organizations with CMM-based Software Process Improvement programs
Project Scope:	Each software subsystem for all projects within the Product Pipeline
Policy Requirement:	Yes
Responsibility:	Deployment Manager

Appendix C Cycle Time Example from Company B

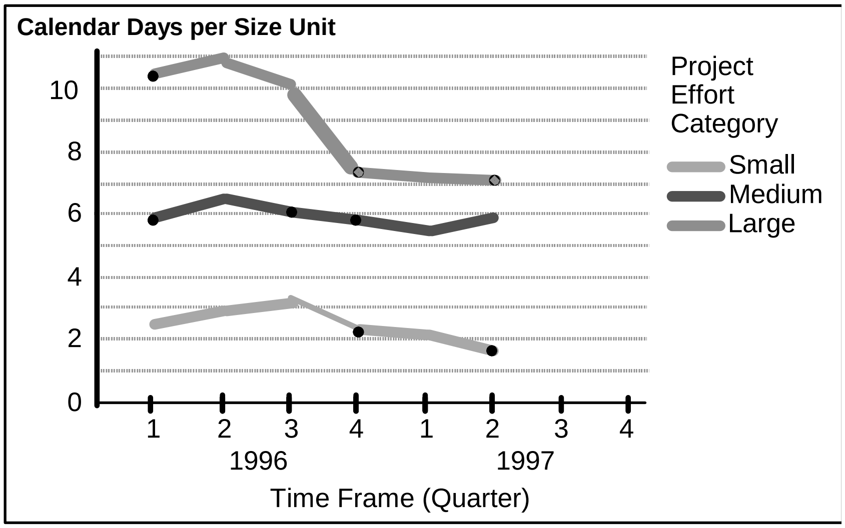
Cycle Time

Objective To monitor trends in development elapsed time as input toward improvement at the technical unit level and across the Enterprise.

Questions

- What is the cycle time trend for each of the project effort categories?
- Are the trends the same for the different project effort categories?
- What is the rate of change from year to year?
- How does the rate of change compare among the different project effort categories?

Indicator/Display



Inputs Data is to be segregated into three project effort categories (small, medium, and large) and only submitted for projects completed during the quarter.

Data Elements: There are two types of input data:

Organizational Reference information:

- Name of Organization
- SEI Maturity Level

- Reporting Period
- Contact Person
- Contact Phone Number

For each project completed during the period, the following Cycle Time Indicator Data:

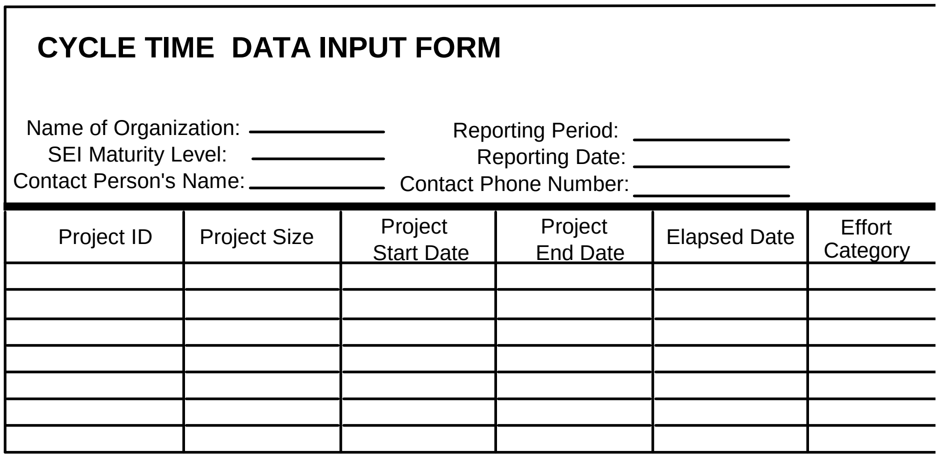
- Project ID
- Project Effort Category
- Project Start Date
- Project End Date
- Project Elapsed Days
- Project Size in standard units, according to Citicorp guidelines

Responsibility for Reporting:

The project manager is responsible for collecting and submitting the input.

Forms:

The consolidated data must be submitted using the Form shown below.



CYCLE TIME DATA INPUT FORM

Name of Organization: ____________
SEI Maturity Level: ____________
Contact Person's Name: ____________
Reporting Period: ____________
Reporting Date: ____________
Contact Phone Number: ____________

Project ID	Project Size	Project Start Date	Project End Date	Elapsed Date	Effort Category

Algorithm

- The completed projects are grouped into the three effort categories (Small, Medium, Large) according to the criteria described in the size determination section of this document.
- For each project effort category, the average project elapsed time per size unit is calculated using the following formula:

Average Elapsed Time per Size Unit =

$$\frac{\text{Sum of Project Elapsed Times}}{\text{Sum of Project Sizes}}$$

- The Average Elapsed Time per Size Unit is plotted for each effort category.

Assumptions

- Measurements are based on elapsed calendar days without adjustment for weekends or holidays.
- Projects are categorized into the three effort categories (Small, Medium, Large) according to the criteria described in the size determination section of this document.
- Project size is measured and converted into standard size units according to Citicorp guidelines.

Interpretation A steep downward/decreasing trend can be the result of multiple factors,

- The specific project category was inefficient to begin with.
- We successfully invested to improve the process/methodology/tools used for this project category.
- Re-use was successfully deployed.
- We have increasing expertise working on these projects.

A level trend can be the result of canceling effects from different projects. For example, suppose there are 10 units across the bank that have submitted data, and five of these units have shown improvements, but the other five have negative improvement. The results is a level trend at the organization level. Therefore when we see a level trend we may want to investigate what % of organization have improved and what % of organization did worse than before.

An upward/increasing trend can be the result of changing technology or processes. We may want to look for changes in the process/method/tools, major business changes that greatly affect the management of business requirements, exceptional events that impact the development effort, such as, a data center disaster.

X-References Schedule Predictability
Size

Probing Questions

- Has the process/methodology (including different development techniques and tools) being used been changed?

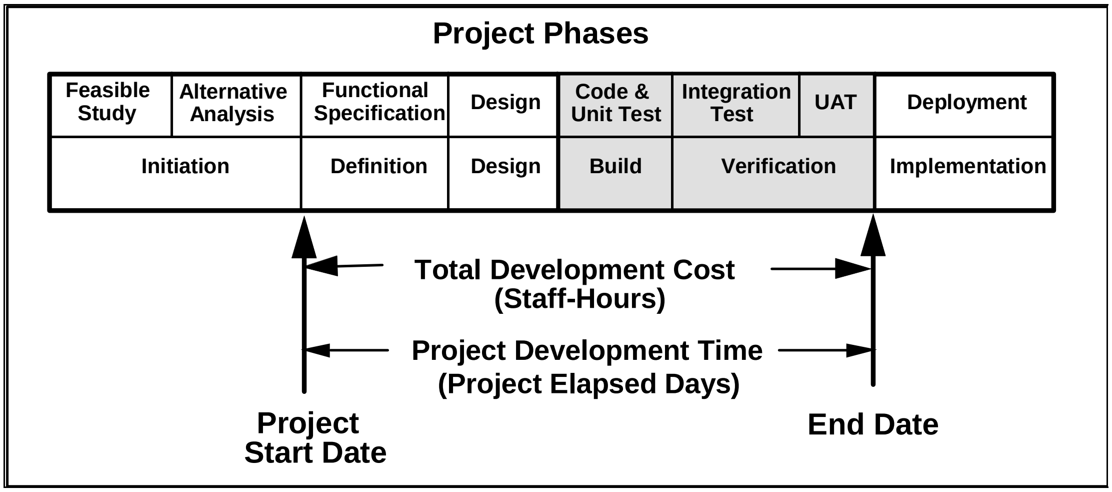
- What are the new process/methodologies?
- What are the exceptional events?

Evolution At the initial data collection stage the data will be used to establish the baseline, i.e. we establish the “Average Project Elapsed Time for the different Project Categories. This can be used as the initial organizational goal to measure our projects’ cycle time distribution. At a minimum, this quantity indicator should be used as input to all project estimates.

After the initial two years we can then use the data to establish future Project Development Process Goals, i.e. x% improvement over Y years. While this is an organizational (enterprise) goal, at the organizational/project levels it can be further refined. This should be a standard practice for the Level 3 and above organizations.

Over time we will re-establish the baseline. This will be essential with ongoing technology changes and as more and more organizations in Citibank advance up the SEI/CMM maturity ladder.

Definitions Project Cycle Time**:** The project cycle time used for the Cycle Time indicator is illustrated below.



Project ID: A site dependent identification of the each project.

Project Start Date: Actual calendar date the project starts. When users formally accept one of the alternatives recommended by technology to meet their initial request. Typical this is after the feasible study and alternative analysis phase, before the Functional Specification phase begins.

The project start time is immediately after the receiving the Request for Services.

Project End Date: Actual calendar date the project ends. When the user formally signs off the UAT. This can be the UAT for the first pilot site or there is a regional/global business user group that performs the UAT and signs off the acceptance at the regional/global level.

Project Elapsed Days: Actual Calendar days between the Project Start date and Project End Date . Measurements are based on elapsed calendar days without adjustment for weekends or holidays.

Project Elapsed Days = Project End Date - Project Start Date

Average Elapsed Time per Size Unit =

Sum of Project Elapsed Times

Sum of Project Sizes

Project Effort Categorization: The completed projects are grouped into the three effort categories (Small, Medium, Large) according to the criteria described in the following table.

Categories	SMALL	MEDIUM	LARGE
Development Effort	< 300 Hrs	300 - 1800 Hrs	> 1800 Hrs.

Appendix D Earned Value Management (Cost and Schedule)

Business Area: Cost and Schedule

Goal/Purpose:

The purpose of these indicators is to monitor contract performance for contracts that use Earned Value Management (EVM). This indicator will track the Cost Performance Index (CPI) and the Schedule Performance Index (SPI) in relation to the target values.

Questions:

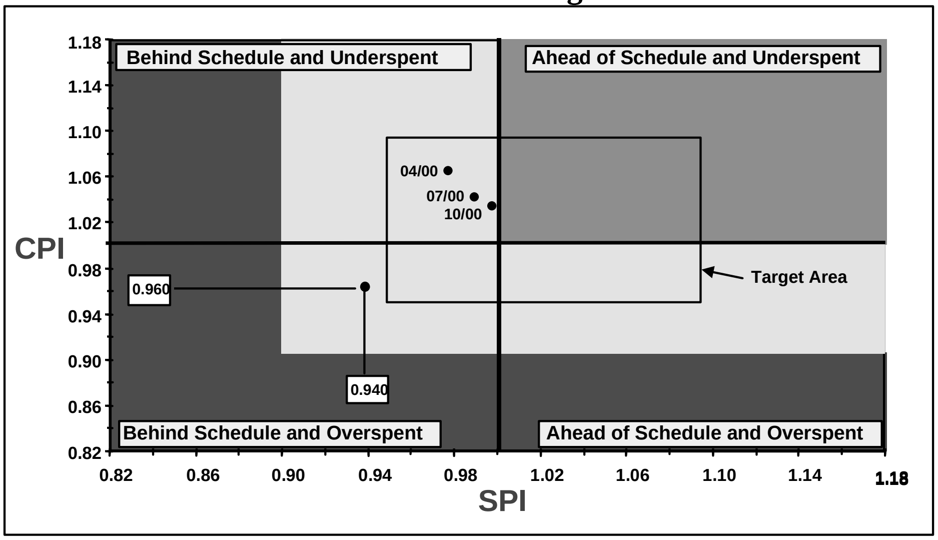
- Is the CPI and the SPI within their target area?

Perspective:

Project manager

Visual Display:

Earned Value Management



Where

Green	CPI ≥ 1.0	and	SPI ≥ 1.0
Yellow	CPI < 1.0	or	SPI ≤ 1.0
Red	CPI < 0.9	or	SPI < 0.9

Target Area

CPI = .95 to 1.08
SPI = .95 to 1.08

Input:

Data Elements:

CPI = Cost Performance Index = BCWP / ACWP

- BCWP: Budget Cost of Work Performed
- ACWP: Actual Cost of Work Performed

SPI = Schedule Performance Index = BCWP / BCWS

- BCWP: Budget Cost of Work Performed
- BCWS: Budget Cost of Work Scheduled

Source: TBD

Interpretation/Analysis

In the displays shown:

- In the display shown the CPI/SPI indices are well with in the target area.
- The trend of the three points is toward the center goal.

Open Issues

- The CPI and the SPI values for the target box must be determined and agreed upon.
- The CPI and the SPI values for the breakpoints for Green, Yellow, and Red must be agreed upon.
- Current status outside the "Target Area": When the current status falls outside the target area, specific actions (such as special reviews) must be defined.
- Specific actions, like special reviews, must also be defined for contracts in the Red area.

Notes

- This display makes it very easy to determine if a project is:
  - Behind Schedule and Underspent
  - Ahead of Schedule and Underspent
  - Behind Schedule and Overspend
  - Ahead of Schedule and Overspent

- The values for the breakpoints for Green, Yellow, and Red as well as the target box values for the CPI and SPI were taken from 5 Oct 2000 guidance briefing for reporting earned value on ACAT I & II program contracts.

- Typically contracts decline in performance over time, and many achieve the level of “Green” only initially or after rebaselining. So, the guidelines help highlight that we should not become unrealistically comfortable at a threshold of CPI and SPI $\geq$ 0.950.

- This indicator can be used two ways:
  - monitor the performance of a single project during development
  - monitor the performance of multiple projects by each point indicating a different project.

Appendix E Status of Software Engineering Processes

INDICATOR TITLE/DESIGNATOR: *Status of Software Engineering Processes*

As of Date: 17 May 03

GOAL OR TACTIC SUPPORTED: *Stabilize Engineering Processes*

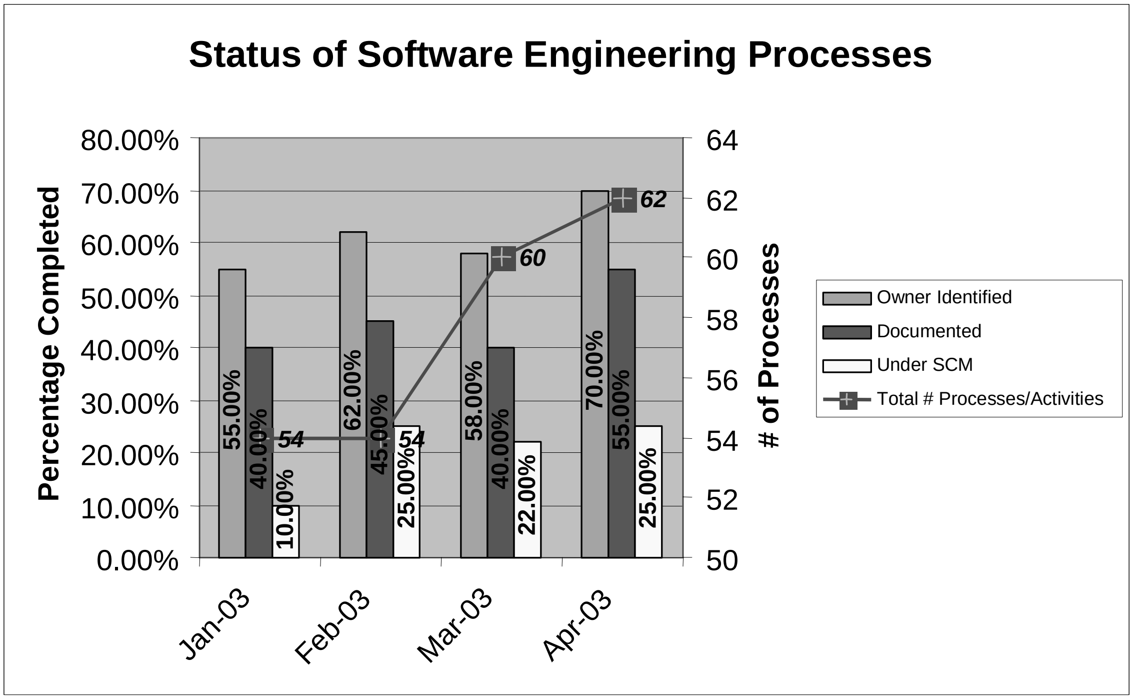
OBJECTIVE: To depict the stabilization efforts of the acquisition processes within {}

OWNER: Engineering Process Group

QUESTIONS:

- Are the software engineering processes stabilizing?
- How are they being stabilized?

VISUAL DISPLAY:



INPUTS:

Indicator Element	Definition
Total # Processes/Activities[3]	The total number of processes associated with the current development activities.
Owner Identified	The process owner has been established and assigned
Documented	The process has been documented and approved
Under SCM	The process is under configuration (change) control

DATA COLLECTON AND REPORTING:

How:	**Audit/Inspection of the processes**
Collection Frequency:	Quarterly
Collected By:	{org name}
Reporting Frequency:	Results will be reported quarterly

ALGORITHM:

Indicator Element	Definition
Owner Identified	(Number of processes with identified owners/Total number of processes/activities) * 100 (primary Y-axis)
Documented/Baselined	(Number of documented and baselined processes/Total number of processes/activities) * 100 (primary Y-axis)
Under SCM	(Number of processes under configuration control /Total number of processes/activities) * 100 (primary Y-axis)
Total # Processes/Activities	Count of all the processes/activities (2nd Y-axis)

ASSUMPTIONS:

- There are processes/activities being performed or mandated from other sources that are not necessarily accounted for
- Adequate resources (personnel, time, tools, etc.) are available to properly perform the process

ANALYSIS:

- As the processes are identified, documented, and training provided for the processes, the software engineering processes will be stabilized
- The application of stabilized processes will result in more predictability of the resulting software product

[3] Processes/Activities are based upon IEEE 12207.0, *Standard for Information Technology – Software Lifecycle Processes*.

PROBING QUESTIONS:

- What is the “enforcement” mechanism to ensure the processes are being followed?
- How are the deviations from the processes being identified and tracked?
- Are the documented processes “version controlled” (e.g. under SCM control)?
- What are the requirements for a fully documented process?
- What is the threshold for stability (100% is goal, but what is realistic)?
- What side effects, if any, have been caused by corrective efforts?

EVOLUTION:

- The quarterly reporting would be changed to semi-annual (or perhaps annual)
- The identified processes would change (new ones added, etc.) and the indicator would reflect a change as a result (new process areas added or activities added within the existing areas, etc.)
- {org} would most likely perform the long-term reporting. The frequency of the collection would also change as the situation changes.
- When processes are established, this indicator will need to be converted to procedural adherence (audits, appraisals)

CROSS-REFERENCES:

- Indicator SSEP 002: *Process Activity Training*

References

URLs are valid as of the publication date of this document.

[Basili & Weiss 84] Basili, V. & Weiss, D. “A Methodology for Collecting Valid Software Engineering Data.” *IEEE Transactions on Software Engineering 10*, 6 (November, 1984): 728-738.

[Basili & Rombach 88] Basili, Victor R. & Rombach, H. Dieter. “The TAME Project: Towards Improvement-Oriented Software Environments.” *IEEE Transactions on Software Engineering 14*, 6 (June 1988): 758-773.

[Basili 89] Basili, Victor R. “Using Measurement for Quality Control and Process Improvement.” *Proceedings of the Software Engineering Process Group Conference*. Pittsburgh, Pennsylvania, June 21-22, 1989. Pittsburgh, PA.: Software Engineering Institute, Carnegie Mellon University, 1989.

[Baumert & McWhinney 92] Baumert, John H. & McWhinney, Mark S. *Software Measures and the Capability Maturity Model*. (CMU/SEI-92-TR-25, ADA257238). Pittsburgh, PA.: Software Engineering Institute, Carnegie Mellon University, September 1992. http://www.sei.cmu.edu/publications/documents/92.reports /92.tr.019.html

[Crawford & Stephens 04] Crawford, Paul & Stephens, Mark. “Transitioning From Business Objective to Measurable Objective.” *Proceedings of the European SEPG 2004*. London, England, June 14-17, 2004. Milton Keynes, UK: European Software Process Improvement Foundation, 2004.

[CMMI 02] CMMI Product Team. *CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Continuous Representation* (CMMI-SE/SW/IPPD/SS, V1.1) (Continuous: CMU/SEI-2002-TR-011) and *Staged Representation* (CMU/SEI-2002-TR-012). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu/cmmi/models/models.html

[Goethert & Hayes 01]	Goethert, Wolfhart & Hayes, Will. *Experiences in Implementing Measurement Programs*. (CMU/SEI-2001-TN-026, ADA3399165). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, November 2001. http://www.sei.cmu.edu/publications/documents/01.reports /01tn026.html
[Park et al. 96]	Park, Robert; Goethert, Wolfhart; & Florac, William. *Goal-Driven Software Measurement – A Guidebook*. (CMU/SEI-96-HB-002, ADA313946). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, August 1996. http://www.sei.cmu.edu/publications/documents/96.reports /96.hb.002.html
[Raytheon 98]	Raytheon Metrics Team. *Metrics: A Management Guide for the Development and Deployment of Strategic Metrics*. Dallas, TX: Raytheon Corporation, 1998.
[Rombach 89]	Rombach, H. Dieter & Ulery, Bradford T. “Improving Software Maintenance Through Measurement.” *Proceedings of the IEEE 77,* 4 (April 1989): 581-595.
[van Solingen & Berghout 99]	van Solingen, Rini & Berghout, Egon. *The Goal/Question/Metric Method, A Practical Guide for Quality Improvement of Software Development*. London, England: McGraw-Hill International (UK) Limited, ISBN 007 709553 7, 1999.
[Zubrow 98]	Zubrow, David. “Measurement With a Focus: Goal-Driven Software Measurement?” *CrossTalk 11,* 9 (September 1998): 24-26.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. **AGENCY USE ONLY**
(Leave Blank)

2. **REPORT DATE**
September 2004

3. **REPORT TYPE AND DATES COVERED**
Final

4. **TITLE AND SUBTITLE**
Applications of the Indicator Template for Measurement and Analysis

5. **FUNDING NUMBERS**
F19628-00-C-0003

6. **AUTHOR(S)**
Wolfhart Goethert, Jeannine Siviy

7. **PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. **PERFORMING ORGANIZATION REPORT NUMBER**
CMU/SEI-2004-TN-024

9. **SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. **SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. **SUPPLEMENTARY NOTES**

12A **DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified/Unlimited, DTIC, NTIS

12B **DISTRIBUTION CODE**

13. **ABSTRACT (MAXIMUM 200 WORDS)**

Organizations often do not achieve the potential benefits of a sound measurement program due to the inconsistent construction and interpretation of indicators derived from measurement data. This technical note presents guidance for adapting and completing an *indicator template*—a tool the Carnegie Mellon® Software Engineering Institute (SEI) has developed to precisely describe an indicator—including its construction, correct interpretation, and how it can be utilized to direct data collection and presentation and measurement and analysis processes. An indicator template can help an organization to define *indicators*, or graphical representations of measurement data, which describe the who, what, where, when, why, and how for analyzing and collecting measures. This technical note defines each field of the indicator template, provides example inputs, and shows how the template may be used in the context of a process improvement effort that uses the SEI's Capability Maturity Model® Integration framework and/or Goal-Driven Software Measurement.

14. **SUBJECT TERMS**
Indicator template, Goal-Question-(Indicator)-Measurement, GQ(I)M, GQM, indicator, Goal-Driven Software Measurement, measurement and analysis, CMMI

15. **NUMBER OF PAGES**
45

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