Implementing Goal-Driven Measurement

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Overview
Outline

Benefits of Measurement

Goal-Driven Measurement

Example –Enterprise Wide Metric Program
Measurements Are Used for Many Purposes

Aggregate Data
- corporate/industry comparisons
- business decisions

Manage Projects
- plan
- track

Describe Products
- qualify
- classify

Improve Processes
- understand
- control

Baseline Measurement Process
- communicate clearly
- use process consistently
Measurement & Process Improvement

Software measurement programs are best initiated as part of a process improvement program that

• clarifies measurement’s purpose and role
• clarifies which measures to collect
• provides a mechanism for acting on findings
• involves the technical staff at all levels

Process improvement must have a measurement component that

• focuses on program objectives
• evaluates process effectiveness
• determines program benefits
Benefits

Measurement **does not by itself improve the process**; however, the visibility it gives **provides insight** into planning, controlling, managing, and improving. For example:

- historical data help us **predict and plan**
- actual versus plan data helps us **communicate progress** and **support decision making**
- analyzing trends helps us **focus on problem areas**
- **monitoring process improvement activities** helps us identify what works and what does not
### Benefit of Measurement

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Design</th>
<th>Coding</th>
<th>Functional Test</th>
<th>System Test</th>
<th>Field Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>40%</td>
<td>50%</td>
<td>~1.KDM</td>
<td>~1.KDM</td>
<td>~1.KDM</td>
</tr>
<tr>
<td>3%</td>
<td>5%</td>
<td>7%</td>
<td>~1.KDM</td>
<td>~1.KDM</td>
<td>~1.KDM</td>
</tr>
<tr>
<td>~1.KDM</td>
<td>~1.KDM</td>
<td>~1.KDM</td>
<td>~6.KDM</td>
<td>~12.KDM</td>
<td>~20.KDM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>~6.KDM</td>
<td>~12.KDM</td>
<td>~20.KDM</td>
</tr>
</tbody>
</table>

- **Fault Origin**: 10%
- **Fault Detection**: 5%
- **Cost per Fault**: 25%
- **KDM** = Kilo-Deutsch Marks

Data provided by Dr. Paulish of Siemens from “Engineering an Effective Software Measurement Program”
Goal-Driven Measurement

When using goal-driven measurement, the primary question is not:

“What indicators or measures should I use?”

rather, it is:

“What do I want to know or learn?”

Goal-driven measurement is not based on a predefined set of measures and Indicators.
Goal-Driven Measurement

- **Goal(s)**
  - Questions
    - Indicators
      - Measures
        - **Goal(s)**
          - Question 1
          - Question 2
          - …
          - Question n

- **Reporting Periods**
  - Total SLOC
    - Planned
      - Actual
        - Trouble Reports
          - Module
            - Number
              - Weeks

- **Size**
- **Effort**
- …
- **Defects**
- **Milestone Dates**
What Are Indicators?

An indicator is usually a graph or table that you define for your program needs.

An indicator is a representation of a measure or group of measures that provides information about a project issue.

Indicators frequently make comparisons between two values, such as planned and actual values.

An indicator is a measure or combination of measures that provides insight into the software process, a project, or the product itself.
Using Indicators to Identify Measures

When you are trying to gain insight into an issue or question, it is sometimes easier to think of how data would be displayed to identify the measures you need.

- A graph is easy to “think about” and “talk to” when you are getting input from others.
GQ(I)M Puts Each Measure in the Context of Addressing Goal-Related Questions

Measurement  
Goal 1  
Question 1  
Indicator 1  
Measure 1

Measurement  
Goal 2  
Question 2  
Indicator 2  
Measure 2

...  

Measurement  
Goal n  
Question m  
Indicator j  
Measure k
Why Start with Goals? Why Not with Measures?

Scenario

Organization “A” is collecting the following information

- **Staff (full time equivalent)**
  - total (actual, planned) per month

- **Defects**
  - severity, length of time open

- **Requirements**
  - total, TBDs, cumulative changes by month

- **Test Cases**
  - number passed, in analysis, not run

*Is this the “right” stuff to collect?*

What questions are these measures trying to answer?
Are These the Right Charts to Answer Your Questions?

**Staffing Levels**

As of Jul 20

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planned</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Build 1**
- **Build 2**

**Number of Defects**

<table>
<thead>
<tr>
<th>Severity Levels</th>
<th>x &lt; 30</th>
<th>30 ≤ x &lt; 60</th>
<th>60 ≤ x &lt; 90</th>
<th>x ≥ 90</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity 1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Severity 2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Severity 3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>Severity 4</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Severity 5</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td><strong>20</strong></td>
<td><strong>3</strong></td>
<td><strong>8</strong></td>
<td><strong>6</strong></td>
<td><strong>47</strong></td>
</tr>
</tbody>
</table>

As of 31 Jan 00

**Requirements**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cumulative Changes</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBD</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Now</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Test Case Completion by Function**

*Status as of: 2/24/00*

<table>
<thead>
<tr>
<th>Test Cases Complete</th>
<th>Test Cases</th>
<th>Funct 1</th>
<th>Funct 2</th>
<th>Funct 3</th>
<th>Funct 4</th>
<th>Funct 5</th>
<th>Funct 6</th>
<th>Funct 7</th>
<th>Funct 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass</td>
<td>2</td>
<td>6</td>
<td>12</td>
<td>14</td>
<td>7</td>
<td>69</td>
<td>12</td>
<td>4</td>
<td>34</td>
</tr>
<tr>
<td>Not Run</td>
<td>6</td>
<td>14</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>22</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Fail</td>
<td>14</td>
<td>14</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>22</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

As of 31 Jan 00

© 2008 Carnegie Mellon University
## Goal-Driven Measurement

<table>
<thead>
<tr>
<th>If the Goal was</th>
<th>Improve the development process.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Typical Questions</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Staffing:</strong> Are the right type of staff members assigned to this project?</td>
<td></td>
</tr>
<tr>
<td><strong>Defects:</strong> How many and of what type of defects are our customers seeing?</td>
<td></td>
</tr>
<tr>
<td><strong>Requirements:</strong> Are the requirements fully defined on time?</td>
<td></td>
</tr>
<tr>
<td><strong>Testing:</strong> Have improvements in the development processes decreased the amount of testing?</td>
<td></td>
</tr>
</tbody>
</table>

Indicators are not addressing these questions!

The bottom line ——— Have to know goal (why) before deciding what to collect.
Measurement Clarifies and Focuses Understanding of Issues

...but it all starts with

GOALS
SMART Goals

Well-defined goals have five common characteristics. They are:

- **Specific**: Clear and concise statement of what will be accomplished
- **Measurable**: Concrete, observable statement of what will be different once the goal is achieved
- **Attainable**: Feasible in terms of time, cost, and the degree of “stretch” the goal presents
- **Relevant**: Tied to a specific element of an organization’s goals
- **Time-bound**: Include a specific date for achievement
Measuring Goal Achievement

Goal → Success criteria

Strategy to accomplish goal

Tasks to accomplish goal

Task 1
Task 2
Task 3
... Task n

Success indicators

Impact

Progress indicators

Analysis indicators

Customer Detected Defects

Percent Towards Threshold

0 5 10 15 20 25 30 35 40 45 50

Proj. A
Proj. B
Proj. C
Proj. D
Proj. E

Completed Projects

% Slip

Schedule Slip

% Slip

Completed Projects

For project manager

Roll-up for higher management

© 2008 Carnegie Mellon University
Example

Goal

Establish the leading peach orchard in Georgia by selling more peaches than my competitors.

Success criteria

Greater market share than my competitors

Strategy

• Discover what motivates peach buying.
• Boost productivity by planting more of the most desirable variety of peach trees.
• etc.

Tasks

• Establish baseline.
• Survey which peach type most popular.
• Determine peach planting plan.
• Plant trees.
• Better marketing.
• etc.

Analysis indicators

Preference

Success indicators

Market share

Reporting Periods

Preference

Progress indicators

Status

Task 1
Task 2
Task 3
Task 4
Measures to Construct Indicators

- **Goal**
- **Strategy to accomplish goal**
- **Success criteria**
- **Success indicators**
  - Reporting Periods
  - %
- **Tasks to accomplish goal**
  - Task 1
  - Task 2
  - Task 3
  - Task n
- **Analysis indicators**
- **Progress indicators**
  - For project manager
  - Roll-up for higher management
  - MEASURES
Measures Address Multiple Indicators

Required Data Elements (measurements)

Cross Reference Matrix

<table>
<thead>
<tr>
<th>Indicators</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effort</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Defects</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Analysis</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Availability

Source

Success

Progress

Analysis

Definition Checklist

QA

CM

QA

CM

QA

CM

QA

CM

QA

CM

QA

CM

QA
Precise Operational Definitions

Precise operational definitions take into account the following

- **Communication** - will others know precisely what was measured, how it was measured, and what was included or excluded?

- **Reproducibility** - could others, armed with the definition, repeat the measurements and get essentially the same results?

- **Traceability** - are the origins of the data identified in terms of time, source, sequence, activity, product, status, environment, tools used, and collector?
SEI Core Measures

Provide checklists, guidelines, and examples for defining and reporting four kinds of measures that address fundamental management issues:

<table>
<thead>
<tr>
<th>Unit of measure</th>
<th>Issues addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Counts of physical source lines of code</td>
</tr>
<tr>
<td></td>
<td>Counts of logical source statements</td>
</tr>
<tr>
<td>Effort</td>
<td>Counts of staff-hours allocations</td>
</tr>
<tr>
<td>Schedule</td>
<td>Calendar dates tied to milestones, reviews and audits, and deliverable products</td>
</tr>
<tr>
<td>Quality</td>
<td>Counts of software problems and defects</td>
</tr>
</tbody>
</table>
### General Format of Checklist

**Identification Section**

<table>
<thead>
<tr>
<th>Attribute #1</th>
<th>Include</th>
<th>Exclude</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value N</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Attribute #2                  |         |         |          |

| Attribute #M                  |         |         |          |
Attribute 2: Hour Information

Attributes of “staff-hours”

Staff-hour

Regular Time

Salaried

Hourly

Overtime

Salaried

Hourly

Paid

Unpaid

Piad

Unpaid

What type of staff-hours are counted if I said that project “A” has expended 10,000 staff-hours?
### Example: Precise Definition
**Attribute 2 of Staff-Hour Definition Checklist**

<table>
<thead>
<tr>
<th>Hour Information</th>
<th>Totals include</th>
<th>Totals exclude</th>
<th>Report totals</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Regular time</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salaried</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Hourly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Overtime</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salaried</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Compensated (paid)</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uncompensated (unpaid)</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hourly</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compensated (paid)</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Uncompensated (unpaid)</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Ref: Software Effort & Schedule Measurement: A Framework for Counting Staff-hours and Reporting Schedule Information
http://www.sei.cmu.edu/publications/documents/92.reports/92.tr.021.html
Goal-Driven Measurement Workshop

**Goal(s)**
- Business => Sub-Goals => Measurement

**Questions**
- What do I want to know or learn?

**Indicators**
- SLOC
- Staff-hours
- Trouble Reports

**Definition**
- Checklist

**Infrastructure**
- Assessment

**INDICATOR TEMPLATE**
- Goal Questions
- Perspective
- Input/Measures
- Algorithm
- Assumptions
- Interpretation
Completed indicator templates are the output of GQ(I)M work.

Documents the why, what, who, when, where, and how.
Mapping CMMI Measurement and Analysis to Indicator Template

Measurement & Analysis

Indicator Template

- Date
- Indicator Name/Title
- Objective
- Questions
- Visual Display

SP 1.1 Establish Measurement Objectives

SP 2.2 Specify Measures

SP 2.3 Store Data & Results

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

SP 2.4 Communicate Results

Data Collection
- Input(s)
- Data Elements
  - Definitions
- How
- When/How Often
- By Whom
- Form(s)

Perspective

SP 2.1 Collect Data

SP 1.3 Specify Data Collection Procedures

SP 1.4 Specify Analysis Procedures

Data Elements

© 2008 Carnegie Mellon University
Goal-Driven Measurement Workshops

Management Team
- BUSINESS GOALS
- SUCCESS CRITERIA
- SUCCESS INDICATORS

Improvement Team
- Develop Strategies, Tasks
- Progress Indicators
- Analysis Indicators

Planning Task Matrix

<table>
<thead>
<tr>
<th>Planning Tasks</th>
<th>Data Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>Y N</td>
</tr>
<tr>
<td>Task 2</td>
<td>Y</td>
</tr>
<tr>
<td>Task 3</td>
<td></td>
</tr>
<tr>
<td>Task n</td>
<td></td>
</tr>
</tbody>
</table>

Data
- Elements:
  - Size
  - Defects
- Avail
  - +
  - 0
  - -
- Source:
  - QA
  - CM
  - ?
  - Etc.

Management
- Team Improvement Team
- SUCCESS CRITERIA
- BUSINESS GOALS
- SUCCESS INDICATORS

SUCCESS INDICATORS

SUCCESS CRITERIA

Develop Strategies, Tasks

Progress Indicators

Analysis Indicators
GQ(I)M Summary

The goal-driven software measurement method provides a well-defined process for identifying, defining, and implementing measurements.

Each organization must define measurements based on their own unique business goals.

Goal-driven software measurement is not based on predefined metrics.
Two Versions of Course/Workshop

Public Offering – Case Study

- Attendees from different organizations
- Case study to get everybody on same wavelength to illustrate the methodology

Tailored version – customer requirements/needs

- Customer specific goals
- Most requested
- Delivered at a customers site

Both delivered as a facilitated workshop
Outline

Benefits of Measurement

Goal-Driven Measurement

Example –Enterprise Wide Metric Program
Enterprise Wide Metrics Program
Objective

To establish an enterprise-wide metrics program that characterizes software progress and performance across a global enterprise.

To establish an initial, simple set of metrics that can be used across the enterprise to serve as the common “meter stick.”

To deploy this so that all organizations (maturity level 3 and higher) can use this program as part of their ongoing improvement efforts.
CTO’s Enterprise Process Improvement Goals

Increase Productivity by a factor of 2 over 5 years

Improve Quality by a factor of 10 over 7 years

Improve Predictability to within 5% over 7 years

Reduce Development time by 40% over 7 years

Reduce Maintenance effort by 40% over 7 years
Enterprise Metrics Program Participating Citicorp Sites
Selection of Indicators

Evaluation areas

• can indicator be interpreted correctly?
• does it provide an accurate and high-level view?
• could you collect the data in your organization?
• are there any major barriers?
• do the definitions provide enough information?

Other considerations

• number of indicators in each measurement area
• total number of indicators
Enterprise Initial Core Measures—1

**Schedule predictability**: Indicator designed to answer questions about the enterprise’s ability to plan well and deliver the products on schedule.

**Effort predictability**: Indicator designed to improve cost estimation and the ability to bring projects in on budget.

**Cycle time**: Indicator used to track improvements in getting products to market as quickly as possible.

**Quality**: Indicator for the quality of the development and testing process as well as the quality of the software in the field.

**Maintenance Effort**: Indicator used to track non-discretionary maintenance, enhancements, and defect corrections, as well as the number of open trouble reports.
Enterprise Initial Core Measures—2

**Customer satisfaction**: Indicator to track two components of customer satisfaction—satisfaction with the implemented solution and with the working relationship with the implementing team.

**Cost of quality**: Indicator that breaks overall costs (effort hours) into:
- **rework**—effort for fixing defects discovered prior to release
- **appraisal**—effort for inspection and testing
- **prevention**—effort incurred by process improvements aimed at preventing defects
- **performance**—effort associated with building the product
Enterprise Metrics

Schedule Predictability

Effort Predictability

Percent Deviation

Project Size:
- Small
- Medium
- Large

Cycle Time

Quality

Defect Density at UAT

Number of High Priority Field Defects

Cost of Quality:
- Rework
- Appraisal
- Prevention
- Performance

Cycle Time

Calendar Days per Size Unit

Defect Density

Number of High Priority Field Defects

Customer Satisfaction

Implemented Solution

Working Relationship

Maintenace Effort

Percent of Tech. Staff-Hours

Open Trouble Reports

Cost of Quality:
- COQ – Large Projects
- COQ – Medium Projects
- COQ – Small Projects
Indicator Template

Document the why, what, who, when, where, and how
Example of Indicator Template (Page 1)

Cycle Time

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

Questions
• What is the cycle time trend for each of the project size categories?
• Are the trends the same for the different project size categories?
• What is the rate of change from year to year?
• How does the rate of change compare between the different project size categories?

Indicator/Display

Calendar Days per Size Unit

<table>
<thead>
<tr>
<th>Project</th>
<th>Time Frame (Quarter)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>2000 1 2 3 4 1 2 3 4</td>
</tr>
<tr>
<td>Medium</td>
<td>2001 1 2 3 4</td>
</tr>
<tr>
<td>Large</td>
<td></td>
</tr>
</tbody>
</table>
Precise Definitions

Key dates—start and end times.

Project Phases

<table>
<thead>
<tr>
<th>Feasible Study</th>
<th>Alternative Analysis</th>
<th>Functional Specification</th>
<th>Design</th>
<th>Code &amp; Unit Test</th>
<th>Integration Test</th>
<th>UAT</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiation</td>
<td>Definition</td>
<td>Design</td>
<td>Build</td>
<td>Verification</td>
<td></td>
<td>UAT</td>
<td>Implementation</td>
</tr>
</tbody>
</table>

Project Start Date

Estimation Start Date

End Date (ship date)
### Staff-Hour Definition Checklist

<table>
<thead>
<tr>
<th>Activity</th>
<th>Totals include</th>
<th>Report totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary development activity</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Development support activities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concept demo/prototypes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tools development, acquisition, installation, &amp; support</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Nondelivered software &amp; test drivers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintenance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-discretionary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Defect correction (bug fixes)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regulatory/compliance</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Release upgrade</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Interface (external and internal)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Enhancements</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Legacy Systems</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Non-Legacy Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Employment Class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Citicorp employee</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full time</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Part time</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Temporary employee</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Subcontractors</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Consultants</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
Any Questions or Clarification?
# For Additional Information

<table>
<thead>
<tr>
<th>Name</th>
<th>Customer Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telephone</td>
<td>1-888-201-4479</td>
</tr>
<tr>
<td>FAX</td>
<td>412 / 268-5758</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:customer-relations@sei.cmu.edu">customer-relations@sei.cmu.edu</a></td>
</tr>
<tr>
<td>World Wide Web</td>
<td><a href="http://www.sei.cmu.edu">http://www.sei.cmu.edu</a></td>
</tr>
</tbody>
</table>
| U.S. Mail       | Software Engineering Institute  
                              Carnegie Mellon University  
                              Pittsburgh, PA 15213  USA |