

Carnegie Mellon University
Software Engineering Institute

1

AD-A280 915



DTIC
ELECTE
JUL 01 1994
S F D

Software Cost and Schedule Estimating:
A Process Improvement Initiative

Robert E. Park
Wolfhart B. Goethert
J. Todd Webb

May 1994

This document has been approved
for public release and sale; its
distribution is unlimited.

DTIC QUALITY INSPECTED 2

10090

94-20241



94 7 -1 031

Carnegie Mellon University does not discriminate on the basis of race, sex, age, national origin, ethnicity, religion, sexual orientation, or disability in any of its programs on the basis of race, ethnicity, national origin, or disability. The University has adopted the following policies to implement the Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973, which prohibit discrimination on the basis of race, ethnicity, national origin, or disability.

In addition, Carnegie Mellon University does not discriminate on the basis of sex, age, national origin, ethnicity, religion, sexual orientation, or disability in any of its programs on the basis of sex, age, national origin, ethnicity, religion, sexual orientation, or disability. The University has adopted the following policies to implement the Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973, which prohibit discrimination on the basis of sex, age, national origin, ethnicity, religion, sexual orientation, or disability.

Inquiries concerning application of these policies may be directed to the Office of the Vice President for Academic Affairs, Carnegie Mellon University, 15213 telephone (412) 268-6564. For information regarding the University's policies on sexual harassment, contact the Office of the Vice President for Academic Affairs, Carnegie Mellon University, 15213 telephone (412) 268-2055.

Special Report

CMU/SEI-94-SR-03

May 1994

Software Cost and Schedule Estimating: A Process Improvement Initiative



Robert E. Park

Wolfhart B. Goethert

J. Todd Webb

Software Process Measurement Project

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Approved for public release.
Distribution unlimited.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

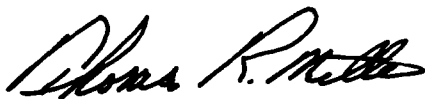
SEI Joint Program Office
HQ ESC/ENS
5 Eglin Street
Hanscom AFB, MA 01731-2116

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.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER



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

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

Copyright © 1994 by Carnegie Mellon University

This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at 52.227-7013.

This document is available through Research Access, Inc., 800 Vinial Street, Pittsburgh, PA 15212.
Phone: 1-800-685-6510. FAX: (412) 321-2994.

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: FDRA, Cameron Station, Alexandria, VA 22304-6145. Phone: (703) 274-7633.

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

Table of Contents

List of Figures	iii
List of Tables	iv
1. Introduction	1
1.1 Background	1
1.2 Objectives	1
1.3 Products	2
1.4 Plans	2
1.5 Opportunities for Leverage	4
1.6 Requirements for Success	4
1.7 Why this Report?	5
2. The Need for Improvement—A Preliminary Survey	7
2.1 Survey Questions	7
2.2 Survey Results	
Where Are Estimates Used?	9
How Well Are Estimates Meeting Needs?	11
Respondents' Roles with Respect to Estimating?	13
What Aspects of Estimating Seem to be Working Best?	15
What Improvements Would Be of Most Help?	17
What Emphasis Should the SEI Place on Improving Software Estimating?	21
How Would Respondents Like to Participate?	22
Who Should the SEI Contact?	22
3. Software Cost Estimating Improvement—The Plan of Attack	23
3.1 Principal Thrusts and Products	23
3.2 Future Thrusts	32
4. Process Modeling	33
4.1 Why Define Estimating Processes?	33
4.2 Methods for Describing or Defining a Cost Estimating Process—Some Considerations	34
4.3 Examples of Modeling Methods	36
EITVOX	36
IDEF0 and Design/IDEF®	41
State Charts and Statemate®	43
5. Estimating Process Templates—Some Preliminary Examples	45
5.1 Example 1: A Graphical Template for Parametric Estimating	45
5.2 Example 2: A Process Template for Bid-and-Proposal Environments	47
5.3 Example 3: Treating Estimating Processes as Process Assets	52
6. Schedule, Status, and Points of Contact	55
6.1 Schedule	55
6.2 Status	56
6.3 Points of Contact	56

References	57
Acronyms	59
Appendix A: Survey Form	61
Appendix B: Task Plan	67
Appendix C: Templates for Technical Collaboration Agreements	85

List of Figures

Figure 1-1	An Outline of Principal Thrusts and Planned Products	3
Figure 1-2	Investment of Resources by Participating Organizations	4
Figure 2-1	Government and Industry Uses of Software Estimates	10
Figure 2-2	How Well Are Software Estimates Meeting Needs?	11
Figure 2-3	Comparison of Today's Estimating Abilities to the Ideal	12
Figure 2-4	Principal Roles of Respondents	14
Figure 2-5	What Emphasis Should the SEI Place on Improving Software Estimating?	21
Figure 3-1	Principal Thrusts	23
Figure 4-1	The ETVX Paradigm	37
Figure 4-2	Using ETVX to Show the Stages of a Process	37
Figure 4-3	Using ETVX to Decompose a Process into Subprocesses	37
Figure 4-4	A Template for Process Description	39
Figure 4-5	IDEF Building Blocks	41
Figure 4-6	A Top-Level IDEF0 Representation of a Possible Estimating Process	42
Figure 5-1	Graphic Template for Parametric Estimating	45
Figure 5-2	Estimating as a Process Asset	53
Figure 6-1	Schedule as Proposed in Early 1993	55

List of Tables

Table 2-1	Uses of Software Estimates	9
Table 2-2	Principal Roles of Respondents	13
Table 2-3	What is Working Well?	15
Table 2-4	Improvements That Would Most Help	17
Table 2-5	Respondents' Suggestions for Things Organizations Could Be Doing Today to Improve Software Estimating	20
Table 2-6	People Interested in Working With the SEI to Improve Software Estimating	22
Table 5-1	Defining the Scope	48
Table 5-2	Technical Analysis	49
Table 5-3	Business Analysis	50
Table 5-4	Follow-Through	51
Table 6-1	Status of Technical Collaborations	56

Software Cost and Schedule Estimating: A Process Improvement Initiative

Abstract. This report describes efforts that have been initiated by the Software Engineering Institute to improve the practice of software cost and schedule estimating. These efforts involve support and participation from both industry and government. They are motivated by the Capability Maturity Model, which identifies the key roles estimating and cost management play in establishing repeatable software processes. Products from the initiative will include templates, criteria, and guidelines for establishing defined estimating processes, training materials, and examples for teaching good estimating practice, and evaluations of the abilities of contemporary cost models to meet today's estimating needs.

1. Introduction

1.1 Background

In 1993, the Software Engineering Institute (SEI) launched a joint industry-government initiative to improve the estimating capabilities of software organizations. The initiative addresses concerns about planning and cost management expressed by senior executives in both government and industry. It also supports the Capability Maturity Model, which describes the important roles that estimating and cost management play in establishing mature software processes [Paulk 93a and 93b].

1.2 Objectives

The initiative has three objectives:

1. To improve the abilities of industry and government organizations to estimate software sizes, costs, and schedules.
2. To provide criteria and processes for communicating verifiable software estimates, both within organizations and between contractors and customers.
3. To develop a capability at the SEI for helping organizations to improve their estimating processes. With this capability in place, organizations will be able to look to the SEI for constructive help in establishing the defined estimating processes that are needed to support the Capability Maturity Model.

1.3 Products

The software cost estimating improvement initiative will produce templates and guidelines for defining, implementing, and sustaining reliable estimating processes. Supporting (enabling) products will include process templates, process criteria, model criteria, cost model evaluations, training materials, and examples that can help software organizations define and make repeatable the processes they use for cost, schedule, and size estimating.

As by-products, the initiative will lay foundations for normalizing other quantitative metrics, so that performance results can be compared and contrasted across different organizations, processes, or time periods. We frequently forget that unnormalized comparisons are valid only when all other things are equal. Examples of metrics that must usually be normalized include

- productivity measures,
- cost and schedule risks,
- effectiveness of process improvement actions, and
- returns on investment.

Chapter 3 provides a more complete discussion of the products we expect the initiative to produce.

1.4 Plans

Our plan is to build on the SEI's ability to attract broad industry participation, and to use this participation to leverage investor resources. By working with strategic partners, sponsors, and technical collaborators—and with inputs and reviews from the software community at large—we will assemble, organize, desensitize, and disseminate summaries of successful practices gleaned from a number of proprietary estimating processes. In this way, many people will be able to benefit from the progress and lessons learned by others.

Figure 1-1 shows an outline of our plan. Our starting point is to work with our technical collaborators and sponsors to help them map (and benchmark) their existing estimating processes. Here we (and they) learn what today's capabilities really are. Mapping existing processes almost always identifies missing elements and inconsistencies that become immediate targets for process improvement. The organizations we work with then have early products they can use to help guide process improvement efforts. By observing and tracking these efforts, we will be able to assemble guidelines for success that can be shared with others.

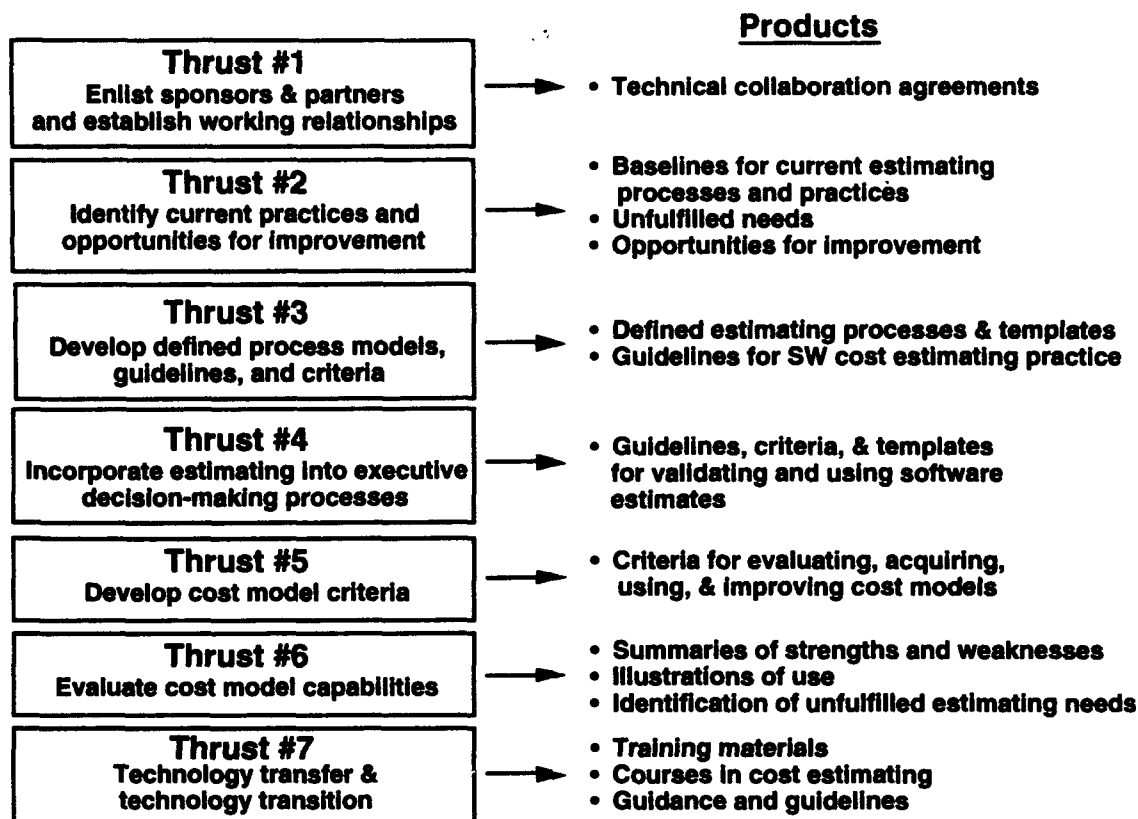


Figure 1-1 An Outline of Principal Thrusts and Planned Products

As we gain insights into the similarities and differences in estimating practices and in the improvement actions of collaborators and sponsors, we will use these insights to construct criteria, guidelines, and templates for good estimating practice. We will circulate these products for review and comment—first with our collaborators and sponsors, then for more widespread review—so that others can benefit from the experience and advice of the organizations who have offered to participate in this endeavor. We will use the comments we receive to revise and strengthen our products, and then we will publish the results so that they will be available to be used by the software community at large.

Accomplishing these objectives requires both deep probes and broad coverage. We will seek depth of knowledge by working closely with a small set of technical collaborators and sponsors, probing their processes and needs in detail, so that strong and weak points can be identified and root causes understood. We will seek breadth by using the insights that result from in-depth probes to prepare strawman templates, criteria, and guidelines, which we will circulate to broader audiences for review and comment.

Additional information about our task plan—principal thrusts, products, technology transfer, cost model evaluations, and cost model improvement actions—are discussed in Chapter 3. The full task plan is presented in Appendix B.

1.5 Opportunities for Leverage

This initiative provides sponsors and collaborators significant opportunities for leveraging their individual investments. Sponsors gain because their support brings them results not just from SEI resources, but also from the experience and efforts of industrial organizations and others who contribute to the initiative. Collaborators gain because they derive early benefits from lessons learned elsewhere in defining and improving software estimating processes. Moreover, sponsors and collaborators, by working with the SEI, are both in positions to help shape the direction of the initiative and ensure that it addresses needs that are important to their organizations.

Figure 1-2 shows the sources and levels of effort that we are currently anticipating. This is the picture as of May 1994. Loral Federal Systems and SETA Corporation have committed a combined total of 1.7 staff months per year to this work. Agreements with another two collaborators are in the review and approval cycle, and discussions are underway with a third organization. Other organizations, both federal and industrial, may yet join the initiative and add further resources.

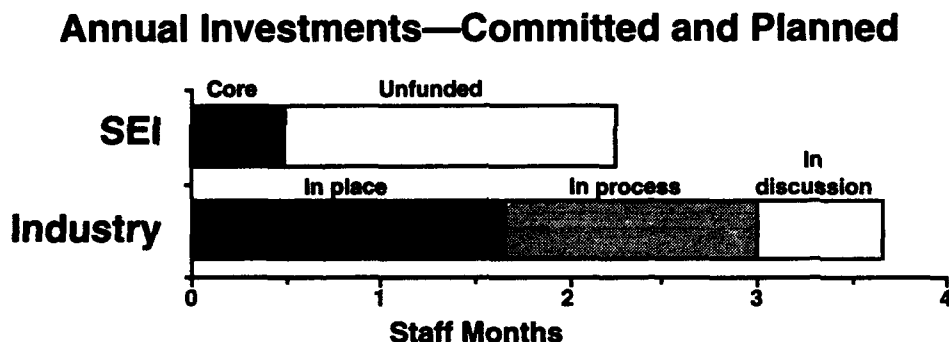


Figure 1-2 Investment of Resources by Participating Organizations

Appendix C contains a template for establishing technical collaboration agreements with the SEI. If you would like to explore the possibility of working with the SEI and others to improve your software estimating capabilities, please contact us at the address listed in Section 6.3.

1.6 Requirements for Success

Success in a process improvement initiative of this sort will depend heavily on active participation from collaborators and sponsors. This participation includes open doors, access to estimating processes and practices, and commitment of resources to process improvement efforts. To achieve its full potential, the initiative must have participants from both industry and government organizations.

Success also requires financial sponsorship. If we are unsuccessful in enlisting this sponsorship—and we don't have it yet—it is unlikely that the SEI will be able to sustain the initiative. Uncertainties concerning financial sponsorship are currently our greatest single risk to success.

1.7 Why this Report?

This report has been prepared to bring the software cost estimating improvement initiative to the attention of organizations that might like to participate. It also gives those that are participating a single reference where the most current information about the initiative is collected. And, although the initiative is just getting underway, there are things of substance to report.

The chapters that follow

- summarize the results of a survey that we conducted that looked at the needs people see for improvements in software cost estimating (Chapter 2),
- outline the principal thrusts and proposed products of the initiative (Chapter 3),
- discuss some of the methods we are exploring (Chapter 4),
- illustrate some tentative examples of process templates that have been suggested (Chapter 5), and
- report on our progress in enlisting technical collaborators and sponsors (Chapter 6).

We hope this information will help you judge whether you might like to join in the work we have planned.

2. The Need for Improvement—A Preliminary Survey

We prepared and distributed a preliminary survey to assess the need for improvements in cost estimating and to determine the importance that the SEI should assign to this work. The survey sought information about how well today's software estimating is meeting people's needs. It also asked people to identify the aspects of estimating that are working best for them and the improvements that are most needed in their organizations. In addition, the survey provided an opportunity for people to tell us how they might like to participate in the initiative and help shape its directions.

We distributed the survey to the following groups:

- Senior aerospace industry and government executives who participated in the 1990 SEI Executive Seminars.
- 400 attendees at the 1993 Software Engineering Symposium.
- Subscribers of the SEI.
- Current and former resident affiliates of the SEI.
- User groups associated with the COCOMO and SLIM cost models.
- The SEI Measurement Steering Committee.
- The 1994 MITRE Software Engineering Economics Conference.
- Government sponsors of SEI work.
- Other potential sponsors and technical collaborators.

As of December 31, 1993, we had received 249 responses. The views expressed by respondents are summarized in the sections that follow.

2.1 Survey Questions

We designed the survey so that each participant could answer the questions from a personal viewpoint, without having to call in the experts in his or her organization.

We asked the following questions:

1. Reflecting on what you have seen over the last two or three years, where are software estimates used in your organization?
2. Based on your personal observations, how well would you say that estimates for software cost, schedule, and size are meeting the needs of your organization?
3. What is your principal role with respect to software estimating?
4. In thinking about how you have seen software estimates produced, transmitted, or used, what aspects of estimating seem to be working best?
5. What improvements would most help you or your organization?

6. What emphasis should we at the SEI be placing on improving the processes and practices associated with software cost, schedule, and size estimating?
7. If you (or your organization) would like to work with the SEI in the software cost estimating improvement initiative, please indicate where your principal interests lie.
8. Who should the SEI contact to follow up on your interests?

We provided structured layouts such as check-boxes and thermometer scales for responding to questions 1, 2, 3, 6, and 7. Questions 4 and 5 asked for free-form responses, and question 8 asked for a name and address.

Appendix A contains a copy of the survey. If you would like to add your views to those we have already received, we welcome your insights and advice. Please copy this form and use it to send your comments to us.

2.2 Survey Results:

Where Are Estimates Used?

The first survey question sought information about where organizations are using software estimates. The question was

Reflecting on what you have seen over the last two or three years, where are software estimates used in your organization?

We provided checkboxes for twelve uses:

- Concept exploration
- Design evaluation
- Bid/no-bid decisions
- Proposal preparation
- Proposal evaluation
- Contract negotiation
- Project planning & scheduling
- Project tracking
- Project staffing
- Resource leveling
- Estimates to complete
- Replanning & rescheduling

We asked people to check all uses that apply to their organizations. We also provided two unlabeled (Other) checkboxes, so that responders could extend the list by identifying applications we may have missed.

Table 2-1 summarizes the results.

Uses	Gov't/Military	Industry	Academia	Unknown
Project planning & scheduling	64	150	3	3
Project staffing	43	122	0	3
Estimates to complete	46	114	2	4
Project preparation	36	115	2	3
Replanning & rescheduling	37	101	2	2
Project tracking	32	104	1	3
Contract negotiation	31	80	1	4
Proposal evaluation	43	64	0	2
Resource leveling	20	60	1	2
Concept exploration	25	54	2	3
Design evaluation	25	52	2	2
Bid/ no-bid decision	13	63	2	2
Other	17	11	2	1
Number of Respondents	81	159	4	5

Table 2-1 Uses of Software Estimates

Figure 2-1 is a graphical summary of the profiles of responses we received from government and industry respondents. These profiles suggest that patterns for using software estimates in government organizations are very similar to those in industry, except possibly for government's more frequent use of estimates in proposal evaluations.

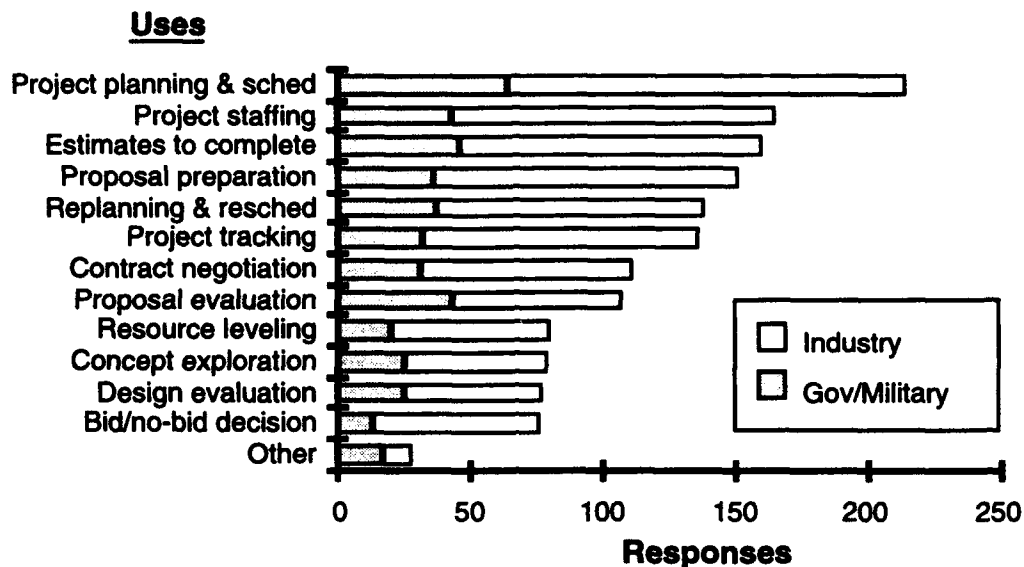


Figure 2-1 Government and Industry Uses of Software Estimates

Figure 2-1 also suggests something worthy of follow-up investigation: the use of estimates for continuing project management (project tracking, replanning & rescheduling, and estimates to complete) appears less wide-spread than the use of estimates for up-front actions (project planning & scheduling, staffing, and proposal preparation). This suggests that a number of organizations may not be tracking progress against plans.

How Well Are Estimates Meeting Needs?

The second question addressed the ability of today's estimates to meet user needs. The question was

Based on your personal observations, how well would you say that estimates for software cost, schedule, and size are meeting the needs of your organization?

We provided three thermometer scales so that respondents could register separate views for cost, schedule, and size.

The six panels of Figure 2-2 show the patterns of response from government organizations on the left and industry organizations on the right. The number of responses from each group is shown at the top of each panel. In general, government employees tended to report software estimating to be somewhat less successful in meeting needs than did respondents from industry. Whether this reflects insider's perceptions of the estimating capabilities of government agencies or of the usefulness of estimates received from industry is not presently clear.

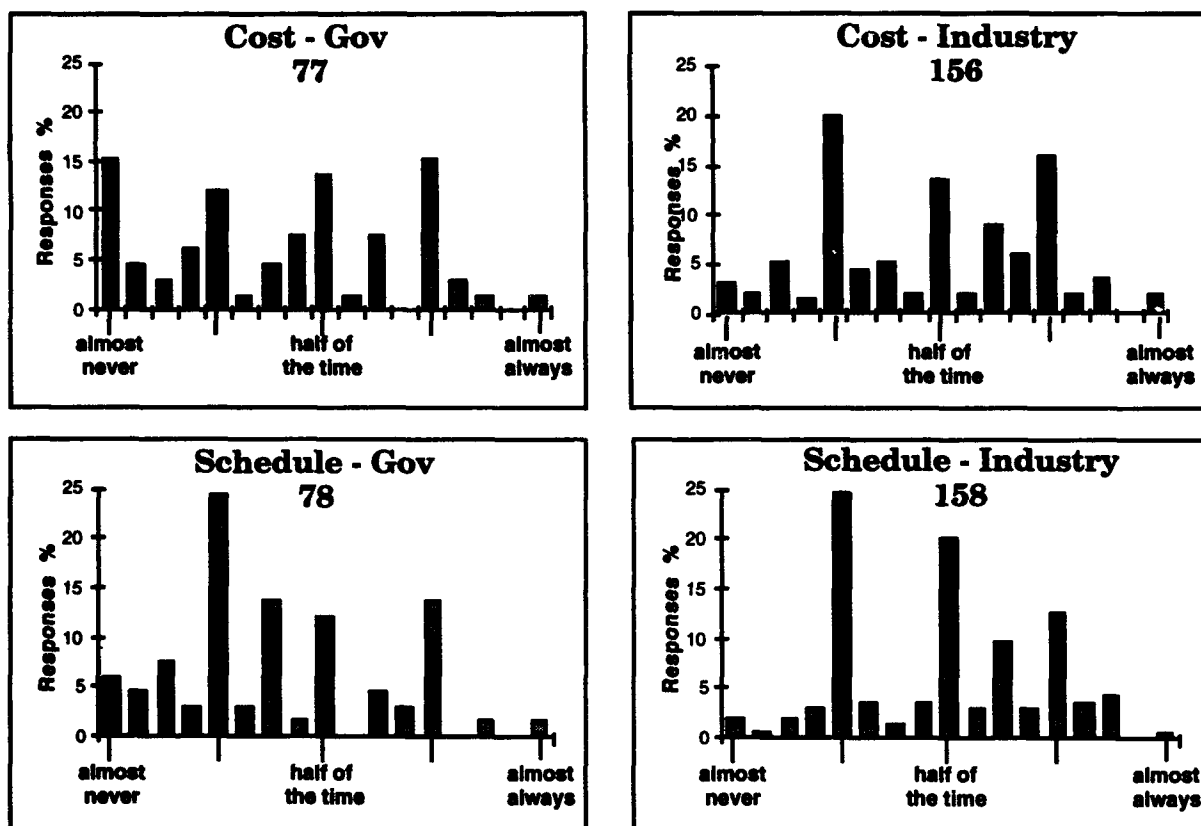


Figure 2-2 How Well Are Software Estimates Meeting Needs?
(Continued on next page)

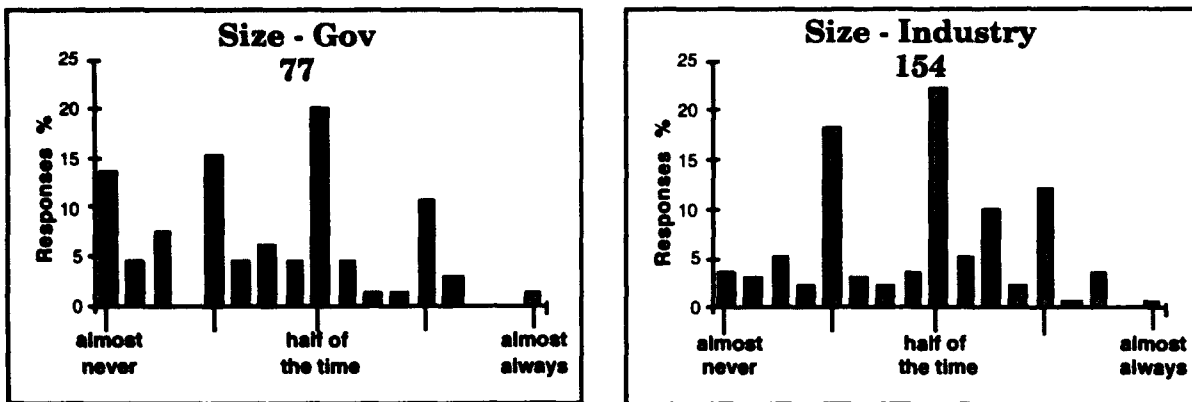


Figure 2-2 (Continued) How Well Are Software Estimates Meeting Needs?

Some caution should be used when interpreting the profiles in Figure 2-2, as the expanded vertical scale tends to mask the difference between the state of practice and the state most organizations would like to achieve. To put the results in an appropriate perspective, we compared them to the idealized response—100% almost always satisfied. Figure 2-3 shows these comparisons for the combined responses received from government and industry. The room for improvement is apparent.

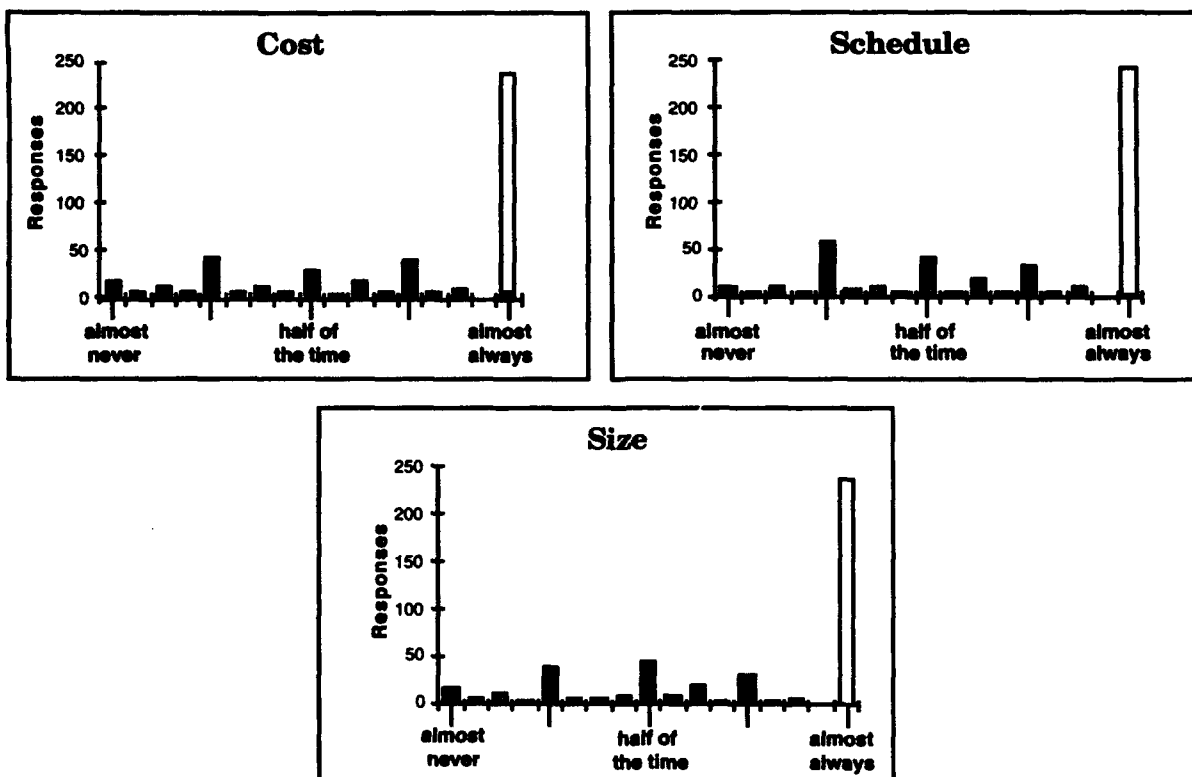


Figure 2-3 Comparison of Today's Estimating Abilities to the Ideal

Respondents' Roles with Respect to Estimating?

The third question in the survey asked about the functional roles of respondents. We wanted to classify responses to see, if possible, whether people with different roles hold different views or have different needs for improvements in estimating. The question was

What is your principal role with respect to software estimating?

A list of nine functional roles was supplied, and we asked respondents to check all that applied to them. Most respondents reported multiple roles. The responses are summarized in Table 2-2.

Roles	Gov't/Military	Industry	Academia	Unknown
Producer of estimates	43	114	3	4
Prog/proj planning & mgt	42	111	3	3
Process improvement	31	92	1	4
Proposal evaluator	31	43	1	1
Business/enterprise mgt	26	26	2	1
Consultant	10	31	0	1
Teacher, trainer	8	25	1	0
Other	15	8	1	0
Cost model developer	3	14	0	0
Number of Respondents	81	159	4	5

Table 2-2 Principal Roles of Respondents

The profiles of responses from government and industry respondents are shown graphically in Figure 2-4. These profiles are very similar, with the exception that the roles of proposal evaluator and business/enterprise manager occur with greater frequency in government than in industry.

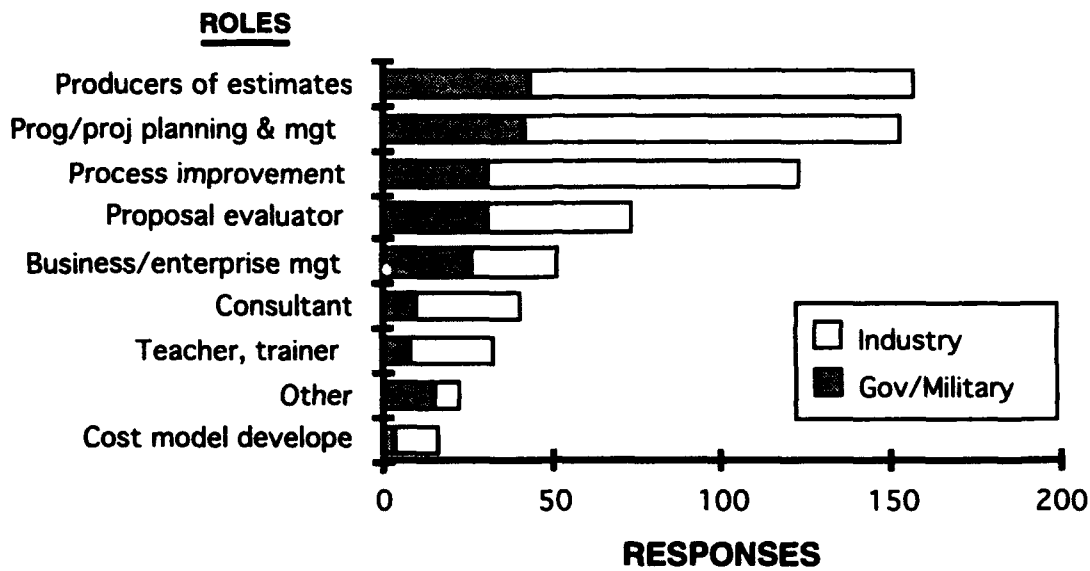


Figure 2-4 Principal Roles of Respondents

In preparing and distributing the survey, we had in mind three principal classes of people we wanted to survey:

- producers of estimates
- users of estimates (project managers, program managers, acquisition managers, etc.)
- enterprise managers (those responsible for ensuring that their organization has a reliable estimating capability, and that all projects and programs use it)

We sought sufficient responses from each class to determine if perceptions for the need for improvement in software estimating varied by class. The responses we received showed no discernible differences among these three classes. Producers of estimates, users of estimates, and enterprise managers all seem to have similar perceptions of the capabilities of today's software estimating practices.

What Aspects of Estimating Seem to be Working Best?

Question 4 sought to identify where potential strong points of estimating might lie. We asked

In thinking about how you have seen software estimates produced, transmitted, or used, what aspects of estimating seem to be working best?

We expected a variety of responses and did not want to constrain thinking, so we made this a free-form response. The results varied widely—even more than anticipated. Few patterns or recurring themes are evident, making effective summarization difficult. Table 2-3 illustrates some of the responses we received. We have attempted to group these responses according to principal themes.

Aspects of Estimating That Are Working Well		
Response Class	Community	
	Government & Military	Industry
Area we are best at	- LOC based on prototypes - determining resources required to perform development - being able to utilize projections with program office to determine what capabilities would be included in a specific build and what would be postponed to future developments	- estimates of overall effort - productivity predictions - schedule - cost/effort given size - time & resources given size - given end-date, what staff is needed to complete - estimating the activities to be done
Structured processes	- we have some structured processes in place, and smaller jobs seem to be more accurate and reliable than the larger ones	- follow a logical, structured process - objective & repeatable method - standard approach - top-down partitioning & decomposition, then bottom-up estimating - defined process plus tools - use of guidelines by task or process - use of checklist to assure all items are covered - consistent format

Continued...

Table 2-3 What is Working Well?

Aspects of Estimating That Are Working Well (Continued)		
Response Class	Community	
	Government & Military	Industry
Techniques	- using data from previous projects - basing estimates on historical data - functional decomposition rather than LOC estimation - if contractor is known, bottom-up estimates of software productivity work best - Delphi technique for on-going/similar projects - almost any method works if people just take the time to really scope out a project to determine what needs to be done	- using historical data - using prior experience - comparison to past, known experience - estimation by similar work or similar developments - use of detailed work breakdown structure and standard estimating forms - bottom-up approach - definitions of terminology
People	- estimates by experienced system analyst & program manager - best estimates are produced by trained, experienced users	- getting the right people to conduct estimating effort - increase involvement of people who will do the work - best estimates are produced by trained, experienced users - estimates developed at lowest possible level (by the folks who do the work)
Models	- use of models in planning stages - models give buyer and seller a common ground and a set of criteria to work from	- utilizing parametric models "calibrated" with historical data - estimating models based on empirical data obtained from a high number of projects - use of a common model with common understanding of parameters - any model used by an experienced software engineer estimator - using models to do what-if analysis

Table 2-3 (Continued) What is Working Well?

What Improvements Would Be of Most Help?

We also attempted to identify where the most likely targets for improvement might lie. We asked the following question:

What improvements would most help you or your organization?

The responses here were even more scattered. Apparently almost everyone sees a need to improve software estimating, but few see the same needs. This may indicate that root cause analyses are needed. If so, process mappings may be of even greater benefit than we have been anticipating.

Examples of the responses we received are shown in Table 2-4, grouped according to the general areas they address.

Improvements That Would Most Help		
General Areas	Community	
	Government & Military	Industry
Size	- sizing model: it would help to have better defined parameters for various environmental factors - get away from "source lines of code" as the metric for estimating cost, size, and schedule. Replace with estimates from development staff - use function/feature point count along with lines of code for sizing and metrics - identify scoping and estimation for system engineering activities	- better models and processes for software sizing - improvements in the ability to produce detailed and accurate estimates of schedule and size - sizing metrics & accuracy - estimating size at requirements and system architecture levels - methods for estimating size at end of requirements phase - size estimating procedures, models, and standards should be made available - size estimating model that is simple to use Continued...

Table 2-4 Improvements That Would Most Help

Improvements That Would Most Help (Continued)		
	Community	
General Areas	Government & Military	Industry
Models	- a turnkey system for creating estimates - one model that covers all aspects of acquisition process, with inputs for support organization costs. Thus, an entire cost estimate can be performed within a model - industry-wide standard definitions of input parameters to the various estimating models	- good project planning/tracking tools that allow an integrated approach and provide statistics for future estimates - develop a consistent software estimating model and use it consistently throughout the organization - a software tool that bases its estimates on a database of metrics
Model Characteristics	- sizing methods for 4GL's (or tools provided by 4GL vendors) which equate 4GL size/development times to those of traditional development approaches - better ability to tailor models to different development processes - dealing with new technologies and methodologies. Examples: object-oriented software development effects of CASE tools the C language	- estimating models that specifically address I-CASE development (with code generators) and other tools that offer productivity enhancements such as screen painters and the like - good case model with Ada project data. Also models for extremely large systems (e.g., > 2.5 million source lines of code) - better alignment of tools to the processes of the CMM - a better, more dynamic model of different life cycles
Databases	- historical data - capability to capture actual data and recalibrate models for organization use - better methods to track and capture actual data for later comparison with estimates - development of database of specific parameters for use in project management/metrics	- better historical data to assist in size estimation - cost estimating database for comparative purposes - a framework for determining: what data to collect how to build models to use the data - analytical tools to convert metrics databases to calibration inputs
Metrics	- having standardized, well-defined methods for counting source lines of code - consistent definition of lines of code (or task) - records of past and current software efforts and results (i.e., maintenance of metrics)	- improve the metrics collection process - setting up an effective metrics program (process metrics) - lack of standard measurement definitions and terminology causes significant problems

Continued...

Table 2-4 (Continued) Improvements That Would Most Help

Improvements That Would Most Help (Continued)		
General Areas	Community	
	Government & Military	Industry
Process	- a standard process. All the tools in the world will not help you if you use them incorrectly - a unified and common process to estimate software maintenance costs - ability to accurately and consistently estimate cost/schedule and effort required for software maintenance and enhancements - a work breakdown structure for cost estimating - having a well-defined process in place that would provide feedback to the estimator at specified milestones during development - a process that would produce two independent estimates which could be compared - realistic, formal policies on software development processes - a process rather than a tool - senior management adherence to estimates and processes	- checklists and performance standards for each phase so that there is a consistent repeatable approach to estimating - well-defined process that would provide feedback at specified milestones during development - having standard estimating templates available, possibly with representative work breakdown structures for small, medium, and large projects - sound practices and procedures, tied to clear standards and tools, with methodology for implementation - an integrated approach that addresses size estimation, tracking, and accounting systems with enough detail to track time and effort throughout development phases on a unit or module basis, and historic data that one can use on future estimates - formalized prescriptive on what needs to be included in effective cost estimating techniques - ways to express the uncertainty in the estimated values (due to imperfect knowledge and potential risk) - integration of plans/estimates with project management

Table 2-4 (Continued) Improvements That Would Most Help

Some respondents went further than the comments summarized in Table 2-4, identifying things that organizations could be doing today to improve software estimating. These suggestions are summarized in Table 2-5.

Respondents' Suggestions for Things Organizations Could Be Doing Today to Improve Their Software Estimating	
Government & Military	Industry
- maintain a comprehensive database of historical metrics/definitions - make {cost, schedule, performance, metrics, measurement} a routine management tool - management must require estimates for a useful purpose - train future software cost estimators - our organization should develop a policy that dictates standard usage of estimating process, methods, and tools. Due to lack of policy, estimating is at the discretion of whoever is "in charge" of the proposal, project, program, etc.	- capture data during and after the project - more formal analysis of project data collected - track actual hours worked vs. the 40-hour work week - build a documented, realistic database of diverse results garnered from projects to be applied at the proposal/project start-up phases - perform follow-up data collection to assess the accuracy of the software estimates generated - set up an effective metrics program (process metrics) - project managers should define the metrics and data most useful to them for project planning and management

Table 2-5 Respondents' Suggestions for Things Organizations Could Be Doing Today to Improve Software Estimating

Question 6 sought advice that would help the SEI and its sponsors assess the priority they should assign to software cost estimating improvement. The question was

Figure 2-5 shows the combined responses from all respondents. The desire for having the SEI do work in this area is clear.

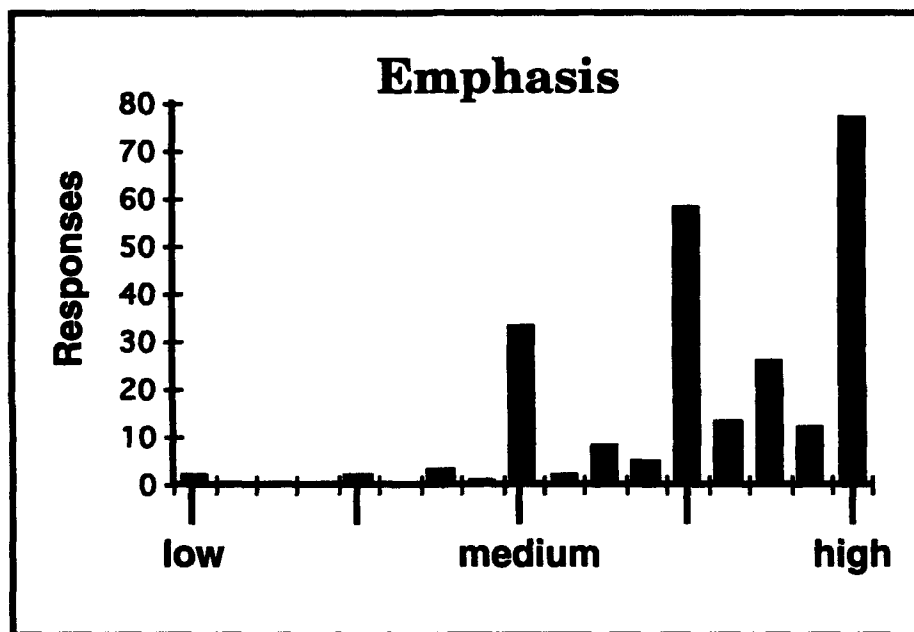


Figure 2-5 What Emphasis Should the SEI Place on Improving Software Estimating?

How Would Respondents Like to Participate?

We also used the survey to help us identify people and organizations who would like to participate in the initiative. We asked

If you (or your organization) would like to work with the SEI in the software cost estimating improvement initiative, please indicate where your principal interests lie:

We listed five ways to participate, and we provided checkboxes for two levels of interest for each. The results are most encouraging. They suggest that the initiative has many supporters, and that it will have many resources to draw on. The prospects are good that for relatively small investments, participants and sponsors can expect substantial returns.

The responses to question 7 are tabulated in Table 2-6.

Principal interests	Number interested	No. highly interested
Providing examples of estimating processes and practices	56	55
Providing access to people, processes, and data	61	41
Reviewing draft products	64	123
Working actively with the SEI to improve software cost, schedule, and size estimating	51	74
Becoming a technical partner or sponsor	35	33

Table 2-6 People Interested in Working With the SEI to Improve Software Estimating

Who Should the SEI Contact?

The final question on the survey asked for the respondent's name and organization, and for the person the SEI should contact to follow up on the interests expressed.

We will use this information to help us identify potential collaborators and sponsors, and to construct mailing lists of reviewers for the products we develop. If you expressed interest in working with the SEI on cost estimating improvement, you can expect to receive materials for review as soon as they are ready.

3. Software Cost Estimating Improvement—The Plan of Attack

3.1 Principal Thrusts and Products

Our plan for software cost estimating improvement has seven principal thrusts. Figure 3-1 shows the names of the thrusts and how information developed in each flows to support others.

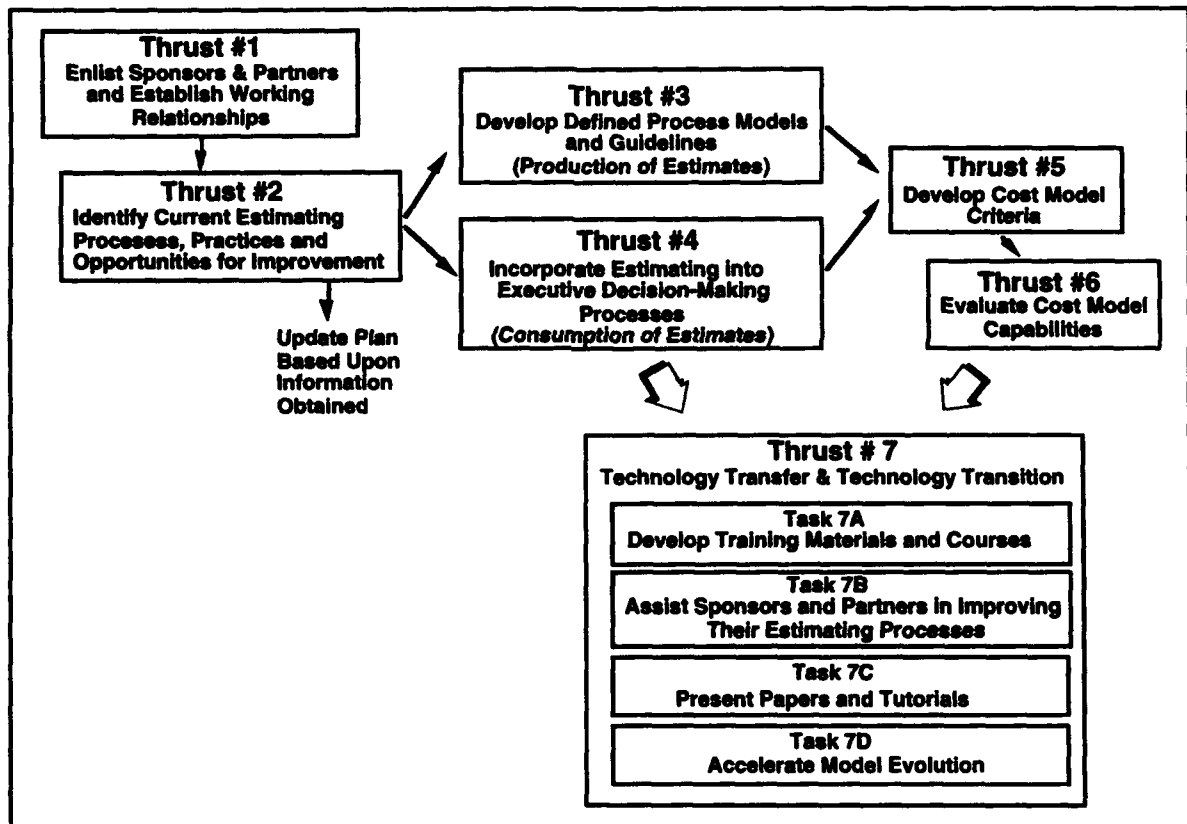


Figure 3-1 Principal Thrusts

Each of the thrusts has a purpose, tasks to be performed, and products to be produced. The pages that follow describe these elements.

Thrust 1: Enlist Sponsors & Partners and Establish Working Relationships

Purpose Thrust 1 encompasses the planning, organizing, and recruiting actions that are essential to successfully launching and sustaining an effective initiative. It includes the administrative and marketing efforts that are directed toward enlisting sponsors and establishing the collaboration agreements and working relationships that will provide funding and technical support for the initiative.

Tasks

- Ensure that industry and government organizations are informed of the initiative and have an opportunity to volunteer to take part.
- Establish technical collaboration agreements and confidential disclosure agreements with participating organizations.
- Secure funding to support the SEI portion of the initiative efforts.
- Recruit resident affiliates to support the initiative and ensure that both their home organizations and the SEI receive full benefit from their work.

Products

- Briefings and presentations to government leaders and prospective participants.
- Task plans and schedules.
- Funding support.
- Technical collaboration agreements.
- Confidential disclosure agreements.
- Resident affiliates to work on the initiative.

Thrust 2: Identify Current Estimating Processes, Practices, and Opportunities for Improvement

Purpose	As sponsors are found and technical collaboration agreements established, the SEI will work with these organizations to help them identify and map the processes they employ today for producing and using software estimates. The first objectives are to understand the existing processes and map them to the needs they seek to meet, to identify the elements that seem to be working best, and to find starting places for improvement. The secondary objectives are to assemble benchmarking information that will enable sponsors, collaborators, and others to evaluate their estimating capabilities relative to the capabilities of others in industry and government.
Tasks	• Identify and define the estimating processes and practices that are used today to support planning, evaluating, and controlling software acquisition and development. • Use these defined processes to identify best practices, unfilled needs, and opportunities for improvement. • Provide this information to sponsors and technical collaborators, so that they can make early use of it in process improvement and benchmarking activities. • Assemble the information in generic, nonproprietary forms. • Publish the results, so that they can be used throughout the software community.
Products	• Baseline descriptions of software estimating processes, practices, and tools that organizations use today for developing and acquiring software systems. • Lists of unfilled estimating needs and opportunities for improvement. • Feedback that participating organizations can use to guide process improvement efforts.

Thrust 3: Develop Defined Process Models and Guidelines for Estimating Practice

Purpose Thrust 3 will use the practices identified in Thrust 2 as a foundation for developing process models, criteria, and guidelines for establishing and sustaining reliable software estimating.

Tasks

- Develop templates for defined software estimating processes.
- Support these templates by developing and publishing guidelines and criteria for implementing and sustaining an effective estimating capability.

Products

- Criteria for effective estimating.
- Templates and defined process models for producing verifiable and repeatable software estimates.
- Guidelines and methods for comparing software costs across different contractors, work sites, and development organizations.
- Guidelines for initiating and sustaining improved software estimating capabilities.
- Guidelines for incorporating feedback from project tracking into plan revisions.
- Guidelines for incorporating feedback from project tracking into estimates for other projects.
- Criteria and methods for effectively communicating and interpreting estimation results.

Thrust 4: Incorporate Estimating into Executive Decision-Making Processes

Purpose Thrust 4 is directed toward improving the use of software estimates. The goal is to integrate software estimating into executive decision-making processes.

Where Thrust 3 deals with producing estimates, this thrust deals with their consumption and use. Our hypothesis is that many reasonable estimates fail today because of ineffective communication. Users of estimates often have so little insight into the assumptions and methods used to produce estimates that they cannot trust the results. Many estimates also hit the wrong targets because the initial information provided to estimators is flawed and incomplete.

The purpose of this thrust is to provide criteria and structured methods that organizations can use to obtain and validate software estimates, and then to apply them effectively as quantitative baselines for planning and managing software activities.

Tasks Develop guidelines, criteria, process templates, and evaluation methods that can help managers and acquisition organizations improve their abilities to obtain, validate, and use estimates when planning and managing software efforts.

Products

- Guidelines for obtaining reliable estimates from developers and estimators.
- Guidelines and criteria for understanding and validating software estimates.
- Guidelines and criteria for using software estimates in
 - bid and proposal activities,
 - project planning,
 - project tracking,
 - business-area planning, and
 - strategic planning.
- Papers and tutorials that show how to
 - communicate and interpret estimates effectively,
 - understand and validate software estimates,
 - use software estimating to improve management practice, and
 - use software estimating to avoid unpleasant surprises.

Thrust 5: Develop Cost Model Criteria

- Purpose** Thrusts 5 applies the criteria and processes developed in Thrusts 3 and 4 to develop criteria that will help organizations use (and choose among) the software cost models that exist today.
- Tasks** Develop criteria and examples for evaluating, selecting, and using software cost models.
- Products** • Criteria for evaluating and acquiring software estimating models and tools.
 • Criteria for using software-specific estimating models and tools.
 • Criteria for improving upon today's software estimating models and tools.

Thrust 6: Evaluate Cost Model Capabilities

Purpose Thrust 6 applies the criteria developed in Thrust 5 to evaluate the capabilities of existing software cost, schedule, and size models.

Tasks

- Evaluate the capabilities of existing software cost, schedule, and size models and show how these models can best be used to support the process guidelines identified in Thrusts 3 and 4.
- Structure the evaluations so that they provide advice and guidance for organizations operating at different maturity levels and in different development environments.
- Identify the most pressing needs that are not met by today's cost, schedule, and sizing models. Provide this information to cost model vendors, and assist them in using the information to develop improvements to their estimating tools.

Products

- Evaluations of cost model capabilities.
- Summaries of the strengths and weaknesses of existing software estimating models and tools, in light of how well they help organizations execute the processes identified in Thrusts 3 and 4.
- Guidelines and examples for obtaining reliable estimates with existing software cost models and tools.
- Lists of needs that are poorly met by today's cost models and tools (opportunities for improvement for commercial model vendors).
- Guidelines and recommendations for acquiring and installing effective suites of software estimating tools.

Thrust 7: Technology Transfer and Technology Transition

Purpose The efforts in Thrust 7 parallel those in Thrusts 3-6. As soon as the SEI has constructive methods, criteria, and processes to offer, we will begin working directly with collaborators, sponsors, cost model vendors, and others to implement, evaluate, and propagate these methods in real-world environments.

Tasks Establish and lead efforts that get products from the initiative into software engineering and executive decision-making practice. This involves the following components:

- Working with SEI sponsors and collaborators to establish and evolve defined estimating processes within their organizations.
- Developing training materials that they and others can use to support the teaching of effective estimating practice.
- Working with cost model vendors to focus and accelerate the development and improvement of software estimating tools.

Subtasks The technology transfer thrust has four subtasks:

Subtask 7.1 *Assist sponsors and collaborators.* Assist sponsoring organizations and technical collaborators in improving the processes, practices, tools, and data that they use for estimating software costs, schedules, and sizes. The objectives are to produce early success stories and to provide opportunities for prototyping and evaluating the materials, guidelines, and process models developed in Thrusts 3-6.

Products

- Guidelines and assistance for installing and sustaining effective software estimating processes, derived from lessons learned in actually doing it.
- Guidelines and advice for using specific cost models to help with both routine and difficult estimating tasks.
- Criteria for success in software estimating.
- Consulting, facilitating, and training assistance.
- Tutorial and course materials for software engineering education programs.

Subtask 7.2 *Develop educational materials and courses.* Develop materials and courses that will assist managers, practitioners, and software engineering process groups in establishing and using defined processes for software estimating. Collect and devise examples that show how existing cost, schedule, and size models can best be used to improve estimating reliability and responsiveness. Assemble these materials in forms that will enable others to use them in software engineering and cost estimating courses.

Products

- Estimating examples and materials that others can use in courses in software engineering, management, and cost estimating.
- Tutorials that the SEI and others can use as bases for developing full-scale courses in software cost and schedule estimating.
- Improvements to materials and examples on software estimating that are used in existing SEI courses and executive seminars.

Subtask 7.3 *Focus and accelerate the development of improvements to software cost, schedule, and sizing models.* Work with cost model vendors to improve the tools they produce and market. Through persuasion and encouragement, motivate model evolution and guide it in directions that provide improved support for defined estimating processes and practices.

Products

- Market opportunities for cost model vendors.
- Guidance to vendors and developers of cost models that helps them produce and evolve products that meet the estimating needs of the software engineering community.
- Accelerated evolution and improvement of software estimating tools and products.

Subtask 7.4 *Prepare and present papers and tutorials on software estimating.* Illustrate and promote effective estimating processes and practices.

Products

- Tutorials and papers delivered at conferences such as: the SEI Software Engineering Symposium; Software Engineering Process Group National Conferences; Tri-Ada; the Annual National Conference on Software Technology; the DoD Cost Analysis Symposium; COCOMO, SLIM, PRICE, and SEER Users' Group meetings; and

conferences of the Institute of Electrical and Electronics Engineers, the International Society of Parametric Analysts, the Society of Cost Estimating and Analysis, and IFPUG (the International Function Point Users Group).

- Technical reports.
- Papers and articles for professional journals.

3.2 Future Thrusts

Two additional thrusts have been identified, but they are not included in the initiative at the present time. These are

- Estimating for post-deployment system support (maintenance and evolution), and
- Estimating for internal systems development

We are deferring these thrusts not because they are unimportant, but because tackling them now would spread our resources too thin. Moreover, our experience suggests that acquisition and development environments offer a potential for identifying broadly applicable estimating principles and processes, and that methods that work well for estimating development efforts often can be applied to maintenance, support, and internal development. The reverse is less often true.

4. Process Modeling

The SEI initiative is directed toward process improvement. The first step in any process improvement effort is to understand the processes that are currently used. There may be reasons for people doing what they are doing now. We should make sure we understand these reasons before attempting to redesign their processes.

Understanding (and definition) of processes must be documented. This documentation must be in forms that

- can be analyzed
- can be converted to defined processes or to improvements to defined processes
- can be communicated to others

Unless a process is very simple, models of the process are almost always needed to achieve these objectives.

The purpose of a process model is to describe a process in ways that account for its *important properties*. Process models are used for baselining, for gaining insights, for designing and communicating improvements, and for measuring and interpreting results. They occur in many forms and many settings—some of which we may not customarily recognize as models. For example, the process definitions that lie at the heart of level 3 of the Capability Maturity Model are inherently models.

Because process definition plays so strong a role in improving process capability, we are exploring several methods for constructing models of processes. We have been seeking (and are continuing to seek) effective ways for identifying, defining, and communicating the essential elements of good estimating practice. This chapter discusses some of the methods we have been examining. It provides motivation for their use and reviews some of the attributes of processes that should be considered when choosing a particular kind of modeling.

4.1 Why Define Estimating Processes?

The need for good estimates (and good estimating processes) is stated clearly in the key practices of the Capability Maturity Model (CMM) [Paulk 93b]. Three of the CMM's key process areas for level 2 (repeatable) processes are project planning, project tracking, and subcontract management. These process areas must have reliable estimates for size, effort, schedule, and cost if they are to be performed successfully. The CMM requires that the procedures for producing these estimates be documented. This implies, in turn, that the processes for deriving estimates must be defined—a requirement that, for other parts of the software process, is not encountered until level 3.

Thus cost estimating, which is a subset of the software process, is itself a process. To understand an estimating process, we must decompose it into comprehensible pieces. This is similar to decomposing a software system into its component pieces. Improving an estimating process can then proceed according to the basic steps of process improvement described by Humphrey [Humphrey 89]:

1. Understand the current status of the process.
2. Develop a vision of the desired process (its component pieces and their inputs, outputs, and interactions).
3. Establish a list of required process improvement actions, in order of priority.
4. Produce a plan to accomplish the required actions.
5. Commit the resources to accomplish the plan.
6. Start over at step 1.

Describing a cost estimating process (or any other process, for that matter) is an important part of steps 1 through 4. Our goal is not just to understand current estimating processes, but to be able to manage and improve them. We must also be able to implement desired estimating processes and train employees in their use. Descriptions and definitions of existing estimating processes and understanding of how well these processes perform today will give us the starting points we need to begin our improvement journey.

4.2 Methods for Describing or Defining a Cost Estimating Process—Some Considerations

There are several methods we can use for describing or defining a cost estimating process. The first considerations in choosing among them are

- When will the description or definition be used?
- Who will use it?

One method may be appropriate for designing or analyzing the process and another for guiding people in implementing it.

A suitable vocabulary and an understanding of interrelationships are prerequisites to understanding any problem. The methods we use must provide both a vocabulary for discussing the process and a means for understanding process dependencies. Process dependencies often become more evident when they are displayed graphically.

A model provides a vehicle for reasoning about the process, and it promotes discussion and refinement of the proposed definition. A modeling method can also provide a format for comparing different processes.

Once a process is defined, models continue to have value. For example, educating others in implementing and performing a desired process is always a challenge. Models are often useful for introducing the concepts of the process and for explaining the sequence of tasks

that must be performed. Graphical models, in particular, frequently provide clarity of exposition that is not achievable with simple text.

Sometimes models that are useful for gaining understanding and for identifying improvement opportunities use notations that are unfamiliar to people who work within the process. When this is so, it may be appropriate to rely on one model to define the process, but to supplement that model with another or with natural language descriptions.

Another important consideration in choosing a method for describing or defining a process is the content of the process. Here again a suitable vocabulary is important. For example, if the process is concurrent or asynchronous and the methods chosen to represent the process cannot describe this behavior, the best that can happen is that the process definition will be unnecessarily complex. A more likely outcome is that the representation will be inaccurate and incomplete. Handling of exceptional circumstances, communications (internal and external), and iterations are other aspects of processes that can require specialized vocabularies and notations.

The Software Process Definition Project at the Software Engineering Institute (SEI) has been developing a structured framework for defining software processes [Armitage 93]. This framework provides checklists that summarize the information needed for a process to be enactable. In this framework, a process is defined by its entity classes (agents, artifacts, and activities) together with descriptions of the aspects of class relationships and behavior. The entity classes describe who does the work (the agents), what is produced or consumed (the artifacts), and how the artifacts are produced or consumed (the activities). Aspects include entry and exit criteria, states and transitions, and pre- and post-conditions.

A related view [Over 93] holds that there are four process modeling perspectives:

- functional (what is done),
- organizational (who does it, where it is done),
- behavioral (when it is done, how it is controlled), and
- informational (what information entities are involved, how these entities are interrelated).

In this view, a process definition (and a method for defining processes) can be judged by how well it addresses each perspective. [Over 93] also discusses some emerging requirements for process modeling approaches. These are organized as

- representation capabilities needed,
- representation capabilities desired,
- modeling capabilities desired, and
- enactment capabilities desired.

The salient concept is that there are many different methods for defining a process. Every method has both strengths and weaknesses. Depending on the nature of the process being defined and the purpose of the definition, one method may be more useful than others. Alternatively, combinations of methods may be needed. Whatever the case, one requirement

for a useful process definition must be that it be enactable. The recent work by Armitage et al. cited above has identified the elements that must be present in any process definition for this requirement to be met.

In short, there are many reasons to turn to process models as important enablers for process improvement. In our work in this initiative, we foresee using process models to

- clarify thinking and facilitate thought processes
- identify agents, artifacts, activities, and relationships
- locate and identify inconsistencies and missing elements
- guide discussions and refinements
- provide vocabularies and tools for analysis and solution finding
- aid in communicating and teaching the process after it is defined
- help make process definitions enactable

4.3 Examples of Modeling Methods

Some of the process modeling methods we have begun exploring are

- EITVOX (entry criteria, input, task, validation, output, exit criteria)
- IDEF0 and Design/IDEF®
- State charts and Statemate®

The following discussions provide brief illustrations of these methods.

EITVOX

EITVOX stands for entry criteria, input, task, validation, output, exit criteria. This modeling method is an extension of the ETVX (entry, task, validation, exit) paradigm described by Radice and Phillips in their 1988 book *Software Engineering—An Industrial Approach* [Radice 88].

Figure 4-1 illustrates the ETVX notion that, whatever the level of abstraction or refinement for a work activity, there must be entry and exit criteria, there is a task to be done, and there is a need to validate that which is done. Without these four basic building blocks, process models are incomplete, and there are no assurances that products are being developed as required.

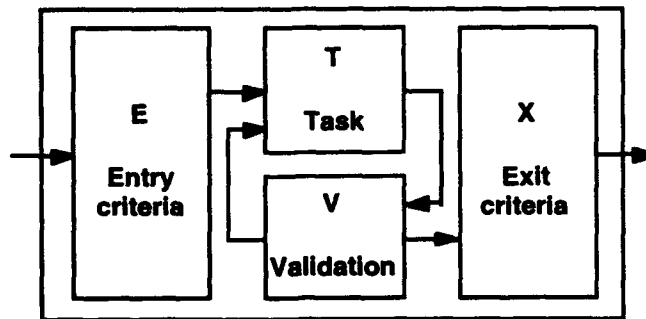


Figure 4-1 The ETVX Paradigm

ETVX models can be linked together to describe stages in a process (Figure 4-2). They also can be decomposed to describe subprocesses (Figure 4-3). Although linking describes sequences of action, it does not imply that later stages must wait for completion of earlier stages before they can start. All that is required for a process to begin is that its entry conditions be satisfied. This readily permits modeling of asynchronous, parallel, and concurrent activities.

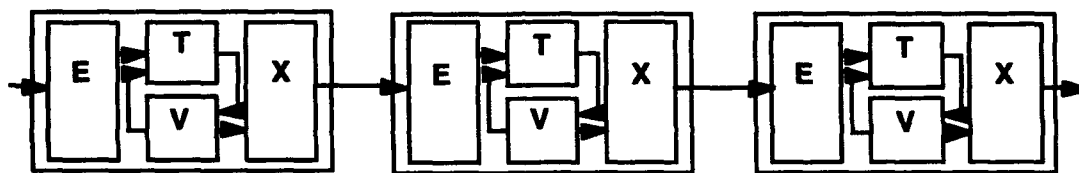


Figure 4-2 Using ETVX to Show the Stages of a Process

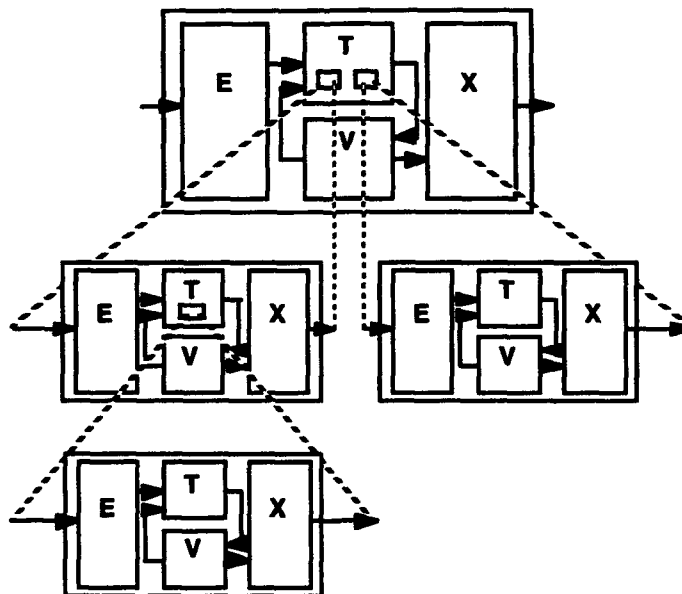


Figure 4-3 Using ETVX to Decompose a Process into Subprocesses

EITVOX extends the ETVX methodology so that it also accounts for the inputs and outputs at each stage. This includes accounting for the origins and destinations of the inputs and outputs. With these additions, EITVOX takes on many of the properties of an elaborated data-flow diagram—but in a way that accounts not just for data flow, but also for data availability.

The notation (and tool) we have been using for probing, recording, and organizing EITVOX information is a process template developed by the Software Process Definition Project [SEI 94]. Figure 4-4 shows the parts of this template that describe activities. The annotations in Figure 4-4 are generic ones, provided by the template's authors to illustrate the kinds of information required for each block.

We have been using this template to help our technical collaborators map their existing estimating processes. It has been quite effective in uncovering missing links, undefined components, and unassigned responsibilities. Even where documented processes exist, probing guided by the template has identified incompletely defined activities, inputs that appear miraculously from nowhere, outputs that are not captured and retained for future use, and intertangled or unclear responsibilities. Each of these findings has become a target for clarification and process improvement.

Our initial experience is that the template supplies a structure and rigor that are difficult to achieve with the verbal descriptions and flowcharts used in most documented processes. The template's greatest strength, especially when used to decompose tasks into subactivities, lies in ensuring that nothing is overlooked.

The difficulties we have encountered when using the template have taken two forms. First, our process mappings were more labor intensive than we had anticipated. In fairness, this is not a critique of either the template or of EITVOX, but of the difficulties inherent in constructing reasonably complete process definitions. Had we taken a less structured route, we probably would have missed many of the discoveries we made.

Our second criticism is of more concern. Although we have found EITVOX templates to be very useful for probing, structuring, and gaining insights (and we suspect they will be helpful also for implementing process improvements), we have not found them well suited for communicating summary information to the decision makers whose support must be enlisted for real process improvement to proceed. We conclude that the EITVOX paradigm and its templates must be supplemented by simpler, more visual methods for communicating effective summaries of defined processes. As it stands now, we don't know how to do this without glossing over elements that should be understood if correct decisions are to be made. We expect to report on our progress in balancing these conflicting needs in future reports.

Activity: *Enter the ID and name of the Activity*

Purpose *Describe the purpose or rationale for this activity.*
(Why is this activity performed?)

Performed by *(Who is responsible for performing this activity?)*

Name or ID of Agent
<i>List the organizational units or roles.</i>

Entry Criteria *(When can this activity begin?)*

State or Condition	From Activity	[and] [or]
<i>State as a simple or compound rule in terms of the state of an activity, product, or agent.</i>	<i>List the source activity that results in this state or condition.</i>	

Inputs *(What products are used by this activity?)*

Product name or ID	Source activity name or ID
<i>List the ID and product name of each input to the activity.</i>	<i>List the source activity for this input.</i>

Continued on next page

Figure 4-4 A Template for Process Description

Activity: ID or Name of Activity, continued

Parent Activity

Enter the ID or name of the parent activity to describe the activity hierarchy.

Sub-activity, Procedure, or Method

(How is this activity implemented?)

Step	Description
	<i>Describe the sub-activities or procedures to be followed for this activity. For activities at the bottom of the hierarchy, enumerate the steps.</i>

Exit Criteria

(When is this activity completed? What activity is next?)

State or Condition	To Activity	[and] [or]
<i>State as a simple or compound rule in terms of the state of an activity, product, or agent.</i>	<i>List the destination activity for this state or condition.</i>	

Outputs

(What products are produced by this activity?)

Product name or ID	Destination activity name or ID
<i>List the ID and product name of each output from the activity.</i>	<i>List the destination activity for this output.</i>

Figure 4-4 (Continued) A Template for Process Description

IDEF0 and Design/IDEF®

IDEF0 is a diagramming method for describing systems in terms of the processes they perform. It is derived from (and is very similar to) SADT™ (Structured Analysis and Design Technique) [Colquhoun 93]. Design/IDEF® is a software tool for constructing IDEF0 models.

IDEF0 is one of a family of models developed under U.S. Air Force sponsorship for the ICAM (Integrated Computer Aided Manufacturing) program. Two other models are IDEF1X (ICAM Definition Method 1—extended), for logical data modeling, and IDEF2 for modeling system dynamics.

The atomic building blocks of IDEF0 models are boxes and arrows. The boxes represent activities, and the arrows describe interfaces between the activities or between an activity and its environment.

The notion of interface is specialized in an IDEF0 model. For example, if an activity is seen as a function, then there may be interfaces that act as input to the activity and others that act as output. If the activity is not one function, but a set of mappings from inputs to outputs, another interface may act as the control that determines how to perform the mapping from inputs to outputs. Finally, there may be an interface that provides the mechanism by which the activity is performed.

In their text on SADT™ [Marca 87], Marca and McGowan describe the different interfaces as

- Input: Things used and transformed.
- Control: Things that constrain or direct how activities are performed.
- Output: Things into which inputs are transformed.
- Mechanism: How activities are realized (e.g., the physical aspects of an activity).

Each of the four sides of an activity box in an IDEF0 model accepts one type of interface. The standard relationship is shown in Figure 4-5.

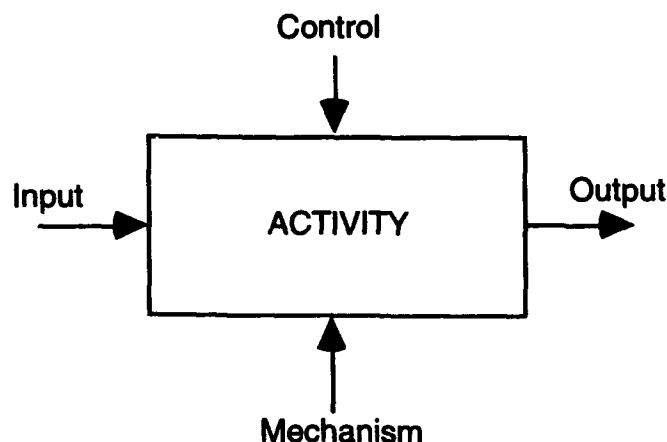


Figure 4-5 IDEF Building Blocks

When viewing an IDEF0 model, we know that the inputs to the activity always enter the activity box from the left and the outputs leave the activity box to the right. This mapping between interface type and sides of the activity box adds substantial clarity to a process model. At a glance we can determine the relationship between activities joined by an interface arrow.

Hierarchy fits naturally into the IDEF0 paradigm. For example, a model can be seen as a single, top-level activity with interfaces to its environment. This top-level activity can be decomposed into its component activities. This decomposition can be carried to any level of detail. An IDEF0 model is the sum of all of these views.

Interface arrows in an IDEF0 model can be branched and joined. In this way, concurrency can be represented. Synchronization between concurrent activities can be represented via control interfaces.

Tool support for IDEF0 is available. This support typically includes graphical drawing aids, together with a data dictionary that can be used to help bind textual clarification of an activity to the activity.

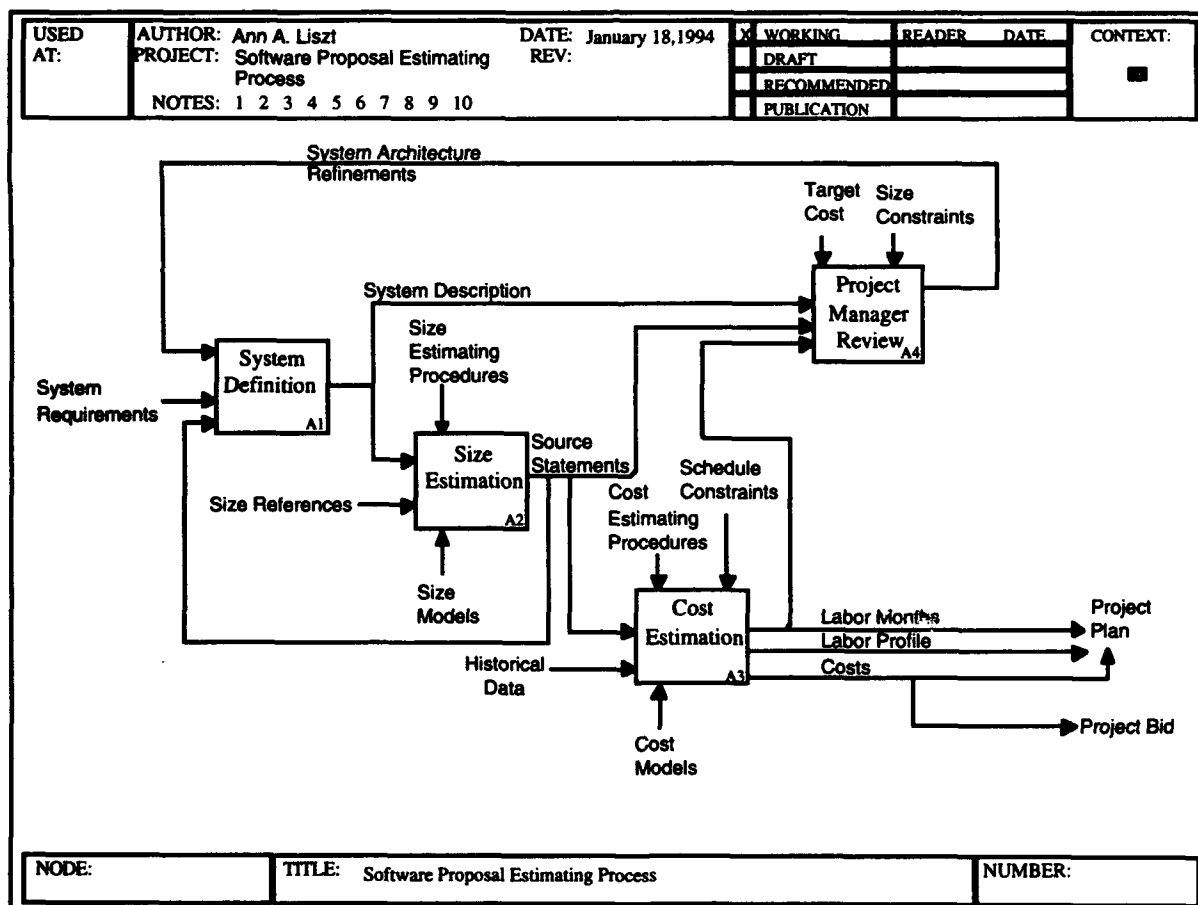


Figure 4-6 A Top-Level IDEF0 Representation of a Possible Estimating Process

Figure 4-6 is an example of how a top-level view of an estimating process might look when expressed in IDEF0 notation. In a more complete model, each activity block would be decomposed into its constituent activities.

Our experience in using IDEF0 models to describe and define estimating processes is as yet too limited to enable us to draw conclusions or offer recommendations. One of our collaborating organizations is about to start using IDEF models for their process definition work. They will soon be applying IDEF tools to construct descriptions of the size estimating processes we have helped them map with EITVOX templates. We expect to have reports of first-hand experience with IDEF models to share in the near future.

State Charts and Statemate®

The state chart is a visual mechanism for displaying and analyzing processes that was proposed by David Harel [Harel 87, Harel 88]. Statemate® is a commercial tool from i-Logix, Inc., that uses state chart notation to provide formal views or perspectives of a process.

State machines are ubiquitous in computer science. The notion has achieved this stature because of a natural mapping between states and many algorithms or processes. State machine notations can also be applied to software cost estimating processes where there are distinct "states" in the process where specified conditions hold. These states are reached when appropriate preconditions are satisfied, and they endure until defined post conditions are met.

State machines have both a well developed theory and an intuitive graphical notation to represent the theory. However, there are some relationships between states that have been difficult to represent in the traditional notation. Three of these relationships are hierarchical decomposition, concurrency, and synchronization. Harel's initial work extended the graphical representation of state machines to include these relationships. This work was directed at what Harel called *reactive systems* (real-time, user-interface driven systems). States in a cost estimating process (and in other software processes) also possess these kinds of interrelationships.

There are sometimes aspects of a process that cannot be easily defined with Harel's extended notation. For example, if one were to decompose a process using traditional functional decomposition, there may be relationships between activities that are at a coarser grain than state transitions. We must be able to define these activities and the information that flows between them without giving up the ability to specify the behavior of the activity.

Even with these two views of the process accounted for, there still remain attributes of processes that we are ill equipped to describe in state chart notation. These can be called the organizational attributes: who does the work and what communication mechanism is involved?

The Statemate tool has been designed to represent all these aspects of a process. A review of an exercise in modeling a general software process is provided by Marc Kellner in [Kellner 89]. He describes the different perspectives or views of a system as

- **Functional perspective** - representing what tasks are being performed and what information flows are pertinent to those tasks.
- **Behavioral perspective** - representing when tasks are performed, as well as aspects of how they are performed through feedback loops, iteration, complex decision-making conditions, entry and exit criteria, etc.
- **Organizational perspective** - representing where and by whom in the organization the tasks are performed and the physical communications mechanisms used for information transfer.

The Statemate tool also provides support for analyzing the process and simulating its operation.

We have not yet attempted to apply the Statemate tool or state chart mapping to describe a software estimating process. How well these tools will work is yet to be determined. At this point, they are simply two of the tools we are considering. We will need experience in their use to determine whether they are worth pursuing as process description and process improvement aids.

•

5. Estimating Process Templates—Some Preliminary Examples

This chapter presents three examples of potentially useful templates for describing, developing, and defining software estimating processes. These are preliminary, top-level examples only. They do not as yet represent processes endorsed by the SEI or by its sponsors and collaborators. Our purpose is simply to show you what some templates might look like and to stimulate thoughts and discussions that could lead to enrichments and alternative forms. We expect that more detailed, lower-level templates will also be needed to give constructive assistance and guidelines for implementing and executing specific components of estimating processes.

5.1 Example 1: A Graphical Template for Parametric Estimating

This example (or one much like it) has received use within the International Society of Parametric Analysts. It shows the flow of information and the activities associated with calibrating and using parametric cost models.

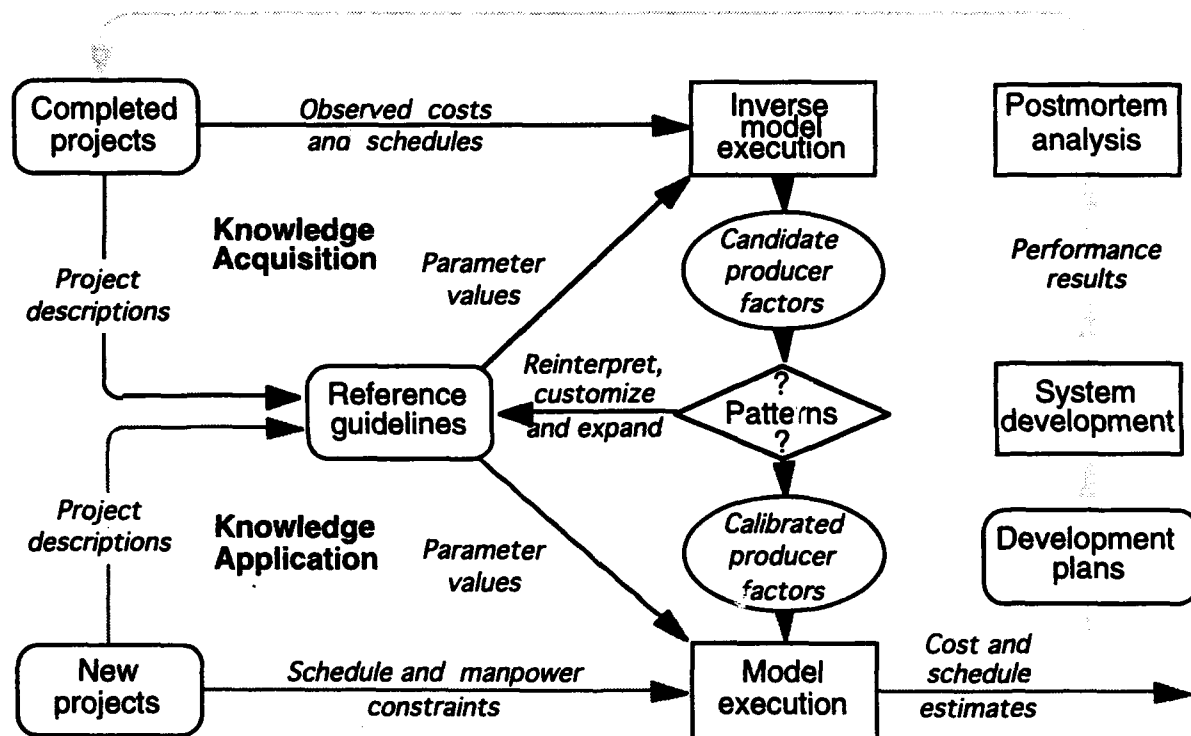


Figure 5-1 Graphic Template for Parametric Estimating

This template has its origins in three underlying principles that evolved gradually following the introduction of parametric costing models in 1975 [Park 89]:

1. Estimates are made by people, not by models. They require reasoned judgments and commitments to organizational goals that cannot be delegated to any automated process.
2. All estimates are based on comparisons. When people estimate, they evaluate how something is like, and how it is unlike, things that they or others have seen before.
3. Before people can estimate, they must acquire knowledge. They must collect and quantify information from other projects, so that they can place their comparative evaluations on demonstrably sound footings.

The heart of the process depicted in Figure 5-1 lies in the upper left quadrant, under the label of **Knowledge Acquisition**. We prefer this term to the more common term **Calibration**, because the latter sounds like something that must be done to a model or process before it can be used for estimating. This common view is often counterproductive, because it implies a permanent adjustment or setting that is needed to make a model accurate for all estimates.

But the nature of parametric estimating is that it is comparative, not absolute. In the process shown in Figure 5-1, estimates are made relative to experiences the estimator or others have quantified. The key element is descriptive consistency, not model accuracy. Knowledge acquisition is more than just calibration of models. Rather, it involves quantitative measurement and description of products, processes, and environments, so that the information that is captured can be used to base future estimates on demonstrated process capabilities.

Knowledge acquisition is thus quite different from calibration of laboratory instruments. The focus is always on the entities being measured, not on the measurement devices themselves. Also, knowledge acquisition is both continuous and never-ending, and it takes place at many levels. Cost model developers supply the initial knowledge when they organize and present their reference guidelines. Individuals add more knowledge when they use the cost models to develop local measures of products and processes. Organizations assemble and organize further knowledge when they pull this information together and sort it into consistent patterns to form corporate references and guidelines. And the software community expands upon this knowledge when it shares this information through professional societies and with model developers, so that even broader patterns can be identified and published for all to use.

All these efforts have the objective of ensuring that when the time comes to prepare an estimate, the position will be as shown in the lower left quadrant of Figure 5-1. Here, because of the exploratory work organizations have done—and with the measurements they have made—the estimator is equipped with quantified, well-understood methods and tools for relating proposed activities to results that have been achieved on previous occasions.

The strength of the parametric process is that it is both robust and self-correcting. Because it relies on consistency rather than on model accuracy, most of the contentious questions that

relate to accuracy become irrelevant. So long as organizations apply their tools when estimating in the same way they apply them when acquiring knowledge, they assure that their estimates are consistent with demonstrated capabilities.

As a by-product, summaries of the steps followed when assembling the estimating knowledge base and applying it to new projects provide an audit trail that others can use to validate the reasonableness of estimates. This aids not only in communicating estimates, but also in highlighting management issues that must be dealt with to successfully meet assigned costs and schedules.

5.2 Example 2:A Process Template for Bid-and-Proposal Environments

This example is based on ideas presented by Raymond L. Kile at the 7th COCOMO Users' Group Meeting in October of 1991 [Kile 91]. We have introduced a few modifications to incorporate our own personal experiences. The setting is software development in a bid-and-proposal environment. The process presumes use of one or more parametric cost models. Thus, it is closely related to the example in Figure 5-1.

Kile's original template had eight stages. We have added a ninth stage to make explicit the activities associated with collecting and analyzing information for future estimates. We have grouped the nine stages into four major phases:

1. Defining the scope
2. Technical analysis
3. Business analysis
4. Follow-through

The following sections describe the activities and products of each phase.

Phase 1: Defining the Scope

The first phase (stages 1-3) defines the scope of the development effort. It produces a design baseline, a size baseline, and an environmental baseline. Here estimators work with software engineers to define and quantify the physical, environmental, and organizational characteristics associated with producing the major software components. The products are lists of cost model parameter values for each major component, supported by written rationales for each value selected.

Table 5-1 outlines the activities and products required for defining the scope of a software development effort.

Defining the Scope		
Stage	Activities	Products
1. Design Baseline	Define the design in detail sufficient to identify all computer software configurations items (CSCIs) and the functionality of each.	List of CSCIs together with descriptions of their functionality. List of comparable projects and CSCIs (the estimating reference set).
2. Size Baseline	Estimate the expected size of each CSCI and determine the extent to which reuse will be used.	Estimated size for each CSCI. Estimated extent of reuse within each CSCI. Uncertainty estimates for size and reuse.
3. Environmental Baseline	Determine and quantify the physical, environmental, and organizational characteristics associated with producing each CSCI.	Lists of cost model parameters for each CSCI, their initial settings, and written rationales for each setting.
IMPORTANT: At the completion of stage 3, all parameters are defined. Subsequent changes to estimated costs or schedules require commitments and actions from management to change the conditions that the baseline parameters describe.		

Table 5-1 Defining the Scope

At completion of stage 3, all parameters are defined. Discipline is introduced by requiring that subsequent changes to estimates must be accompanied by commitments from management to change the underlying designs or conditions that these parameters describe. This means that if managers or customers do not like the estimates that result, they must do something structurally to make alternative parameter values possible. They cannot just direct estimators to change their numbers.

Phase 2: Technical Analysis

The second phase, consisting of stages 4 and 5, is where the technical analyses are performed.

In stage 4, the design, size, and environmental baselines are used in conjunction with the organization's cost models to produce baseline estimates for costs and schedules. These estimates are accompanied by an auditable trail of all cost model inputs.

Stage 5 then transforms the cost and schedule estimates into a project estimate. Factors not addressed adequately by the organization's cost models are addressed, and activities included in the cost models that do not apply to the current project are eliminated. The product is a complete estimate of costs and schedules for the project, together with projected staffing profiles and documentation of all adjustments that were made to the baseline estimates.

The stages for the technical analysis phase are outlined in Table 5-2.

Technical Analysis		
Stage	Activities	Products
4 Baseline Estimate	Use the baseline products in conjunction with the organization's cost model(s) to develop baseline estimates for costs and schedules.	Outputs from cost model(s) showing schedules and costs. An auditable trail of all model inputs, including written documentation describing how each input was derived and its relation to experience on previous projects.
5. Project Estimate	Adjust the baseline estimates by accounting for factors not addressed by the cost model(s) and by eliminating cost model activities and elements that do not apply.	A complete estimate of the costs and schedules for the software portion of the project. Project staffing profile requirements. Auditable documentation and rationale for each adjustment.

Table 5-2 Technical Analysis

Phase 3: Business Analysis

The third phase (stages 6 and 7) is where business analyses are performed. The products are the project bid (stage 6) and the associated risk analyses (stage 7).

For a project bid, estimators adjust the project estimate to account for factors such as competition, schedule constraints, budget constraints, personnel constraints, and uncertainties in size and reuse. Auditable documentation and supporting rationale are provided for each adjustment.

The risk analyses in stage 7 identify the cost and schedule risks associated with the contract. The products are risk assessments, risk graphs, and parameter-by-parameter explanations of the risks.

The stages for the business analysis phase are outlined in Table 5-3.

Business Analysis		
Stage	Activities	Products
6. Project Bid	Analyze and adjust the project estimate to account for factors such as expected competition, budget constraints, schedule constraints, personnel constraints, and size/reuse uncertainties.	Project bid. Auditable documentation and rationale for each adjustment.
7. Risk Analysis	Compare the final/approved project bid (BAFO, contract award, etc.) to the project estimate to determine the cost and schedule risks associated with the bid/contract. Vary the size, functionality, and environmental parameters to perform what-if analyses. Determine the size, reuse, and/or environmental settings required to validate the project bid.	Risk assessments. Risk graphs. Bid memorandum with parameter-by-parameter explanation of the risks.

Table 5-3 Business Analysis

Phase 4: Follow-Through

The fourth phase (stages 8 and 9) ensures effective follow-through. It deals with estimates to complete and with the postmortem analyses needed to collect and record reliable reference information for future estimates.

In stage 8, estimators use actual (measured) values of product characteristics and progress to construct revised estimates for the costs and time needed to complete the project. Products of stage 8 include updated estimates for size and reuse, updated parameter values, and rationales for each change.

Stage 9, postmortem analysis, is central to process improvement. In this stage, the organization captures the feedback needed for recalibrating cost models and for augmenting and improving parametric guidelines and reference baselines. Here all parameters are re-evaluated in light of end-of-project knowledge, and the organization's cost models are used to refine, recalibrate, and make self-consistent the recorded values that describe the project.

The stages for the follow-through phase are outlined in Table 5-4.

Follow-Through		
Stage	Activities	Products
8. Estimate to Complete (post award, continuous)	Use status, measured values, and updated projections to develop a revised project estimate and to determine the costs and schedules required to finish the project.	Updated size estimates. Updated reuse estimates. Updated parameter values and rationales for changes. Revised project estimate. Cost to complete. Schedule to complete.
9. Postmortem Analysis	Reevaluate all parameters. Apply project results and constraints in conjunction with the organization's cost model(s) to refine and make self-consistent the parametric values that describe the product and the organization's performance. Add the results to the organization's estimating database.	Final lists of values for all size, reuse, and environmental parameters, with rationales for each. An analysis of why and how the final results differed from the original project estimate. Updated and recalibrated cost model database. Lessons learned.

Table 5-4 Follow-Through

Caveats

Our template is an untested modification of Ray Kile's work. Although it conveys many important principles, additional modifications may be needed to make it more widely applicable. For example, still to be incorporated (at least in this version) are iterations and feedback paths such as those used for

- concept evaluations,
- design tradeoffs, and
- successive refinements as more detailed design information becomes available.

It would be helpful also to add extensions for integrating the major components (software-to-software system integration) and for integrating software with hardware systems.

Another limitation of the template (as it stands now) is that it presumes a particular form of cost model—one that does not permit schedule or personnel constraints to be addressed until after the baseline and project estimates have been completed. Other cost models exist that allow these types of constraints to be included and analyzed from the very start. When these capabilities are present, it makes sense to amend stages 4 and 6 to take advantage of them.

Our comments here should not be taken as being critical of Ray Kile's work. Ray has been instrumental in breaking new ground, and he has been generous in sharing his process model with us. Moreover, he has continued to evolve his process model beyond the point where we picked it up, and he is using his newer version in the courses and workshops he teaches. One of our open tasks for this initiative is to revisit Ray's work and benefit from the advances he has made.

5.3 Example 3: Treating Estimating Processes as Process Assets

Our third example is more conceptual. It is derived from a figure used to depict the process framework of the Capability Maturity Model [Paulk 93b]. Following suggestions made by Gerald McCarty, a senior member of the technical staff at the SEI, we have adapted the figure so that it shows how organizations could be treating estimating processes as part of their process asset library—collecting and storing them, and then adapting them to new projects as the need arises.

The top-level template that results is shown in Figure 5-2. The textual materials and guidelines that support this template and help users implement the processes it depicts are yet to be developed.

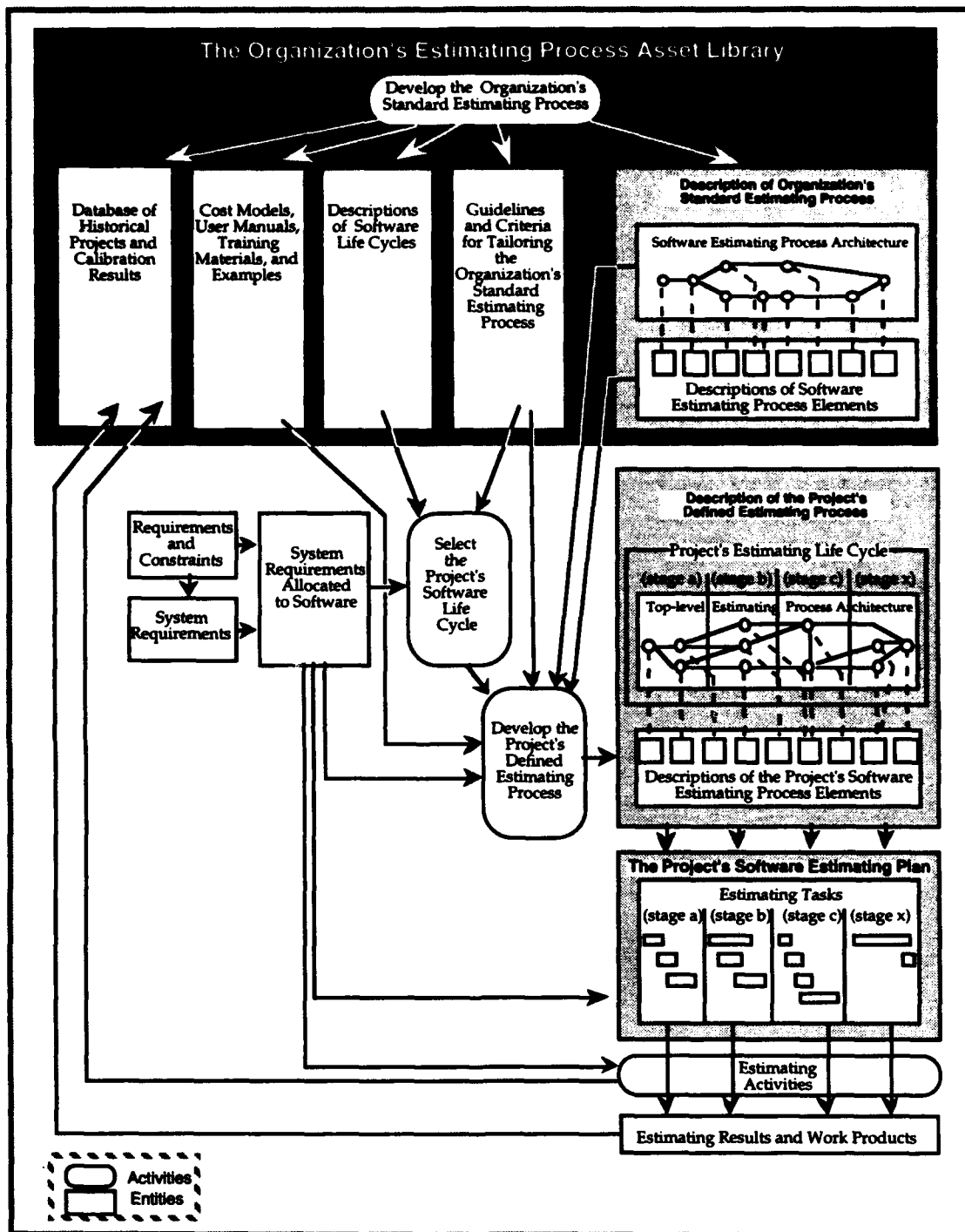


Figure 5-2 Estimating as a Process Asset

6. Schedule, Status, and Points of Contact

6.1 Schedule

The schedule for the initiative that was operative through the first three quarters of 1993 is shown in Figure 6-1. This schedule is now being revised to accommodate the delays in funding we are experiencing and to provide for integrating the efforts of technical collaborators who will be joining the initiative during 1994. Although the time scale will move to the right, the logical flow that Figure 6-1 shows continues to be accurate, as does the potential for overlapping of activities that is depicted.

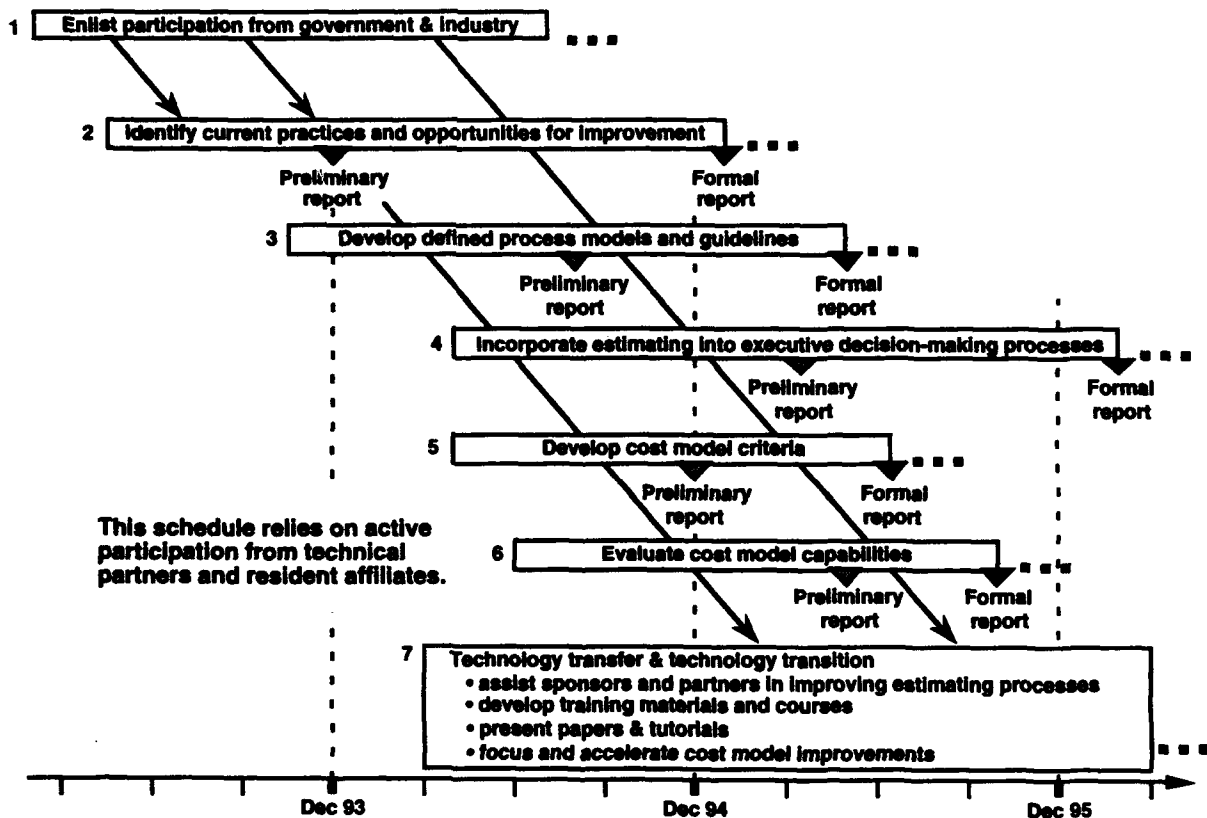


Figure 6-1 Schedule as Proposed in Early 1993

6.2 Status

As of May 1994, the SEI had two technical collaboration agreements in place for cost estimating improvement work, two more in process, and exploratory discussions underway with a fifth organization. Table 6-1 lists the collaborators and summarizes their status.

Technical Collaborators	Status
Loral Federal Systems—Manassas	Work started in August, 1993. Initial process mappings complete in December, 1993.
SETA Corporation	Nondisclosure agreement signed and operational. Technical collaboration agreement in final stages of revision. Size estimating development work underway at SETA.
Texas Instruments	Draft collaboration agreement being processed.
Electronic Data Systems	A draft collaboration agreement is being prepared.

Table 6-1 Status of Technical Collaborations

6.3 Points of Contact

If you would like to explore ways to participate in the software cost estimating improvement initiative, we would be pleased to talk with you. You can contact us by telephone or electronic mail as follows:

Robert E. Park
Telephone: (412) 268-5785
email: rep@sei.cmu.edu

Wolfgang B. Goethert
Telephone: (412) 269-3889
email: wbg@sei.cmu.edu

Alternatively, you can reach us by mail or Fax at:

The Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
Fax: (412) 268-5758

References

- [Armitage 93] Armitage, James W., Kellner, Marc I., and Phillips, Richard W. *Software Process Definition Guide—Content of Enactable Software Process Definitions* (CMU/SEI-93-SR-18). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, August 1993.
- [Boehm 81] Boehm, Barry W. *Software Engineering Economics*. Englewood Cliffs, N.J.: Prentice-Hall, 1981.
- [Colquhoun 93] Colquhoun, Gary J., Baines, Ray W., and Crossley, Roger. "A state of the art review of IDEF0," *International Journal of Computer Integrated Manufacturing*, Vol. 6, No. 4:252-264, 1993.
- [Harel 87] Harel, D. "Statecharts: A Visual Formalism for Complex Systems," *Science of Computer Programming*, 8:231-274, 1987.
- [Harel 88] Harel, D. "On Visual Formalisms," *Commun. ACM* vol. 31, no. 5, pp. 514-530, 1988.
- [Humphrey 89] Humphrey, Watts S. *Managing the Software Process*. Reading, Mass.: Addison-Wesley, 1989.
- [Kellner 89] Kellner, Marc. "Software Process Modeling: Value and Experience," *Technical Review 1989, Software Engineering Institute*.
- [Kile 91] Kile, Raymond L. "A Process View of Software Estimation," *Proceedings of The Seventh International COCOMO Estimation Meeting*. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, November 1991.
- [Marca 87] Marca, D. A. and McGowan, C. L. *Structured Analysis and Design Technique*, McGraw-Hill 1988.
- [Over 93] Over, J. W., et al. *Fundamentals of Software Process Definition*, an SEI Software Engineering Symposium tutorial. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, August 1993.
- [Park 89] Park, Robert E. and Korda, Peter B. *Making Estimates Credible—or, Why Should I believe What You Are Telling Me?* *Journal of Parametrics*, Vol. IX, No. 2, June 1989.
- [Paulk 93a] Paulk, Mark C. et al. *Capability Maturity Model for Software, Version 1.1* (CMU/SEI-93-TR-24). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, February 1993.
- [Paulk 93b] Paulk, Mark C. et al. *Key Practices of the Capability Maturity Model, Version 1.1* (CMU/SEI-93-TR-25). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, February 1993.
- [Radice 88] Radice, Ronald A., and Phillips, Richard W. *Software Engineering, An Industrial Approach*. Englewood Cliffs, New Jersey: Prentice Hall, 1988.

[SEI 94]

Course notebook for *Defining Software Processes*, a process improvement course offered by the SEI. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1994.

Acronyms

BAFO	Best and final offer
CMM	Capability Maturity Model (described in [Paulk 93a] and Paulk 93b))
CMU	Carnegie Mellon University
COCOMO	Constructive Cost Model [Boehm 81]
CSCI	Computer software configuration item (A military term for a major software component. Defined in DoD-STD-2167A)
DoD	Department of Defense
EITVOX	Entry-Input-Task-Validation-Output-Exit (a term for a form of process modeling derived from work by Radice and Phillips [Radice 88])
ICAM	Integrated computer manufacturing
IDEF0	A modeling method, derived from SADT™, that views a system as the set of functions it performs
IDEF1X	A modeling method that views a system by studying the information it contains
IDEF2	A modeling method that views the time-varying behavior of a system
IEEE	The Institute of Electrical and Electronics Engineers, Inc.
IFPUG	The International Function Point Users Group
LOC	Lines of code
PRICE	Parametric Review of Information for Cost Evaluation (a family of cost models developed in the 1970s and 1980s by RCA. Now sold and supported by Martin Marietta PRICE Systems, Moorestown, NJ.)
SADT™	Structured analysis and design technique
SEER	System evaluation and estimation of resources (a family of cost models developed and supported by Galorath Associates, Incorporated, Marina del Rey, CA.)
SEI	Software Engineering Institute
SEPG	Software engineering process group (teams organized by software organizations to guide and coordinate internal software process improvement efforts.)
SLIM	Software Life-Cycle Cost Model (an estimating tool developed and supported by Quantitative Software Management, McLean, VA)

TCA	Technical collaboration agreement (a template for TCAs is presented in Appendix C)
4GL	Fourth-generation language

Appendix A: Survey Form

Software Cost Estimating Improvement —An Introductory Survey—

Background:

As part of the Software Measurement Project, the SEI has launched an initiative to improve the processes and practices used in software cost and schedule estimating. We expect the initiative to be a long-term effort. It addresses concerns expressed by senior executives of the armed services and industry, and it supports the capability maturity model, which describes the important roles that software estimating and cost management play in advancing to levels 2 and 3 of process maturity.

Products from the initiative will include summaries of current estimating practice, templates for establishing defined estimating processes, guidelines for selecting cost models, principles and examples for developing and communicating verifiable software estimates, training materials that support the teaching of effective estimating, and assistance to technical partners and sponsors.

We have attached a short survey that solicits your views on how well today's software estimating is meeting the needs of your organization. The survey also gives you an opportunity to provide your recommendations for the work we have planned, and it gives you a place to let us know how you might like to participate in the initiative and help shape its direction.

Please feel free to reproduce the survey and give it to anyone interested in software estimating. We will use the responses to guide our efforts so that they produce products that meet the needs of the software community. The answers you provide will be treated as proprietary information. They will not be disclosed or cited in attributable ways without your permission.

If you have questions, or if you would like to explore ways to work with us in this initiative, please contact Bob Park at (412) 268-5785 or Wolf Goethert at (412) 268-3889. You can obtain copies of the task plan for the initiative either by writing to us or by calling us at the above numbers.

Instructions:

Please return the completed survey form to:

Dr. Robert E. Park
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Thank you for your time and consideration.

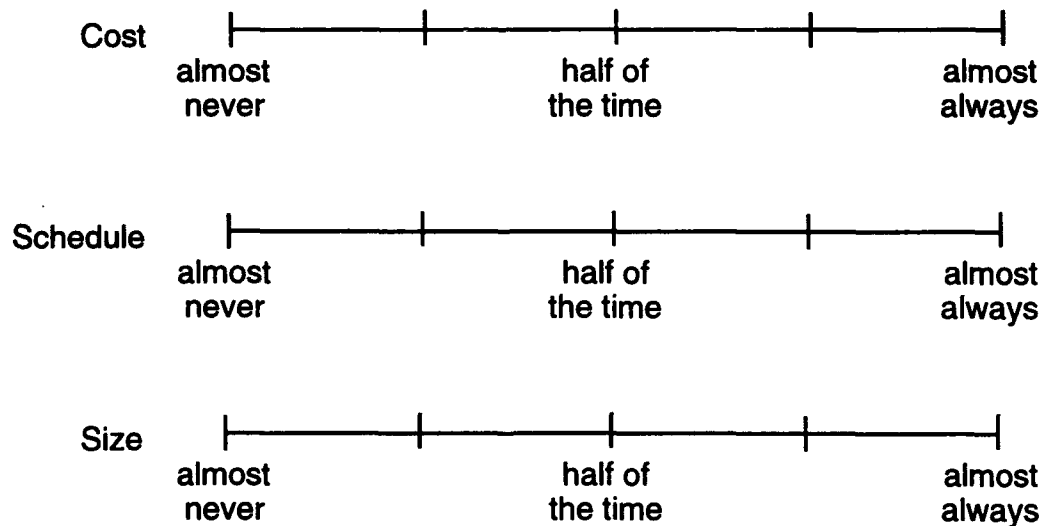
Software Cost Estimating Improvement
— An Introductory Survey —

1. Reflecting on what you have seen over the last 2 or 3 years, where are software estimates used in your organization? (check all that apply)

- | | |
|--|--|
| ☐ Concept exploration | ☐ Project tracking |
| ☐ Design evaluation | ☐ Project staffing |
| ☐ Bid/no-bid decisions | ☐ Resource leveling |
| ☐ Proposal preparation | ☐ Estimates to complete |
| ☐ Proposal evaluation | ☐ Replanning & rescheduling |
| ☐ Contract negotiation | ☐ Other: _____ |
| ☐ Project planning & scheduling | ☐ Other: _____ |

2. Based on your personal observations, how well would you say that estimates for software cost, schedule, and size are meeting the needs of your organization?

(Please mark the scales below to indicate your views.)



3. What is your principal role with respect to software estimating?

(Check all that apply)

- | Roles | Examples |
|--|---|
| ☐ Producer of estimates | those who produce the numbers
- cost estimator
- size estimator
- independent cost analyst
- project manager
- project scheduler |
| ☐ User of estimates for planning or managing programs or projects | program manager
project manager
program & project planner
program or project scheduler
program & project tracker
bid & proposal preparer |
| ☐ User of estimates for evaluating programs or projects | government acquisition manager
acquisition management staff
proposal evaluator |
| ☐ Manager of a business or government enterprise with oversight or approval responsibility for several projects or programs | general manager
business manager
senior government official with responsibility for program review, authorization, or approval |
| ☐ Process improvement, quality improvement, or software measurement | software engineering process group (SEPG) member
quality improvement team member
software measurement |
| ☐ Teacher, trainer, or educator | |
| ☐ Cost model developer or vendor | |
| ☐ Consultant | |
| ☐ Other (please explain) | |

7. If you (or your organization) would like to work with the SEI in the software cost estimating improvement initiative, please indicate where your principal interests lie:

Interested Highly
interested interested

- | | | |
|--------------------------|--------------------------|---|
| ☐ | ☐ | Providing examples of estimating processes and practices |
| ☐ | ☐ | Providing access to people, processes, and data |
| ☐ | ☐ | Reviewing draft products |
| ☐ | ☐ | Working actively with the SEI to improve software cost, schedule, & size estimating |
| ☐ | ☐ | Becoming a technical partner or sponsor |

8. Who should the SEI contact to follow up on your interests?

☐ Please contact me

☐ Please contact the individual listed below:

Name: _____ Telephone: _____

Address: _____

9. Your name: _____ Telephone: _____

Title or position: _____

Company/Organization: _____

Address: _____

Appendix B: Task Plan

31 March 1994

— Task Plan — for The SEI Software Cost Estimating Improvement Initiative

Objectives

1. Improve the ability of government and industry organizations to estimate costs and schedules associated with developing, maintaining, and supporting software systems.
2. Develop criteria and processes for communicating verifiable software estimates, both within organizations and between contractors and customers.
3. Establish a capability at the SEI for assisting sponsors, technical partners, practitioners, and teachers in their efforts to improve the processes and practices of software estimating.

Overview

This is a multiyear initiative that addresses concerns expressed by senior executives of the armed services and industry. The initiative supports the Capability Maturity Model (CMM), which describes the important roles that estimating and cost management play in advancing to levels 2 and 3 of process maturity.

Products from the initiative will include

- benchmarks and baselines of existing estimating practice
- criteria and examples for producing, using, and communicating verifiable estimates
- templates for implementing defined estimating processes
- guidelines for communicating software estimates so that assumptions on which estimates are based are made visible and verifiable
- assistance to partners and sponsors in implementing defined estimating processes and practices
- cost model criteria, evaluations, and usage guidelines
- educational materials and tutorials that support the teaching of effective estimating

The initiative will also identify areas where cost, schedule, and sizing models can be improved, so that evolution of estimating tools can be focused and accelerated.

Approach

The initiative has seven principal thrusts:

1. *Enlist sponsors & partners and establish working relationships*
2. *Identify current estimating processes, practices, and opportunities for improvement*
3. *Develop defined process models and guidelines for estimating practice*
4. *Incorporate estimating into executive decision-making processes*
5. *Develop cost model criteria*
6. *Evaluate cost model capabilities*
7. *Technology transfer and technology transition*

Two further thrusts—*Estimating for post-deployment system support* and *Estimating for internal systems development*—are not included in this initiative at the present time. They are noted here so that they can be planned for and addressed in future work.

Thrust #1: Enlist Sponsors & Partners and Establish Working Relationships

Ensure that industry and government organizations are informed of the initiative and have opportunity to volunteer to take part. Establish technical collaboration and nondisclosure agreements with participating organizations. Secure funding to support the SEI portion of the initiative efforts. Recruit resident affiliates to support the initiative, and ensure that both their home organizations and the SEI receive full benefits from their work.

- Objectives:
- Obtain and sustain industry and government support.
 - Arrange for access to the information needed to assemble factual descriptions of current practices and processes.
 - Recruit technical collaborators and establish working relationships.
 - Recruit resident affiliates to support the initiative.

- Issues:
- This thrust addresses the planning and recruiting efforts that must be performed to ensure successful launching of the initiative.

- Actions:**
- Prepare and maintain current summaries of task descriptions and task plans.
 - Identify industry and government organizations that can provide helpful information about software estimating practices and processes.
 - Identify industry and government organizations that are potential technical partners.
 - Identify organizations that are potential sponsors.
 - Ensure that industry and government organizations are aware of the initiative and have opportunity to volunteer to participate.
 - Select and enlist participants.

- Products:**
- Briefings and presentations to industry and government leaders.
 - Briefings and presentations to prospective participants.
 - Technical collaboration agreements.
 - Nondisclosure agreements.
 - Resident affiliates to work on the initiative.

Thrust #2: Identify Current Estimating Processes, Practices, and Opportunities for Improvement

Identify the estimating processes and practices that are used today to support planning, evaluation, and control of software acquisition and development. This is an assessment and diagnosis step that logically precedes the design of process improvements. As existing processes and practices are cataloged and defined, unfilled needs (both local and global) will be identified, together with opportunities for improvement. Those who participate in the initiative will have first access to this information, so that they can make early use of it in their process improvement and benchmarking activities.

This thrust is divided into two tasks—one directed toward identifying current practices in system acquisition environments and the other aimed at targeting developer environments. Sponsors and technical partners may elect to work with the SEI on either or both of these tasks.

Task 2A: System Acquisition Environments

- Objective:**
- Identify the processes, practices, and tools that US government organizations use for producing or obtaining software estimates.

- Issues:**
- Improvement actions must be based on an understanding of what people are doing now and why they are doing it. Without this information, there is no reason to believe that change will bring improvement, nor are there baselines from which to measure progress.
 - Two useful ways to identify opportunities for improvement are through
 - (1) understanding where people are encountering difficulties in performing current tasks, and
 - (2) identifying tasks that are not being performed because capabilities are lacking.
- Actions:**
- Identify the processes, practices, and tools that are used to generate software estimates today. Do this by visiting and working with sponsors and other acquisition organizations to help them trace out and map their current estimating methods.
 - Identify where and how software estimates are used and why they are at times altered, ignored, or otherwise viewed as unsatisfactory.
 - Identify the methods used to substitute new estimates for old.
 - Identify the methods and processes used to incorporate feedback from project tracking into updated project plans and into the preparation of future estimates.
 - Identify unfulfilled estimating needs.
 - Identify opportunities for improving estimating processes, practices, and tools.
- Products:**
- Baseline descriptions of the software estimating processes, practices, and tools currently used in acquisition environments.
 - Lists and descriptions of unfulfilled needs and opportunities for improvement.
 - Feedback that participating organizations can use to motivate and guide local quality and productivity improvement efforts.

Task 2B: Developer Environments

- Objective:**
- Identify the processes, practices, and tools that developers use to estimate costs and schedules for producing software systems.
- Issues:**
- Good estimating processes, practices, and tools are most likely to originate in organizations that build software systems. This is where project history is best known, where processes are most visible, and where the (often proprietary) data reside.

- When acquisition organizations prepare estimates, even early ones, they must address (even if they approximate) the same issues that developers account for when preparing bids and proposals. Effective methods for estimating development must be understood in order to improve the processes used for making acquisition estimates.
- Understanding the estimating practices and needs of developers can help identify effective methods for communicating the government's estimating requirements.
- Understanding the practices and needs of developers can help identify effective methods for acquiring and exchanging cost data and cost estimating inputs.
- Learning from developers has historically been an effective way for government agencies to improve their estimating practices.

Actions:

- Conduct field investigations with industrial partners and other development organizations to identify the processes, practices, and tools used to produce software estimates.
- Identify where and how software estimates are used within development organizations and why they are at times altered, ignored, or otherwise viewed as unsatisfactory.
- Identify the methods used to substitute new estimates for old.
- Identify the methods and processes used to incorporate feedback from project tracking into updated project plans and the preparation of future estimates.
- Identify unfulfilled estimating needs.
- Identify opportunities for improving estimating processes, practices, and tools.

Products:

- Baseline descriptions of the estimating processes, practices, and tools currently used by software developers.
- Lists and descriptions of unfulfilled estimating needs and opportunities for improvement.
- Feedback that participating organizations can use to motivate and guide local quality and productivity improvement efforts.

Thrust #3: Develop Defined Process Models and Guidelines for Estimating Practice

Develop templates for software estimating processes that organizations can use as prototypes for improving estimating practices. Support these templates by developing and publishing guidelines and criteria for implementing and sustaining effective estimating capabilities, and for executing good estimating practices.

- Objectives:**
- Provide process models and templates that organizations can use as guides for initiating and sustaining improved software estimating practices.
 - Provide guidelines for performing key steps in preparing software estimates.
 - Provide guidelines and methods for comparing software costs across different contractors, work sites, and development organizations.
- Issues:**
- Reliable estimating requires the use of defined estimating processes. Without visible process definition, processes cannot be expected to be repeatable, and users (consumers) of estimates can have little reason to trust (or use) estimates that are presented to them.
 - Defined estimating processes are prerequisites for establishing the criteria needed for evaluating estimating practices, models, and tools.
 - Templates for defined processes can provide effective bases for technology transfer.
 - Small organizations and small projects may use different methods and processes for estimating than are needed for large systems.
 - Organizations at different levels of software process maturity may need different process models for estimating.
 - Examples of defined estimating processes can provide useful references for software process assessments and capability evaluations.
- Actions:**
- Evaluate the estimating processes observed in Tasks 2A and 2B. Identify the characteristics of the best processes and practices.
 - Develop process models for effective software estimating. Note that this may require a sequence of process models that addresses the evolving needs and capabilities of software organizations as they progress through increasing levels of process maturity. It may also require alternative process models and practices for small projects, small organizations, or differing environments.
 - Develop and define process models and guidelines for effectively coordinating feedback from project tracking into current and future estimates.
 - Construct and define methods for comparing costs and cost performance across different contractors and development organizations.
- Products:**
- Templates and defined process models for producing verifiable and repeatable software estimates.

- Guidelines and methods for comparing software costs across different contractors, work sites, and development organizations.
- Guidelines for initiating and sustaining improved software estimating capabilities.
- Guidelines for incorporating feedback from project tracking into current and future estimates.
- Criteria for effective estimating.
- Criteria for the effective communication and use of estimation results.

Thrust #4: Incorporate Estimating into Executive Decision-Making Processes

Develop guidelines, criteria, process templates, aids, and evaluation methods that managers can use to improve the effectiveness with which they obtain and employ estimates when planning and managing software efforts.

- Objectives:**
- Develop process models and guidelines for initiating and sustaining reliable and responsive software estimating capabilities.
 - Develop and provide guidelines for obtaining, using, and communicating software estimates.
 - Show managers and executives how reliable software estimates can be obtained and used to improve the management of software projects, development programs, and business areas.

- Issues:**
- Estimates cannot be viewed as satisfactory until they are successfully used. This thrust addresses methods for
 - providing and communicating requirement and design information to estimators
 - examining and establishing credibility of estimating results
 - integrating estimates into business and management practices
 - communicating rationales for estimates to other managers and to customers
 - establishing and sustaining effective estimating capabilities

- Actions:**
- Identify the practices that users of software estimates employ when incorporating estimates into their planning and decision making.
 - Develop examples and guidelines for establishing effective estimating capabilities.
 - Develop examples and guidelines for obtaining, evaluating, and applying software estimates.

- Develop and define process models and guidelines for effectively coordinating feedback from project tracking into current and future estimates.
- Develop guidelines and examples for using estimates to improve bid and proposal practices.
- Develop guidelines and examples for using estimates as vehicles for communicating with customers.
- Promote effective use of estimating practices in executive planning and decision making.

- Products:**
- Guidelines for initiating and sustaining improved estimating capabilities.
 - Guidelines for obtaining reliable software estimates.
 - Guidelines for using software estimates in bid and proposal activities.
 - Guidelines for using software estimates in project planning and tracking.
 - Guidelines for using software estimates in strategic and business area planning.
 - Presentations and articles that show how to establish and benefit from effective software estimating.

Thrust #5: Develop Cost Model Criteria

Apply the process criteria developed in Thrusts 3 and 4 to develop criteria and examples for designing, evaluating, and using software cost models.

- Objective:**
- Develop criteria for evaluating, selecting, and using software cost and schedule models.
- Issues:**
- This step is a prerequisite for evaluating software cost models. It provides criteria against which to assess cost model performance.
 - Criteria derived from defined process models will provide leverage for motivating vendors to improve cost estimating tools and services.
- Actions:**
- Identify the capabilities that tools and models must have if they are to support the needs of defined estimating processes.
 - Map these criteria to the objectives, methods, and processes stated or implied by the SEI capability maturity model.

- Identify key practices that users of cost models should follow when using different cost models to support defined estimating processes.

- Products:
- Criteria for evaluating and acquiring software estimating models and tools.
 - Criteria for using software estimating models and tools.
 - Criteria for improving upon today's software estimating models and tools.

Thrust #6: Evaluate Cost Model Capabilities

Evaluate the capabilities of existing software cost, schedule, and size models and show how these models can best be used as tools to support the processes identified in Thrusts 3 and 4. These evaluations will be guided by the criteria developed in Thrust 5 and structured to address the requirements and abilities of organizations operating at different maturity levels and in different development environments.

- Objectives:
- Evaluate the capabilities and performance of the principal models and tools that are used today to support software cost and schedule estimating.
 - Prepare examples that illustrate model strengths and weaknesses. Show where and how existing estimating models can be used to perform specific tasks and support defined estimating processes.

- Issues:
- Comparative evaluations of cost model capabilities would help organizations select the tools that are most effective for cost and schedule estimating.
 - Cost models offer frameworks for improving the repeatability, transferability, and communication of estimating methods and results. Model evaluations should address these issues.
 - Evaluations of cost models should address the abilities of the models to help professional estimators develop estimates for real software projects. Previous evaluations have focused almost exclusively on either features or accuracy. No one has examined cost models from the perspective of how well they help estimators execute defined processes.
 - Evaluations focusing on "accuracy" of estimates are potentially misleading and almost always inappropriate. Attributing accuracy to cost models assumes that responsibility for the quality of an estimate lies with the model. In reality, responsibility for quality of results lies with estimators and those who support them with training, tools, processes, and data.

- The purpose of this task is to assess the abilities of existing software cost models to support practicing estimators.
- Actions:**
- Evaluate the abilities of existing software cost models and tools to support the needs of estimating processes identified in Thrusts 3 and 4.
- Products:**
- Reports that evaluate cost model capabilities.
 - Illustrations of how existing models can be used to meet the needs of estimators and managers.
 - Summaries of the strengths and weaknesses of existing software estimating models and tools, with attention to how well they help organizations do their estimating jobs.
 - Identification of needs that are poorly met by today's estimating models and tools.
 - Guidelines and recommendations for acquiring and installing effective suites of software estimating models and tools.
 - Guidelines and practices for obtaining reliable estimates with existing software cost models and tools.

Thrust #7: Technology Transfer and Technology Transition

Establish and lead efforts that get improved estimating processes and tools into software engineering and executive decision-making practice. This thrust has four components.

Note: This plan distinguishes between transfer and transition as follows:

Technology transfer = moving existing technology into organizations where the technology has not previously been used.

Technology transition = moving new technology into organizations.

Task 7A: Assist Sponsors and Partners

Assist sponsoring organizations and industry partners in improving the processes, practices, tools, and data that they use for estimating the software costs and schedules.

- Objectives:**
- Assist sponsoring organizations and technical partners in improving the processes, practices, tools, and data they use when estimating costs, schedules, and sizes associated with developing and supporting software systems.

- Prototype and evaluate the materials, guidelines, and process models developed in Thrusts 2–6.
- Expedite the adoption of the estimating processes and practices developed under this initiative.

Issues:

- The methods, guidelines, templates, and recommendations developed in Thrusts 3–6 should be tested and evaluated in real software environments. Assisting sponsors and technical partners will provide locations and opportunities for this.
- SEI assistance can supply frameworks, structured processes, supporting materials, and legitimacy that help organizations get estimating process improvements underway.
- Assistance to sponsors and partners helps generate early success stories and examples that can expedite technology transfer and technology transition.
- Active assistance from the SEI is likely to accelerate the adoption of the products of this initiative.

Actions:

- Work with sponsoring organizations to install and evaluate defined estimating processes.

Products:

- Guidelines for installing and sustaining effective estimating processes (derived from lessons learned in actually doing it).
- Guidelines and advice for applying specific models to help with difficult estimating tasks.
- Criteria for success.
- Consulting, facilitating, and training assistance.
- Tutorials and course materials for cost estimators and for software engineering education programs.

Task 7B: Develop Educational Materials and Courses

Develop materials and courses that will assist managers, practitioners, and software engineering process groups in establishing and using defined processes for software estimating. Collect and devise examples that show how existing cost, schedule, and size models can best be used to improve estimating reliability and responsiveness. Assemble these materials in forms that will enable others to use them in software engineering and cost estimating courses.

Objective:

- Provide teaching materials that others within industry, government, academia, and the SEI can use to introduce improved estimating processes and practices into software development, support, and acquisition organizations.

- Issues:**
- Technology transfer and technology transition has to be supported by education and training.
 - Developers of new materials should organize and present the products of their work so that the methods and processes can be learned and employed by others.
 - Preparing materials for instructing others is an effective way to ensure internal self-consistency and avoid oversights.
 - The adoption and teaching of materials by others is one clear signal of success.
- Actions:**
- Prepare sequenced sets of visual materials and interactive exercises that can be used to educate and train managers and practitioners in improved software estimating processes and practices.
 - Organize and present prototype tutorials that test the quality and effectiveness of the training products developed.
 - Work with education and training professionals within the SEI to transition prototype tutorials into one or more formal courses in software estimating.
 - Work with members of the SEI Products and Services Division and with the Education and Training Review Board to see that existing SEI courses and executive seminars get updated to include appropriate materials on software estimating.
- Products:**
- Examples and illustrations that can be used in courses in software engineering, management, and estimating.
 - Tutorials that others can use as bases for developing full-scale courses in software cost and schedule estimating.
 - Improvements to the materials and examples on software estimating that are currently presented in SEI courses and executive seminars.

Task 7C: Prepare and Present Papers and Tutorials

Illustrate and make the case for effective estimating processes and practices.

- Objectives:**
- Test and evolve the education and training materials developed in Task 7A.
 - Establish and expand recognition of the SEI as a center of expertise in software estimating.

- Issues:**
 - Technology transfer and technology transition need delivery vehicles. Professional papers and presentations are two such vehicles.
- Actions:**
 - Prepare and present papers and tutorials that promote the use of the processes and practices identified and/or developed under this initiative.
- Products:**
 - Tutorials and papers delivered at conferences such as the SEI Software Engineering Symposium, Software Engineering Process Group National Meetings, Tri-Ada, the DoD Cost Analysis Symposium, the COCOMO Users' Group, and meetings of the IEEE, the International Society of Parametric Analysts, and the Society of Cost Estimating and Analysis.
 - Technical reports.
 - Papers and articles for professional journals.

Task 7D: Focus and Accelerate the Development of Improvements to Software Cost, Schedule, and Sizing Models

Work with cost model vendors to improve the tools they produce and market. Through persuasion and encouragement, motivate model evolution and guide it in directions that provide improved support for defined estimating processes and practices.

- Objectives:**
 - Through persuasion and encouragement, accelerate the evolution of software cost modeling and guide it in directions that provide tools for supporting improvements in defined processes and practices.
 - Expedite the adoption of the estimating processes and practices developed under this initiative.
- Issues:**
 - Tools should support the needs of defined processes.
 - Availability of tools often determines the feasibility of alternative estimating practices and processes.
 - Accelerating the development of automated support for cost estimating needs is an effective way to speed the adoption of improved estimating processes and practices. Without automation, many good practices may not be feasible.
- Actions:**
 - Use the needs, criteria, and lessons learned from the other tasks to guide and persuade cost model vendors to improve the tools they market.

- Products:**
- Guidance for vendors and developers of cost models that helps them produce products that meet the management needs of the software engineering community.
 - Accelerated evolution and improvement of software estimating tools and products.

Future Work

The two thrusts that follow are deferred not because they are deemed unimportant, but because the environments they address are not likely to be as productive as acquisition and development environments for identifying good estimating processes and practices. Also, addressing these thrusts concurrently with the seven already listed would require more resources than are presently available.

Future Thrust #1: Estimating for Post-Deployment System Support

- Objective:**
- Identify and describe the current software estimating processes, practices, and tools used to address post deployment software support (PDSS).
 - Develop process models and methods for getting effective software estimating practices into use and institutionalized in PDSS environments.
- Issues:**
- PDSS environments often differ materially from development environments. Many PDSS projects must address factors and influences that are not present during system development.
 - Costs for PDSS often exceed costs for initial development.
 - PDSS environments differ from development environments in that opportunities exist for basing estimates on empirical data gathered from the very systems that are to be modified. This and other characteristics to be accounted for in PDSS environments can lead to processes and criteria that go beyond those applicable to software development.
- Actions:**
- Conduct field investigations with PDSS organizations to identify the processes, practices, and tools that are used to produce software estimates today.
 - Identify where and how software estimates are used, and where the quality and responsiveness of these estimates could be improved.
 - Identify unfulfilled estimating needs.
 - Identify opportunities for improved estimating processes and tools.
- Products:**
- Baseline descriptions of PDSS estimating processes and practices.
 - Lists of unfulfilled needs and opportunities for improvement.

- Process models for use in installing and sustaining improved software estimating capabilities.
- Process models for producing verifiable and repeatable software estimates.
- Criteria for evaluating and acquiring software estimating models and tools.
- Criteria for creating improved software estimating models and tools.

Future Thrust #2: Estimating for Internal Systems Development

- Objectives:**
- Identify the estimating processes, practices, and tools used by organizations when developing software systems for their own use.
 - Develop process models and methods for getting effective software estimating practices into use and institutionalized in in-house development environments.
- Issues:**
- The characteristics of in-house development environments often differ from those of contracted development environments. Processes used in in-house development are often less formal than in contract-driven projects.
 - Products produced and lessons learned in Thrusts 2-7 should provide useful foundations for improving estimating processes and practices in in-house software environments.
- Actions:**
- Conduct field investigations to identify the estimating processes, practices, and tools that organizations use when producing software for internal use.
 - Identify where and how software estimates are used for planning and managing development and support of local software systems.
 - Identify how the quality and responsiveness of these estimates are viewed.
 - Identify unfulfilled estimating needs.
 - Identify opportunities for improving in-house estimating processes and tools.
- Products:**
- Baseline descriptions of internal estimating processes and practices.
 - Lists and descriptions of unfulfilled needs and opportunities for improvement.
 - Process models for use in installing and sustaining improved software estimating capabilities.

- Process models for producing verifiable and repeatable software estimates in in-house environments.
- Criteria for evaluating and acquiring software estimating models that support development, maintenance, and enhancement of software products for internal use.
- Criteria for creating and improving software estimating models and tools.

Resources and Funding

A. Project Staffing:

The staffing levels that will be assigned by the SEI to this initiative will depend on the levels of funding support received from sponsors and collaborators. We estimate that at least 2.25 technical staff-years per year are needed to successfully sustain the initiative.

Assuming adequate funding, two members of the Software Measurement Project will be assigned to the initiative. Other technical staff will be drawn upon as specialized talents are needed, for a total level of support of 2.25 technical staff-years per year. These resources will be supplemented by resident affiliates and technical collaborators from government and industry.

One member of the administrative staff will provide secretarial services and meeting coordination. This will be a part-time assignment.

B. Support Services:

SEI Information Management staff members will assist in planning and editing technical reports and training materials. These staff members will also provide review and advice for presentations and other project deliverables.

SEI Program Development and Human Resources staff will assist in obtaining sponsors, establishing technical collaboration agreements with industry and government partners, and recruiting and providing facilities for resident affiliates.

C. Funding:

1993: 100% Core.

1994: 25% Core, 75% TO&P.

Future: Transitioning toward 100% TO&P (supplemented by resident affiliates).

D. Staffing requirements:

1993:	18 staff-months
1994:	27 staff-months
1995:	27 staff-months
1996:	27 staff-months

Products and Deliverables

The principal products and deliverables that we have identified at this point are listed below in the order of the thrusts they support. Milestones and schedules will be adjusted to fit the funding provided by SEI sponsors and the case-by-case arrangements that will be made to integrate the efforts of technical collaborators who join the initiative after it is underway.

1. Enlist sponsors & partners and establish working relationships
 - Identifications of sponsors, technical partners, and sources of information
 - Plans and presentations that lay foundations for collaboration and cooperation with sponsors and technical partners
 - Technical collaboration and nondisclosure agreements with participating organizations
2. Identify current estimating processes, practices, tools, and opportunities for improvement
 - Mappings of the estimating processes and practices used by sponsors and technical collaborators (proprietary information, for use by individual sponsors and technical collaborators)
 - Opportunities for improvement in the processes used by sponsors and collaborators (proprietary information)
 - Benchmarks and comparative summaries of current estimating practice
 - Needs analyses
 - Preliminary report
 - Formal report
3. Develop defined process models and estimating guidelines
 - Defined estimating processes for sponsors and technical collaborators
 - Process models, templates, and guidelines for producing and communicating verifiable software estimates
 - Criteria for reliable estimating process
 - Preliminary report
 - Formal report

4. Incorporate estimating into executive decision-making processes
 - Guidelines, criteria, process templates, and evaluation methods for managers to use for improving their use of estimates in planning and managing software development efforts
 - Preliminary report
 - Formal report
5. Develop cost model criteria
 - Criteria for selecting and using software cost, schedule, and sizing models
 - Preliminary report
 - Formal report
6. Evaluate cost model capabilities
 - Evaluations of the abilities of existing software cost, schedule, and sizing models to support the needs of the defined estimating processes identified in Thrusts 3–5
 - Preliminary report
 - Formal report
7. Technology transfer and technology transition
 - A. Assistance to sponsoring and collaborating organizations in installing and sustaining defined estimating processes and practices
 - Process mapping and definition
 - Piloting, prototyping, and assistance to SEPGs and estimating activities
 - Continuing support and technology transition
 - B. Educational materials, tutorials, and courses in software estimating
 - Prototype materials and tutorials
 - Transition to Products & Services
 - C. Presentations of papers and tutorials that explain estimating processes and practices to leaders and practitioners in industry, government, and academia
 - D. Work with cost model vendors to design and initiate improvements in commercial cost, schedule, and size estimating models

Appendix C: Templates for Technical Collaboration Agreements

This appendix provides the 10 March 94 version of the templates for establishing working agreements between the SEI and collaborating organizations. SEI projects and their technical collaborators should use these templates (or more recent versions) to describe the joint efforts they plan to pursue.

TECHNICAL COLLABORATION AGREEMENT
between the
Software Engineering Institute, <SEI Project Name>
and
<Collaborating organization>

<Day Month Year>

The Software Engineering Institute (SEI) is a federally funded research and development center (FFRDC) established and operated by Carnegie Mellon University (CMU) and sponsored by the Department of Defense (DoD).

<Statement briefly describing collaborating organization, similar to the above statement describing the SEI.>

It is the purpose of this Technical Collaboration Agreement (TCA) to.....**<provide a description of work here. This paragraph should be two to three sentences. Word it carefully, as this statement will be included in public TCA reports as the summary description of this TCA.>** This work is described in detail in Attachment 1 to this agreement.

The goals of the SEI in this collaboration are:

- **<goal 1>**
- **<goal 2>**
- **<goal 3>**
- **<etc.>**

The goals of <collaborating organization> in this collaboration are:

- **<goal 1>**
- **<goal 2>**
- **<goal 3>**
- **<etc.>**

Both parties agree that they will not:

- institute against either party any suit or action at law or otherwise, nor in any way aid the institution or prosecution of any claim, demand, action, or cause of action for damages, loss of service, expenses, or compensation for or on account of the performance under this agreement;
- claim, on the basis of this agreement, the endorsement of either party in any promotion of products or services that incorporate the technologies of either party.

Upon signing of this Technical Collaboration Agreement, the SEI and **<collaborating organization>** may state that they are interacting via a TCA.

Nothing in this Technical Collaboration Agreement will grant to either party the right to make commitments of any kind for or in behalf of the other party. This Technical Collaboration Agreement is not intended as, nor will it be considered, as a "team agreement", joint venture, partnership, or other formal business organization. Unless otherwise agreed, in writing, neither party will have the right or obligation to share any of the profits or bear any of the risks or losses of the other party. At all times the parties will remain independent contractors with each responsible for its own employees and representatives. The parties assume no responsibility to the other for costs, expenses, risks, and liabilities, associated with the research, development, exchange, and use of the other's proprietary information.

No rights or obligations other than those expressly recited herein are to be implied from this Technical Collaboration Agreement, including any requirement that the parties contract with each other for the procurement of any products, services, or data resulting from this Technical Collaboration Agreement

<Include the following three paragraphs that address intellectual property issues if this TCA is NOT part of a Strategic Collaboration Agreement (SCA). If the collaborator has concerns about the next three paragraphs in reference to intellectual property issues contact Valerie Weidman for guidance.>

This agreement does not give either party any rights in the other's intellectual property.

The SEI will disclose to **<collaborating organization>** any Defense Department requirement respecting ownership of intellectual property developed in the course of performing a project.

Unless a collaboration agreement specifies otherwise, each party will own all rights in any intellectual property created solely by its employees in the course of performing a project, and each party may exploit with attribution, but without any duty to account to the other for any revenues received, any intellectual property created jointly by employees of both parties in the course of performing a project.

Finally, it is mutually agreed that;

NOTE: (this paragraph of information should be deleted from final draft before signatures are obtained) Use the first of the following two paragraphs if this TCA needs to reference a non-disclosure agreement. The SEI does not require the non-disclosure agreement; it is signed at the customer's request. SEI non-disclosure agreements are available in Cindy Nesta's public folder. Please note that only Valerie Weidman is authorized to sign non-disclosure agreements on behalf of the SEI. If the collaborator prefers to execute their organization's non-disclosure agreement contact Valerie Weidman for review and signature.

Any publication of results of this collaboration by the SEI will honor the confidentiality requirements of the attached non-disclosure agreement signed by the **<collaborating organization>** and the SEI.

OR (Only one of these paragraphs is necessary. Please delete this paragraph and whichever paragraph is not appropriate.)

Any disclosure by one party to another of confidential or proprietary information will only be made after a separate agreement covering such disclosure has been duly signed by authorized representatives of both parties.

Changes to this agreement will not be effective unless presented in writing and signed by both parties.

TECHNICAL COLLABORATION AGREEMENT
between the
Software Engineering Institute, <SEI Project Name>
and
<Collaborating organization>

Either party may terminate this Technical Collaboration Agreement upon thirty (30) days written notice to the other party.

This agreement will be effective for a term of <X> months, from <start date> to <end date> and may be extended by mutual consent in writing.

Agreed by:

Software Engineering Institute

<collaborating organization>

<program manager name>
<program> Program Manager
date:_____

<name>
<title>
date:_____

Julia Allen
Program Development Division
date:_____

<name>
<title>
date:_____

Attachment 1

1. Background and History

Summarize the SEI and collaborator interactions to date. All major historical events/interactions leading to this agreement should be noted here. (For example, "An SEI-led assessment took place in 1987.")

2. Task Descriptions

Summarize the tasks to be performed as part of the collaboration agreement.

2.1 Collaborator Point of Contact

All correspondence regarding this agreement should be directed to:

Name: < >

Address: < >

Phone: < >

Fax: < >

Email: < >

Overnight address: < >

2.2 SEI Point of Contact

All correspondence regarding this agreement should be directed to:

Name: < >

Address: Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Phone: < >

Fax: (412) 268-5758

Email: < >

Overnight address: 4500 Fifth Avenue
Pittsburgh, PA 15213

2.3 Technical Tasks

Describe the tasks to be undertaken by the collaborator; provide as much detail as practical.

Describe the tasks to be undertaken by the SEI; provide as much detail as practical.

2.4 Deliverables

Identify all deliverables and the responsible party.

2.5 Project Review Meetings Schedule

Regular project reviews should be scheduled and listed here. Review meetings should be documented via a written agenda and minutes.

2.6 Project Schedule and Milestones

Describe schedule and milestones.

3.0 Assumptions and Constraints

Document any assumptions made by the SEI or the collaborator that are key to the success of this collaboration.

4.0 Collaborator Resource Planning

Document the level of effort (described in staff hours per month) that the collaborator is contributing.

Collaborator staff hours contributed per month: < >

5.0 Cost Recovery (this section is only included if cost recovery is part of this TCA)

If applicable, document all cost recovery funds shown as a not-to-exceed (NTE) figure.

Costs incurred as part of this collaboration agreement will not exceed <\$ > for the period beginning < > and ending < >.

5.1 The following costs are included in the not-to-exceed (NTE) amount:

Labor < > (This NTE amount covers < > SEI staff days at \$1,500 per day)

Travel Expenses < >

Course Costs < >

Materials < >

5.2 SEI will bill collaborator:

Upon completion of the terms of this agreement < >

On a quarterly basis < >

On a monthly basis < >

Invoices will be submitted by CMU/SEI to the following address:

Address: < >

Phone: < >

Fax: < >

Email: < >

Overnight address: < >

ATTENTION: < >

5.3 Payment will be remitted to:

**Carnegie Mellon University
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213**

ATTENTION: Bernadette Ledwich

PLEASE REFERENCE SEI INVOICE NUMBER ON ALL PAYMENTS

6.0 SEI Resource Planning

Document the level of effort (**described in staff hours per month**) that the SEI is contributing. (This should not include any SEI effort that the collaborator is being charged for under cost recovery.)

SEI staff hours contributed per month: < >

Nondisclosure

The SEI does not require a nondisclosure agreement. However, If a collaborator requests a nondisclosure agreement, the SEI suggests the standard form shown on the following pages. If a collaborator prefers to execute their organization's nondisclosure agreement, the SEI coordinator should contact Valerie Weidman at the SEI for review and signature.

SEI Nondisclosure Agreement

This Agreement entered into as of this _____ day of _____, 19____, by and between the Carnegie Mellon University Software Engineering Institute (SEI) and _____ (DISCLOSER), having an office at _____. The SEI is a federally funded research and development center (FFRDC) established and operated by Carnegie Mellon University (CMU) and sponsored by the Department of Defense (DoD).

The parties agree:

1. DISCLOSER intends to disclose certain information in written or other tangible form to SEI pursuant to this Agreement that may be of a proprietary nature. Any verbal, or other tangible, information shall be summarized in writing by DISCLOSER to SEI within 10 days of the original disclosure date. All information furnished pursuant to this Agreement shall be marked with a proprietary notice.
2. SEI agrees not to disclose any such information received from the DISCLOSER to any third party, except as required by applicable law or legal process, and shall use the same degree of care to avoid disclosure of such information as it employs with respect to its own proprietary information.
3. Any information disclosed hereunder shall not be deemed to be confidential or proprietary and SEI shall have no obligation with respect to any such information which:
 - a. was known to SEI at the time it was submitted, or
 - b. is, or becomes, publicly known through no wrongful act of SEI, or
 - c. is received by SEI from a third party without similar restrictions and without breach of this Agreement, or
 - d. is approved for release by written authorization of DISCLOSER, or
 - e. is independently developed by SEI without the use of the information disclosed hereunder, or
 - f. is furnished by DISCLOSER to a third party without a similar restriction on the third party's rights.
4. SEI shall not be liable for:
 - a. an inadvertent disclosure of the information provided that it uses the same degree of care in safeguarding such proprietary information as it uses for its own proprietary information, and upon discovery of such inadvertent disclosure of such information it shall endeavor to prevent further inadvertent disclosure;
 - b. an unauthorized disclosure of such information by persons who have been in its employ unless it failed to safeguard such information with the same degree of care that it uses for its own proprietary information.
5. The term of confidentiality shall be three years from the disclosure date unless extended by mutual agreement. Thereafter, any and all rights of each party, with respect to the subject matter of this Agreement, shall be determined solely in accordance with such patent rights or copyrights which a party hereto may have or acquire.
6. This agreement will be effective when signed by duly authorized representatives of both parties and will be executed in two counterparts, each of which will be considered an original.

-
- This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Date: _____

Acknowledgment of Received Materials and Information

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Date: _____

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS None		
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-94-SR-03			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute		6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office		
6c. ADDRESS (city, state, and zip code) Carnegie Mellon University Pittsburgh PA 15213			7b. ADDRESS (city, state, and zip code) HQ ESC/ENS 5 Eglin Street Hanscom AFB, MA 01731-2116		
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office		8b. OFFICE SYMBOL (if applicable) ESC/ENS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962890C0003		
8c. ADDRESS (city, state, and zip code)) Carnegie Mellon University Pittsburgh PA 15213			10. SOURCE OF FUNDING NOS.		
			PROGRAM ELEMENT NO 63756E	PROJECT NO. N/A	TASK NO N/A
			WORK UNIT NO. N/A		
11. TITLE (Include Security Classification) Software Cost and Schedule Estimating: A Process Improvement Initiative					
12. PERSONAL AUTHOR(S) Robert E. Park, Wolfhart B. Goethert, and J. Todd Webb					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM TO		14. DATE OF REPORT (year, month, day) May 1994	
15. PAGE COUNT 100 pp.					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB. GR.	capability maturity model technical collaboration agreement		
			process improvement		
			software cost estimating		
			software schedule estimating		
19. ABSTRACT (continue on reverse if necessary and identify by block number) This report describes efforts that have been initiated by the Software Engineering Institute to improve the practice of software cost and schedule estimating. These efforts involve support and participation from both industry and government. They are motivated by the capability maturity model, which identified the key roles estimating and cost management play play in establishing repeatable software processes. Products from the initiative will include templates, criteria, and guidelines for establishing defined estimating processes, training materials, and examples for teaching good estimating practice, and evaluations of the abilities of contemporary cost models to meet today's estimating needs. (please turn over)					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☒			21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution		
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas R. Miller, Lt Col, USAF			22b. TELEPHONE NUMBER (include area code) (412) 268-7631		22c. OFFICE SYMBOL ESC/ENS (SEI)

ABSTRACT — continued from page one, block 19