REUSE ADOPTION GUIDEBOOK

SPC-92051-CMC

VERSION 02.00.05

NOVEMBER 1993

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1993	Technical Report

4. TITLE AND SUBTITLE

Reuse Adoption Guidebook

5. FUNDING NUMBERS

G MDA972-92-J-1018

6. AUTHOR(S)

Produced by Software Productivity Consortium under contract to Virginia Center of Excellence

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Virginia Center of Excellence
SPC Building
2214 Rock Hill Road
Herndon, VA 22070

8. PERFORMING ORGANIZATION REPORT NUMBER

SPC-92051-CMC,
Version 02.00.05

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

ARPA/SISTO
Suite 400
801 N. Randolph Street
Arlington, VA 22203

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Supersedes the Reuse Adoption Guidebook Version 01.00.03 (DTIC # ADA259405)

12a. DISTRIBUTION / AVAILABILITY STATEMENT

No Restrictions

12b. DISTRIBUTION CODE

1

13. ABSTRACT (*Maximum 200 words*)

The Reuse Adoption Guidebook was produced for use by organizations to assist them in improving their competitiveness through the adoption and institutionalization of software reuse technology. Reuse refers to the use of a software asset in the solution of different problems or different versions of a problem. While many forms of reuse technology are currently available and the technology can address many different organizational objectives, crafting the opportunities into an implementable program that meets a particular organization's objectives is a difficult task. This guidebook prescribes a reuse adoption process to help perform that task.

A key component in the reuse adoption process is a Reuse Capability Model, which is a tool that can aid an organization in understanding its current capabilities and in establishing goals for its reuse adoption process.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Reuse, technology transfer, reuse capability model, reuse adoption, domain assessment, risk assessment, economics models	277
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

REUSE ADOPTION GUIDEBOOK

SPC-92051-CMC

VERSION 02.00.05

NOVEMBER 1993

Produced by the
SOFTWARE PRODUCTIVITY CONSORTIUM SERVICES CORPORATION
under contract to the
VIRGINIA CENTER OF EXCELLENCE
FOR SOFTWARE REUSE AND TECHNOLOGY TRANSFER

SPC Building
2214 Rock Hill Road
Herndon, Virginia 22070

CONTENTS

FIGURES

TABLES

This page intentionally left blank.

PREFACE

The technology described in this guidebook is part of a broad approach to software productivity improvement. This preface provides an overview of that approach and identifies the series of guidebooks that support it. These guidebooks were developed by the Software Productivity Consortium under contract to the Virginia Center of Excellence for Software Reuse and Technology Transfer (VCOE). For a complete listing of VCOE guidebooks and products, call the Software Productivity Consortium's Technology Transfer Clearinghouse at (703) 742-7211.

Each technology has been packaged so it can be used without reference to the other technologies. However, it is also possible to combine these technologies into an integrated approach for product development. Figure P-1 shows how the guidebooks for these technologies relate to the practices of software development organizations.

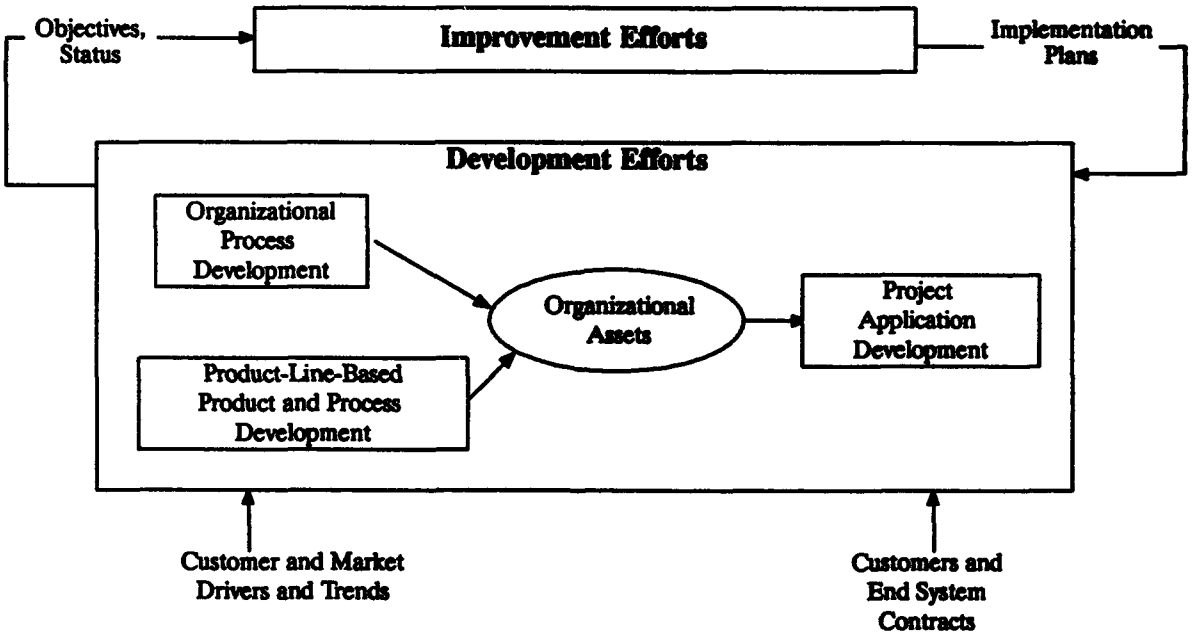


Figure P-1. Structure for Integrated Application of Consortium Technologies

These practices are composed of:

- ***Improvement Efforts (IE).*** Application of technology to improve software development efforts. These efforts require managed approaches to assessment of objectives and current capabilities, planning for the improvement, implementation of the plan, and measurement of success.

- ***Development Efforts.*** Development of products that meet the needs of customers and markets or products that make the organization more competitive in meeting expected future needs.
  - ***Organizational Process Development (OPD).*** Development of standardized organizational process assets (e.g., process and method descriptions, process enactment tools) tailored for a particular organization.
  - ***Product-Line-Based Product and Process Development (PLD).*** Development of integrated product and process assets (e.g., core products and processes for adapting them for particular customer needs) appropriate for a particular product line.
  - ***Project Application Development (PAD).*** The tailoring and application of organizational assets for a particular product development effort.

Table P-1 describes how existing products can be integrated to address your organizational practice.

Table P-1. Consortium Guidebooks and Related Practices

Guidebook	Part Number	Relationship to Software Practice
CoRE: An Object-Oriented Requirements Method	SPC-92060-CMC	Used for defining and analyzing requirements in PAD. Adaptable for use in PLD.
Managing Process Improvement: A Guidebook for Implementing Change	SPC-93105-CMC	Supports IE by providing a process and supporting guidance for initiating and maintaining an organizational process improvement program.
Process Definition and Modeling Guidebook	SPC-92041-CMC	Provides methods for defining and documenting processes so they can be analyzed, modified, and enacted. Supports IE and OPD.
Process Engineering with the Evolutionary Spiral Process Model	SPC-93098-CMC	Used to iteratively plan, manage, and control PAD and PLD. Used to construct organization-specific processes in PLD and tailor them in PAD.
Reuse Adoption Guidebook	SPC-92051-CMC	Supports IE by providing specific process improvement activities for incorporating reuse practices.
Reuse-Driven Software Processes Guidebook	SPC-92019-CMC	Provides development approaches for PLD (domain engineering) and PAD (application engineering) of reusable software assets.
Software Measurement Guidebook	SPC-91060-CMC	Supports IE by providing methods for quantitative assessment of project status.
Using New Technologies: A Technology Transfer Guidebook	SPC-92046-CMC	Supports IE by providing a process that addresses how to get an organization to use new technologies.

This guidebook is the second release of the Consortium's approach to institutionalizing and improving an organization's software reuse practice. This revised version of the guidebook incorporates the lessons learned from applying the domain and reuse capability assessments in three best-of-breed case studies.

For the case studies, the Consortium sought organizations that have successfully institutionalized reuse. The objective was to get the feedback of an experienced organization on the Consortium's reuse adoption technologies. Each of the case study organizations had been practicing reuse for at least 5 years in the development or maintenance of customer products. The case studies included organizations that developed systems for the Department of Defense and organizations that developed commercial products, they included both small organizations (less than 50 staff) and large organizations (more than 1000 staff), and they included organizations in the embedded systems business as well as organizations in the information systems business.

This guidebook includes improvements to the Reuse Capability Model (RCM), an elaborated Domain Assessment Model, and adds a reuse assessment process. This version also addresses the plans for integration of process improvement technologies produced by the VCOE. Specific areas of improvement include:

- ***Reuse Adoption Process.*** The Reuse Adoption process is a decision-support and planning process for defining, developing, and implementing a plan to institutionalize reuse. The process provides step-by-step guidance on initiating a reuse program, defining reuse adoption objectives and goals, assessing the current situation with respect to reuse, developing reuse adoption strategies, analyzing and selecting reuse adoption strategies based on risk and economic viability, developing a reuse action plan, and implementing and monitoring the reuse action plan.

  In this version, the underlying technologies of the Reuse Adoption process have been integrated with the process improvement product line, including a common set of activities for Process Improvement and Reuse Adoption processes. This integration resulted in the renaming of activities and the addition of the Review and Update Reuse Program activity and the reuse program plan.

- ***Domain Assessment Model and Assessment Process.*** The Domain Assessment Model identifies factors influencing a business area's opportunities for reuse. The assessment process provides a mechanism for qualitatively estimating an organization's potential for reuse based on the Domain Assessment Model. The process helps managers and engineers to determine where the best opportunities are for reuse within the business area, what realistic goals are for a reuse program, and what the impact is of reuse on product development.

  The Domain Assessment Model is further elaborated in this version; its factors are clarified and decomposed into a set of attributes for use in the assessment. The assessment process is new material. The model and process have been updated to incorporate feedback from the case studies.

- ***Reuse Capability Model and Assessment Process.*** The RCM identifies the factors critical to improving an organization's reuse capability and prioritizes the factors in a manner that reduces an organization's risk. The RCM and assessment process are used to gain an understanding of an organization's process with respect to reuse (identifying its strengths and improvement opportunities) and to establish improvement goals, as well as the staging of those goals, in a manner that mitigates risk.

The RCM has been updated to reflect feedback from the case studies, including detailed explanations with examples of the critical success factors to improve understandability. An Intergroup Coordination critical success factor has also been added to the model. The assessment process is new material and has also been updated to incorporate feedback from the case studies.

- ***Examples and Artifacts.*** Partial examples are provided to illustrate the activities of the Reuse Adoption process. In addition, annotated outlines are provided for key process artifacts to serve as a model for managers and engineers applying the process to their organization, including the assessment findings report and reuse action plan.

  The annotated outlines are new additions to the guidebook. The examples are updates to version 1 reflecting the new assessment processes.

In addition, some material has been removed from this version of the guidebook. The material in Appendix E (Example Work Products) of version 1 was revamped and included in the body of this guidebook. Appendix F (Synthesis Family of Reuse-Driven Software Development Processes) of version 1 was removed in favor of a reference to the 1993 *Reuse-Driven Software Processes Guidebook* (this year's version of the *Domain Engineering Guidebook*). Appendix G (Reuse Economics Models) of version 1 was deleted in favor of a reference to the user manual for the Reuse Economics Spreadsheet Model, which contains a more up-to-date discussion of the same material.

ACKNOWLEDGMENTS

The Consortium wishes to thank the many people who have contributed to the development of this guidebook. We thank:

- Participants in the three reuse adoption case studies for their effort in conducting the assessments and their suggestions for improving the assessment process.
- Ted Davis, Fred Hills, and Roger Williams for authoring this version of the guidebook.
- Rich McCabe and Roger Williams for managing the development of this guidebook.
- Grady Campbell, Sam Redwine, and Wil Spencer for their thoughtful and thorough review of this guidebook.
- Mary Mallonee and Debbie Morgan for coordinating, editing, and processing this document.

This page intentionally left blank.

1. INTRODUCTION

As demonstrated in European, Japanese, and U.S. organizations, the ability to reuse significant portions of existing software assets offers significant potential for increasing engineering productivity, speed-to-market, and system quality and for decreasing the costs of building and maintaining large, software-intensive systems. However, instituting an effective reuse practice in software development poses substantial challenges to an organization. These challenges derive from a wide variety of technical, managerial, economic, social, and legal factors, which must be addressed if organizations are to succeed.

This guidebook assists your organization in meeting the "reuse challenge" through a well-defined approach for adoption and institutionalization of software reuse technology to improve productivity, quality, and competitiveness. This guidebook will help you:

- Understand your organization's business environment (e.g., market, competition, technology) and estimate the potential for reuse in your business area.
- Understand your organization's abilities (e.g., process, tools, skills, culture) to practice reuse.
- Initiate, plan, and implement a program to improve your organization's ability to effectively exploit the potential for reuse in your business area (Figure 1-1).

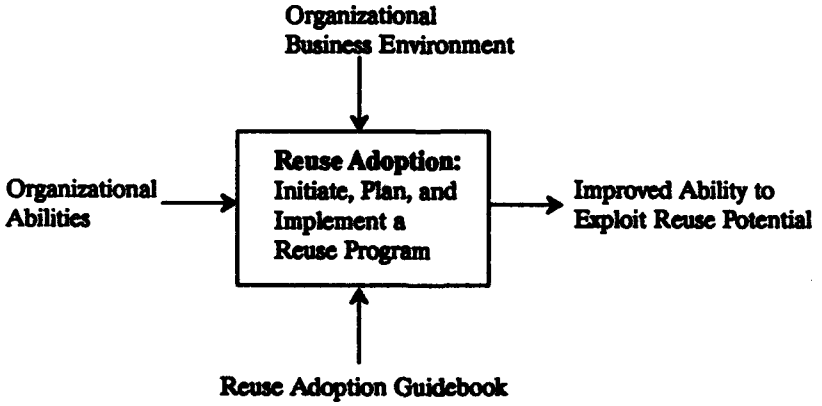


Figure 1-1. Reuse Adoption

Reuse adoption is envisioned as part of a continuous process improvement program; thus, the guidebook assists the end user in identifying both near- and long-term needs and actions. It provides guidance for developing a plan that will put the technology into practice, and it provides guidance on how to address critical institutional barriers, risks, and planning areas that must be addressed to make a reuse effort successful.

This section provides an overview of the guidebook. It identifies the guidebook's purpose, intended audience, and use. Also, it defines reuse and addresses why you should consider implementing a reuse program now and why you should use this guidebook.

1.1 AUDIENCE

The primary audience for this guidebook is business organizations that develop or maintain multiple software systems, versions, or products for one or more customers, i.e., product lines, and believe that software reuse technology may help their organizations become more competitive and better able to meet their goals but that:

- Need to better understand the benefit of reuse technology for their organizations
- Need support in identifying and putting into place required reuse-related changes

These business organizations include:

- Companies that contract with the Department of Defense (DoD) and other government agencies
- Commercial software product developers
- Government agencies, such as the DoD, intending to institute reuse into their development and acquisition processes
- Organizations with no established reuse practice that want to adopt reuse
- Organizations that have an established reuse practice and want to improve

Specifically, this guidebook is used by those individuals within an organization who are responsible for defining and implementing the organization's reuse program, i.e., an organizational effort to improve and institutionalize the practice of reuse in your organization. This guidebook can also be used by those individuals who advocate reuse, control organizational resources, or are affected by a reuse program (see Section 1.5).

1.2 WHAT IS REUSE?

Reuse, as used in this guidebook, refers to the use of an asset in the solution of different problems or different versions of a problem. Reuse may occur within a system (e.g., within a telephone switching system), across similar systems (e.g., commercial and military aircraft tracking systems), or in widely different systems (e.g., user interface services for telephone switches and aircraft tracking systems) (DoD 1992c). A reusable asset is any tangible resource that may apply to the solution of multiple problems, such as specifications, designs, code, test cases, etc.

Reuse technology comes in many forms, including process and method descriptions, common architectures, commercial off-the-shelf (COTS) components, languages and supporting development environments, component library mechanisms, reverse and reengineering tools, and application generators. Additionally, reuse technology can be used to meet many different organizational objectives, including quality improvement, reduced time-to-market, system upgrades and maintenance, downsizing, interoperability, and common product-line look and feel.

1.3 WHY SHOULD YOU WANT TO REUSE?

Software reuse has been propounded for years as a key to increasing software productivity. Also, in many businesses, ad hoc reuse is a common occurrence. So, you may be asking: "Why develop an organized reuse program now? What has changed?" The reasons are:

- Successes have been reported, resulting in improved productivity, quality, and competitiveness.
- Reuse can improve competitiveness.
- Process technology advances facilitate reuse.
- Software product areas have matured, making reuse feasible for many domains.
- Underlying technology has matured to effectively support reuse.
- U.S. government pull for reuse technology has increased.

Each of these points is expanded in the sections below. In addition, Hooper and Chester (1990) provide a cogent summary of research and practice in software reuse.

1.3.1 Successes Have Been Reported Resulting in Improved Productivity, Quality, and Competitiveness

Many organizations are incorporating reuse into their software development processes, and positive results are being reported. These include:

- The U.S. Navy Fleet Combat Direction System Support Activity restructured its shipboard tactical data system maintenance activity to provide a high degree of reuse of common components, and it automated the production of system builds from the reuse component repository (DoD 1992b). As a result, the Navy experienced a three-to-one payoff in trouble report corrections (STARS 1992a).
- TRW created a set of Network Architecture Services components that can be reused across a wide range of expected systems (Royce 1990). As a result, TRW won a major DoD contract.
- Idaho National Engineering Laboratories developed an architecture-based reuse library (AdaSAGE) that they use in development of energy control systems and sell to third parties for use in constructing similar systems. The laboratories have achieved reuse levels of 70% (Jensen, Stewart, and Whittington 1992).
- The U.S. Army Communications Electronics Command developed a standard architecture and components for Command and Control Systems that were used in development of multiple systems (DoD 1992a).
- Toshiba Software Factory reported 50% reuse over its product line in 1989 and increased productivity by 57% (Cusumano 1991).

- Hughes Aircraft Company, Command and Control Systems Division, created a Common Air Defense Ground Environment and reported realizing a 37% savings over projected development costs while building two systems (Benson 1991).

- Boeing Defense and Space Group used the Consortium's Synthesis process (Software Productivity Consortium 1993b) to improve its reuse capability in its modular flight simulator program (Freemon and Crispen 1992).

- Rockwell International (Software Productivity Consortium 1992a) used the Synthesis process to create an adaptable requirements specification for a part of its communication control domain. Users specified an individual system in terms of critical requirements and engineering decisions and were able to generate corresponding work products from the adaptable specification.

- Hewlett-Packard Analytical Product Group initiated a multisite reuse effort that developed assets that could be reused without modification and covered about 10% of lines of code released with the group's software products (Martin, Jackoway, and Ranganathan 1991).

1.3.2 Reuse Can Improve Competitiveness

Increasing international competition emphasizes that commercial organizations have a continuing need to bring advancing technology to bear on their software development efforts. Similarly, many defense contractors find themselves in difficult times due to decreasing defense budgets. One ingredient in keeping your competitive edge will be your ability to leverage existing assets (both staff knowledge and software artifacts) to more effectively meet the needs of future customers. While reuse is not a silver bullet, reuse technology can:

- ***Provide Increased Productivity and Faster Time-to-Market.*** Reuse technology allows you to avoid effort previously required for system production. Reuse of software assets will result, after initial investments have been made, in a decreased cost for system development and a reduced development schedule.

- ***Provide Improved Quality.*** Software that has been used previously and has been tested by a previous project will have fewer latent faults than new software. Additional faults found and corrected upon reuse will further increase the quality for the next user.

- ***Provide Early Requirements Validation.*** Off-the-shelf assets can be reused during the concept definition and requirements development efforts to construct functional prototypes. Customer review of the prototypes can lead to an earlier, clearer understanding of customer needs.

- ***Reduce Risk by Avoiding New Development.*** Risk is minimized by having an existing software asset that has addressed the areas where risk would otherwise lie (i.e., existing software has already addressed the related risks). Note that there is risk in reusing a piece of software with which you are unfamiliar (i.e., software that you did not develop), but you can estimate the expected value of the risk and use this in your cost or contingency planning.

- ***Retain and Leverage Technical Expertise.*** When experienced engineers use their specialized knowledge to create reusable assets, less experienced engineers (of which there is usually a

greater supply) can apply the assets to create a greater number of systems. Also, some of the specialized expertise may be transferred to the less experienced staff.

1.3.3 Process Technology Advances Facilitate Reuse

Process discipline is required for effective approaches to reuse. These "systematic" approaches require the development and management of quality work products over long life cycles for multiple end users. The required process discipline has not been a tradition in most software development organizations.

However, in the last decade, process technology has advanced as a driver of competitiveness. Process definition, process assessment, and process improvement techniques have been developed in the DoD software world largely by and in response to the Software Engineering Institute's (SEI's) software process maturity framework and later Capability Maturity Model (CMM) for Software (Paulk et al. 1993). In the non-DoD software world, similar advances include total quality management (TQM) initiatives and the development of the International Standards Organization (ISO) quality standards known as ISO 9000. These advances are leading organizations toward the capability to manage reuse-driven software processes.

1.3.4 Software Product Areas Have Matured, Making Reuse Feasible for Many Domains

After several instances of a kind of product have been produced, the customers, developers, and maintainers of the systems are able to better understand the commonalities and variabilities in function and implementation. In an increasing number of cases, this has led to the insight that there have been many existing system development efforts that duplicated the development of similar functions. Whether the past duplication was necessary or not is beside the point. What is clear is that future systems in these product families should leverage the experience by engineering the common work products in such a way that a variety of future systems can be created from them.

1.3.5 Underlying Technology Has Matured to Effectively Support Reuse

Successful reuse of components other than code depends on being able to develop and communicate requirements and design abstractions. Methods and representations for requirements and design capture, for instance, are maturing and are better supported by tools. Some of the technologies for reusable software representation have already allowed for partial development of a market for reusable components (an example is EVB's GRACE parts).

Additionally, many tools for automating reuse-specific activities are under development or on the market. Reengineering tools, object-oriented development systems, application generators, and library mechanisms are examples of such tools. Underlying technologies are also under development for communication of assets between developers, brokers, and users, e.g., Reuse Library Interoperability Group (1992).

1.3.6 U.S. Government Pull for Reuse Technology Has Increased

The federal government, especially the DoD, has recognized the above trends and has initiated several efforts to develop software reuse technology for use by industry at large. Current active efforts include:

- Software Technology for Adaptable, Reliable Systems (STARS) is an ongoing Advanced Research Projects Agency (ARPA) program that includes reuse as one of its principle thrusts (STARS 1992a). Demonstration projects for the technologies are currently in progress. STARS has also developed a Reuse Strategy Model for planning reuse-based projects supporting the STARS vision of megaprogramming (STARS 1993).

- Central Archive for Reusable Defense Software (CARDS) is funded through the STARS program. Its primary thrust is the development of Command and Control System assets. Additionally, the program is developing a set of handbooks, including an Acquisition Manager's Handbook (CARDS 1992) and Franchise Plan (CARDS 1993a), which assist management in developing an implementation plan to begin the process of software reuse.

- Defense Information Systems Agency/Center for Information Management is conducting an extensive program focused on building a reuse library infrastructure and developing related technologies, training, and acquisition policy to stimulate reuse-based software development.

- Domain-Specific Software Architecture program has several parallel contracts that are constructing software architectures in which software engineers create building blocks and construction tools that can be later used by systems engineers to create subsystems (Mettala and Graham 1992).

- SEI has developed domain analysis techniques (Kang et al. 1990) and other reuse technology. Additionally, the SEI is in the process of constructing and populating a Process Asset Library that will include reuse-driven software development processes (Over 1992) captured in a form that engineers can adapt to their software development project.

- Virginia Center of Excellence for Software Reuse and Technology Transfer (VCOE) is the ARPA program that produced this guidebook. The VCOE also includes major thrusts in reuse practices, reuse economics, reuse case studies and pilot projects, and reuse education and training. Section 1.5 provides a list of related VCOE guidebooks.

In addition to these technology development efforts, the DoD has established a cross-service management structure that cooperatively addresses management and technical issues. *DoD Software Reuse: Vision and Strategy* (DoD 1992c) outlines the DoD's intended approach to putting reuse technology into practice. Each of the military services has been tasked to develop an implementation plan for the vision and strategy. Further, at least one of the larger DoD programs has already developed program-specific strategies, action plans, and executive overviews for adopting reuse technology (GPALS 1992a, 1992b, 1992c).

1.4 WHY DO YOU NEED A PROCESS FOR REUSE ADOPTION?

You need a process for reuse adoption to increase the effectiveness and the likelihood of success of your organization's effort to adopt reuse. You want to avoid the situation shown in Figure 1-3, in which the adopting organization has only a vague notion of the current situation or what is to be accomplished. Without a well-developed understanding of how reuse relates to its current practice or how reuse will fit into the overall software development effort, the adoption effort is likely to fail.

You are probably aware of efforts to incorporate reuse that have had disappointing results or of organizations that are still avoiding reuse altogether. They are typically characterized by:

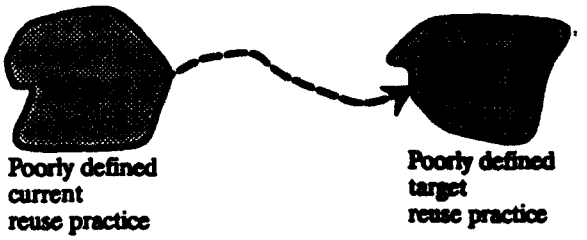


Figure 1-2. Ad Hoc Adoption of Reuse Technology

- Lack of recognition of the potential benefits of reuse
- Incomplete understanding of what is necessary to make reuse work
- Lack of management commitment and direction
- Tendency to focus on technical issues while neglecting social issues

Based on a survey of 113 individuals from 29 companies, Frakes and Fox (1993) report that management and infrastructure support are critical for systematic reuse but that most organizations are providing inadequate support. The report also indicates that the factors impeding reuse are lack of time to practice reuse, lack of trust in externally developed software, and lack of computer-aided software engineering (CASE) and development process support for reuse.

This guidebook helps you avoid the situation in Figure 1-2 and the problems described above by providing a well-defined, bounded series of steps, guidelines, and tools that you use to address the myriad issues critical to success: the Reuse Adoption process.

1.5 HOW TO USE THIS GUIDEBOOK

This guidebook consists of a Reuse Adoption process specification, supporting methods and guidelines, and examples of artifacts produced by the process. The organization of the guidebook is:

- Section 1 defines major terms and messages in this guidebook. It identifies reasons for incorporating reuse into your organization and for using this guidebook's Reuse Adoption process as an aid. Finally, it identifies some of the foundations of the process.
- Section 2, Overview of the Reuse Adoption Process, introduces the major concepts, identifies the roles in which people participate, and describes the major activities of the process.
- Section 3, Specification of the Reuse Adoption Process, identifies, in a structured format, the details of the activities and provides guidance on how to perform each of the activities.
- Section 4, Domain Assessment, describes a method for qualitatively estimating the potential for reuse in an organization's business area.
- Section 5, Reuse Capability Assessment, describes a method for understanding an organization's process with respect to reuse sufficient for planning improvements—identifying process strengths and improvement opportunities.

- Section 6, Reuse Adoption Strategy Development, provides guidance on developing a course of action to implement reuse in an organization in support of organizational objectives.
- The Appendixes provide models, guidelines, examples, and background information that support application of the Reuse Adoption process. Included are:
  - Appendix A, Domain Assessment Model, is used to determine which parts of an organization's business area are good candidates for reuse.
  - Appendix B, Reuse Capability Model, is used to assess the effectiveness of current reuse activities and to support development of goals.
  - Appendix C, Reuse Adoption Risks, identifies reuse-related situations and events that you should be aware of and manage.
  - Appendix D, Summary of Legal and Contractual Reuse Issues, summarizes the current laws and contracting practices that relate to reuse. Its primary focus is on DoD contracting.
  - Appendix E, Reuse Assessment Report Annotated Outline, provides a model report of the results of the domain and reuse capability assessments.
  - Appendix F, Reuse Action Plan Annotated Outline, provides a model plan for implementing reuse in an organization.
  - Appendix G, Assessment Worksheets, provides the forms used in conducting domain and reuse capability assessments.

Table 1-1 identifies the audience types and the sections that each should read. If you want to get an overview of the Reuse Adoption process, you should continue by reading Section 2. If you are going to execute the Reuse Adoption process, you should also read Section 3 and the supporting sections and appendixes.

Table 1-1. Guidebook Use

Audience Type	Section							
	1	2	3-6	A	B	C	D	E-G
Person implementing a reuse program	✓	✓	✓	✓	✓	✓	✓	✓
Person advocating reuse	✓	✓	3.1	✓	✓		✓	
Person controlling resources	✓	✓					✓	
Person affected by a reuse program	✓	✓						

There are several companion documents to this guidebook that will also help you embark on a reuse adoption effort.

- *Reuse Economics Spreadsheet Model User Manual* (Software Productivity Consortium 1993c) and corresponding software provides a tool for making estimates and decisions about the economic desirability of reusing software. It covers costs, productivity, return on investment, incremental funding, and quality.

- *Reuse-Driven Software Processes Guidebook* (Software Productivity Consortium 1993b) identifies reuse-driven software processes (processes that describe activities that create and/or utilize reusable assets) and methods that your organization can adopt.

- *Using New Technologies: A Technology Transfer Guidebook* (Software Productivity Consortium 1993d) provides detailed guidance on transfer of software technology into organizations.

- *Software Measurement Guidebook* (Software Productivity Consortium 1992c) identifies software development process metrics and techniques for you to use to measure the overall effectiveness of your software development efforts as well as measuring reuse in products.

- *Managing Process Improvement: A Guidebook for Implementing Change* (Software Productivity Consortium 1993a) provides detailed guidance on implementing a process improvement program to improve an organization's software process maturity.

In particular, you use the Reuse Economics Spreadsheet Model within the Reuse Adoption process; references to the spreadsheet model and the other guidebooks are made at appropriate points in this guidebook to indicate their use.

1.6 TYPOGRAPHIC CONVENTIONS

This guidebook uses the following typographic conventions:

Serif font	General presentation of information.
Italicized serif font	Mathematical expressions, publication titles, and low-level titles in the process sections of the guidebook.
Boldfaced serif font	Section headings and emphasis.
Boldfaced italicized serif font	Run-in headings in bulleted lists and low-level titles in the process sections of the guidebook.

This page intentionally left blank.

2. OVERVIEW OF THE REUSE ADOPTION PROCESS

The Reuse Adoption process is a defined set of activities to incorporate the practice of software reuse as a permanent part of an organization's culture and way of doing business, i.e., a process to institutionalize reuse. In general, it is an improvement process designed to help you improve your management and technical processes to better exploit opportunities for reuse. In addition, it is a technology transfer process specialized in transferring the reuse technologies needed to improve your process.

Reuse is institutionalized when your organization consistently identifies and exploits the reuse opportunities in its business area. You can institutionalize reuse at different levels of effectiveness and efficiency. The Reuse Adoption process helps you establish an initial level of effectiveness and efficiency appropriate to your organization's situation, then continually improves your effectiveness and efficiency. A high level of effectiveness and efficiency is characterized by an organization that can:

- Identify the high-payoff reuse opportunities in its business area.
- Establish goals for exploiting its high-payoff opportunities and consistently meet its goals.
- Apply the appropriate resources and technologies that will maximize its payoff.

To this end, the Reuse Adoption process, illustrated in Figure 2-1, will help you initiate, plan, implement, and evolve a reuse program. The process will help you:

- Establish sponsorship to get the commitment needed to start a reuse program.
- Understand the current situation with respect to reuse and establish feasible goals for improving your reuse practice and exploiting reuse opportunities.
- Develop alternative strategies for achieving your goals.
- Understand your risks, take action to reduce those risks, and select a suitable strategy.
- Turn your selected strategy into a plan of action and then implement the plan.
- Track the progress of your reuse program and continue to build on your success.

Figure 2-1 illustrates the Reuse Adoption process using a Structured Analysis and Design Technique (SADT) diagram (Marca and McGowan 1987). In this notation, an activity represents work to be performed that is directed by controls to transform inputs into outputs using indicated mechanisms.

This section provides an overview of the Reuse Adoption process. It introduces some of the key concepts, provides a top-level description, including roles and activities, and guides you on how to use the process.

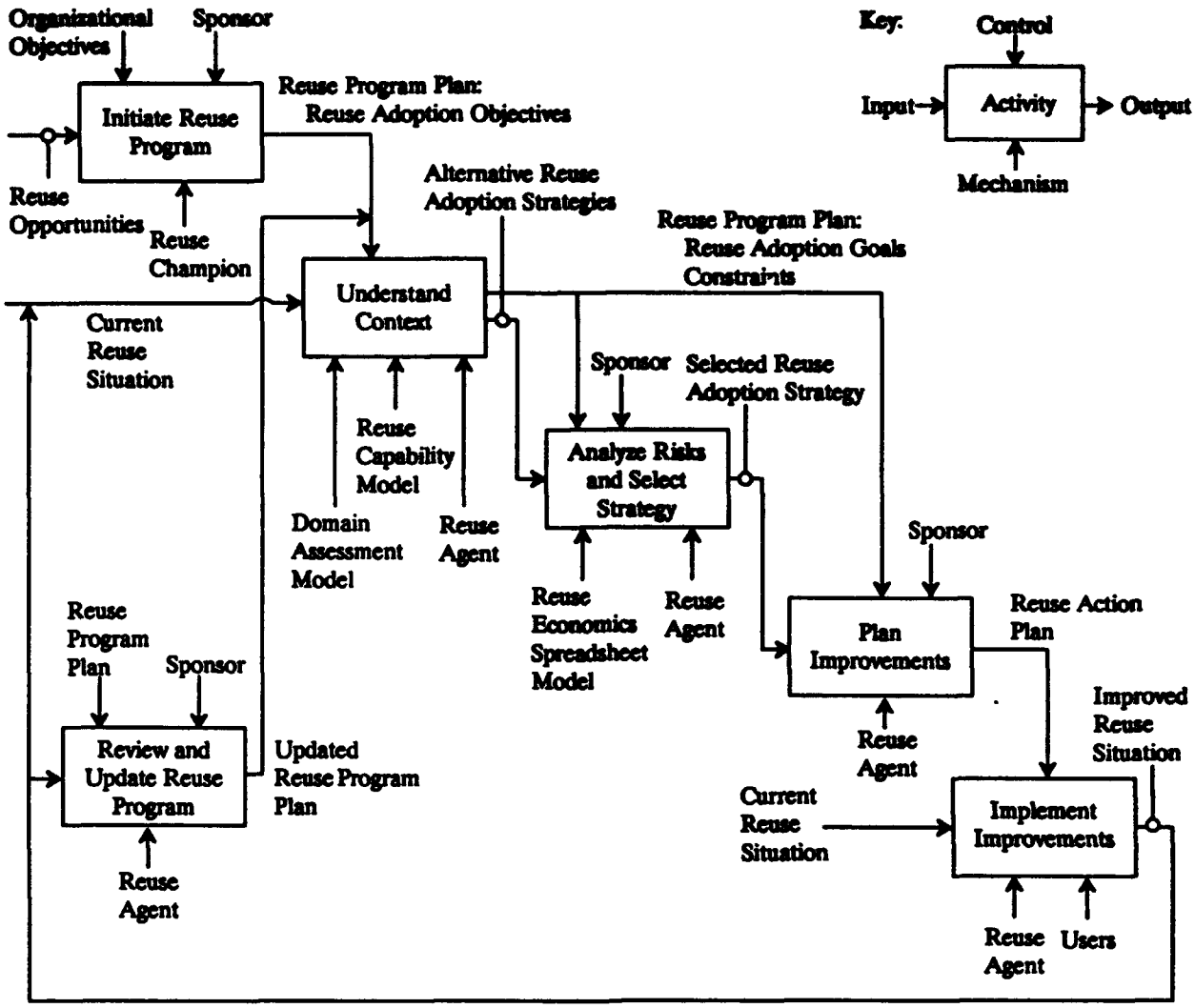


Figure 2-1. Reuse Adoption Process

2.1 REUSE ADOPTION CONCEPTS AND TERMINOLOGY

The Reuse Adoption process is based on concepts from work on reuse, process improvement, technology transfer, planning, risk management, and economics that are derived from a broad base of experience and research. This subsection introduces some key concepts and terminology of reuse adoption:

- ***One Size Does Not Fit All.*** Organizations planning to establish a reuse practice or improve an existing practice are very diverse. They vary in their objectives, organizational structure, size, customer base, type of products, financial situation, and culture. In addition, there are a variety of reuse processes, methods, and tools that an organization could use. There is also a variety of approaches that an organization can take to adopt these reuse technologies. Due to these differing situations and varying approaches, it is very unlikely that one approach to reuse could fit all organizations, or that only one approach is appropriate for a given organization.
- ***Comprehensive View of an Organization's Process.*** Although some companies are successfully practicing reuse, few companies are achieving their full potential for reuse, and many have

had disappointing results from their reuse efforts. Prieto-Diaz (1991) states that the problem is not a lack of technology for reuse; rather, the problem arises when organizations approach reuse as an independent collection of tools and techniques or when an organization focuses on the technical issues of reuse without addressing the nontechnical issues. You can alleviate this problem by taking a comprehensive view of your organization's process and addressing the technical and nontechnical issues within this context.

- ***Assessment-Based Planning.*** Institutionalizing reuse is a large, complex undertaking. Ensuring success in this undertaking requires careful planning. To develop a plan that is achievable requires an organization to have a strong understanding of its current situation. Planning through assessment is a technique for gaining this needed understanding of the situation as a basis for plan development. It has been advocated and used successfully in technology transfer, process improvement, project management, and implementing reuse programs.

  - ***Business Area Potential.*** A key area of assessment is the potential for reuse in your business area. Like any investment, you want to focus your investment in reuse where you will get the best return. This requires that you understand your reuse opportunities—what are they, their magnitude, and the factors that influence them.

  - ***Reuse Capability.*** Another key area of assessment is your ability to effectively and efficiently exploit reuse opportunities. Reuse capability refers to the range of reuse results in effectiveness and efficiency that an organization can expect to achieve through its process. The processes, methods, and tools you use to create reusable assets, utilize reusable assets, and manage your organization directly impact your actual reuse results. For this reason, you must understand your organization's current processes, methods, and tools and their effect on reuse capability (see Appendix B for the formal definition and explanation of reuse capability).

  After you understand your business area potential and reuse capability, you can plan improvements whose costs are commensurate with your potential. Although ideally it would be best to improve your reuse capability to the greatest extent possible, reality dictates otherwise. If potential is very high, then you can justify the investment necessary to improve your reuse capability to make the best of this high potential. However, if the potential is limited, then it may not be cost effective to greatly improve your reuse capability; investing to achieve a more limited improvement would be more appropriate.

- ***Reuse Economics.*** Reuse is often narrowly viewed as a technique that will reduce costs if applied, neglecting the fact that adopting and performing reuse is not free. It would be more appropriate to view the costs associated with reuse as an investment and the benefits gained from reuse as a return on investment (ROI). Understanding the costs and benefits of reuse will be an essential part of your organization's decision-making process. Your decision makers will expect you to make recommendations on how much to invest in reuse, when to invest, where to invest, and what return can be expected from an investment.

- ***Objectives, Goals, Alternative Strategies, Constraints, and Risks.*** Objectives, goals, strategies, constraints, and risks are planning constructs that have been applied to software project planning (Boehm 1988; Software Productivity Consortium 1992b). An objective is the intended or desired result of a course of action. Objectives are generally very broad statements of what

is to be accomplished. A goal, on the other hand, is a specific, time-related, measurable target. Objectives and goals give a sense of direction to an effort, and focusing on objectives and goals can help to keep the effort from straying off course.

A strategy is a plan of action to achieve a goal. Identifying and evaluating alternative strategies helps prevent individuals from pursuing personal preferences or the first solution they could identify when other potentially more effective solutions are possible.

Constraints are limitations on decisions. Identifying constraints can help prevent you from choosing a strategy that would be infeasible. A risk is a potential for incurring undesirable results. Unlike constraints, risks may be controlled. A proactive approach to identifying and averting risks can greatly increase your chance of success.

- ***Evolutionary Implementation.*** Institutionalizing reuse can be a significant change to the way an organization conducts its business. Trying to take on this amount of change at once could lead to "culture shock," where any effort to change is resisted. It also could require greater commitment from the organization and increase the risk that the reuse program will fail due to uncertainty. An evolutionary implementation, on the other hand, helps minimize the impact of change, allow an organization to incrementally commit to reuse, and reduce uncertainty by allowing the organization to increase its understanding of reuse.

2.2 THE REUSE ADOPTION PROCESS

The concepts described in Section 2.1 are an explicit part of the Reuse Adoption process. When you apply the process, you are also applying these concepts. This section provides a top-level description of the process, including its roles and activities.

2.2.1 Roles

When implementing a reuse program, there will be many people performing different functions. Because there is no fixed way in which an organization should assign individuals to these functions, the Reuse Adoption process uses a set of defined roles:

- ***Sponsor.*** An individual or group who authorizes and reinforces the reuse program. The sponsor authorizes the program, oversees the program, provides resources for implementing the program, and advocates the program.

- ***Reuse Champion.*** An individual or group who advocates and supports the reuse program. The reuse champion can be present at any and all levels of an organization, as well as outside the organization (e.g., the customer). Successful reuse champions are usually individuals who are respected for their personal or technical leadership by the sponsors and users.

- ***Reuse Agent.*** An individual or group who is empowered to plan and implement the reuse program. The reuse agent performs the definition, analysis, planning, and monitoring functions of the Reuse Adoption process and is the primary user of the adoption process. The reuse agent should be knowledgeable in the organization's business area, process, technologies, structure, and culture. The reuse agent should also have a high degree of visibility into the organization and access to key individuals and information relevant to reuse.

- *User.* An individual or group who uses the adopted reuse technologies. Individuals at all levels of an organization can be users. To obtain user buy-in to the reuse program, the sponsors and reuse agents should involve users by keeping them informed of the program's activities and by listening and responding to their concerns.

Many individuals may serve in each of these roles. Furthermore, one individual may serve in several roles. The organizational structure of the reuse program may vary from one organization to another. You may be able to take advantage of existing organizational structures, such as a Software Engineering Process Group (SEPG). This has the advantage of ensuring that the reuse program is coordinated with the organization's overall process improvement effort. Because reuse requires a change in behavior of the software developer, the reuse program will need to be fully integrated (in terms of objectives, schedule, authority, responsibility, etc.) with the software process definition activities, whether they are driven by SEI process improvement or other goals.

Przybylinski, Fowler, and Maher (1991) offer an effective organizational structure. In this structure, a senior management steering committee takes the lead and oversees the effort. Middle management serves as chairs or advisors to working groups that deal with the technical and nontechnical issues. The transition group, consisting of the change agents (reuse agents in this case), is responsible for executing the overall implementation.

2.2.2 Activities

The following subsections provide an overview of each activity. Section 3 provides detailed descriptions and guidance for each activity.

2.2.2.1 Initiate Reuse Program

The purpose of the Initiate Reuse Program activity is to obtain authorization and commitment of resources from the sponsor to begin a reuse program. It starts when a reuse champion recognizes the need or opportunity to adopt or improve the organization's reuse practice in support of organizational objectives. The output of this activity is a reuse program plan consisting of the reuse adoption objectives, schedule of adoption process activities, and resources for conducting the adoption process activities.

In this activity, you identify potential reuse opportunities and seek sponsorship to exploit these opportunities. To obtain this sponsorship, you need to identify the sponsors and convince them that exploiting these reuse opportunities will benefit the organization. You can do this by showing how a reuse program could help the organization attain its objectives, addressing the sponsors' business motivations (e.g., cost, schedule, risk, image, etc.). Before approaching your sponsors, you should also consider whether the organization is ready for a change. If the organization is already undergoing some major changes, it may not be a good time to introduce further changes.

The initial reuse program plan defines the objectives and approach you intend to take to investigate reuse. The reuse adoption objectives—what your organization intends to achieve through a reuse program—provide a focus for the program. You should be able to trace the reuse adoption objectives to the organizational objectives to ensure that fulfilling the reuse adoption objectives supports the organization's overall objectives. The Reuse Adoption process can be the basis of your approach for investigating reuse.

You complete this activity by obtaining the sponsor's approval of the reuse program plan. At this point, you are not trying to get the sponsor to commit to a course of action for implementing reuse, only to investigate reuse.

This activity may take place independently or within the context of a more general planning or improvement activity, such as strategic planning, process improvement, or TQM activities. In any case, the purpose is the same, i.e., to obtain authorization and commitment to begin a reuse program that fits your organization.

2.2.2.2 Understand Context

The purpose of the Understand Context activity is to understand the current reuse situation, establish goals for implementing reuse, and develop strategies for reaching those goals. This activity will help you take a comprehensive view of the organization's products and process to estimate the potential for reuse, identify process strengths, and prioritize opportunities for improvement. You will look at the organization's process across management levels, functional boundaries, and the external boundaries to your customers, suppliers, and competitors. Within this context, you will address the technical, managerial, economic, legal, and cultural issues that are critical to successful reuse.

In this activity, you will:

- Assess the organization's reuse potential. This task includes an assessment of your organization's business area potential and its reuse capability. To help you assess your business area potential, the Reuse Adoption process provides a Domain Assessment Model (Appendix A). This model will help you look at the factors influencing your business area potential, such as the quality and availability of existing assets, level of standardization, stability, commonalities and variations, and market needs.

  To help you gain an understanding of your organization's reuse capability, the Reuse Adoption process includes the Reuse Capability Model (RCM). The RCM captures the issues that are critical to improving your organization's reuse capability (e.g., organizational commitment, asset awareness, asset quality, and training). You apply the RCM within the Reuse Adoption process to identify your organization's strengths and opportunities for improvement and to establish reuse adoption goals (where you want to be with respect to your reuse capability). Appendix B provides a detailed discussion on reuse capability and the RCM.

- Establish reuse adoption goals—measurable results toward which the reuse program is directed to achieve the reuse adoption objectives—taking into account your reuse potential. The goals state what must be accomplished and when for you to consider the reuse adoption objectives to be satisfied. You use the understanding gained in assessing your reuse potential to ensure that the reuse adoption goals are achievable. You may establish both near- and long-term goals to define the evolution of your program.

- Identify any constraints on your reuse program—limits on the accomplishment of your reuse adoption goals or choice of strategies. After you assess your reuse potential and identify constraints, you may determine that the reuse adoption objectives or goals are not attainable, so you may have to adjust them or delay or abandon the program. Include the reuse adoption goals and constraints in the reuse program plan.

- Identify alternative reuse adoption strategies—alternative plans of action to achieve the reuse adoption goals and objectives. A complete reuse adoption strategy should include:
  - ***Product Approach.*** What end products will be affected by the reuse effort?
  - ***Business Model.*** Who pays to build reusable assets? Who builds them? Who reuses them? What rights do the customer, developer, and reuser have to the assets? Who pays whom for assets that are reused? Who assumes liability for reused assets?
  - ***Process Approach.*** What processes, methods, and standards will be used to exploit reuse in developing or maintaining software products, to create reusable assets, and to manage product and asset development? What types of software assets will be produced and reused?
  - ***Environment Approach.*** What tools, automated and nonautomated, will be used to support the development process?
  - ***Organizational Approach.*** What organizations will be affected by the reuse program? What organizational structure will be used? Who will manage it?
  - ***Transition Approach.*** How will changes to the organization, its policies, procedures, processes, methods, standards, and tools be accomplished? What is the time frame for the changes?

  You can identify alternative reuse adoption strategies through a variety of means, such as brainstorming, reuse process models, your organization's past experiences with reuse, other organizations' experiences with reuse, and reuse conferences and literature.

  At this point, your alternative reuse adoption strategies may not be completely defined. Also, do not eliminate any strategies unless they are clearly infeasible. Your task is to get as many ideas on the table as possible, further refinement and evaluation will come later.

After you complete the definition of goals, constraints, and initial alternative strategies, you should have the sponsor review and approve the updated reuse program plan to verify that you are heading in the right direction.

2.2.2.3 Analyze Risks and Select Strategy

The purpose of the Analyze Risks and Select Strategy activity is to refine, evaluate, and select a reuse adoption strategy for implementing reuse. In this activity, you can refine the alternative reuse adoption strategies identified in the Understand Context activity and may identify additional alternatives. Your task is to narrow the list of alternatives to a few strategies that best satisfy your objectives and goals for recommendation to the sponsor for a decision.

You can work toward a recommended strategy by iteratively analyzing, refining, and possibly eliminating strategies until you have sufficient confidence in a few strategies to make a recommendation. In analyzing the strategies:

- Identify and assess risks associated with the strategies. What can possibly go wrong, how likely is it to happen, and what would be the consequence?

- Analyze the strategies' impacts on the organization. Who will be affected by the strategies, how will people react, and what resistance can you expect?

- Analyze the economics of the strategies. How much do you need to invest, what is the expected return on investment, when will you break even on the investment, what would be the impact of the strategy on productivity and quality?

- Determine the extent to which the strategies satisfy your objectives, goals, and constraints.

There are a variety of analytical techniques that you can use to carry out an analysis. This guidebook provides a risk checklist (Appendix C) to help you identify risks. To help you understand the costs and benefits of reuse, the Reuse Adoption process includes the Consortium's Reuse Economics Spreadsheet Model (Software Productivity Consortium 1993c). The spreadsheet model provides a tool for making estimates and decisions about the economic desirability of reusing software. It covers costs, productivity, return on investment, incremental funding, and quality.

Other possible techniques include trade studies, utility analysis, red teams, prototyping, simulation, role playing, and more. You should strive to include the users in your analysis as much as possible to get their input and obtain their buy-in to the recommended strategies.

After you complete the analysis, present a recommendation, rationale, and supporting analysis material to the sponsor for a decision.

2.2.2.4 Plan Improvements

The purpose of the Plan Improvements activity is to develop a plan for implementing reuse based on the selected reuse adoption strategy. During this activity, you identify actions and schedule resources for putting the selected reuse adoption strategy into effect. The reuse action plan is a transition plan. You include actions for putting the product, business model, organization, process, and environment approaches of the adoption strategy in place, e.g., training people in reuse techniques, documenting policies and procedures, and acquiring tools. Hence, you derive many of the actions for the reuse action plan from the transition approach of the reuse adoption strategy.

In this activity, you will:

- ***Identify Actions.*** You should identify actions required to implement the reuse adoption strategy, including the:
  - Purpose and description of work to be performed
  - Inputs and expected results or artifacts
  - Resources and personnel required
  - Estimated cost and duration
  - Dependencies on other actions
- ***Schedule the Actions.*** The schedule should satisfy any action dependencies and resource constraints. You should also include periodic reviews and milestones for monitoring progress.

- ***Identify Data Collection Requirements.*** You should collect data to facilitate monitoring the effort against the plan (e.g., estimated, actual start, and completion dates), as well as data to aid in determining whether the reuse adoption objectives and goals are met.

After you develop the reuse action plan, you should have it reviewed by all of the organizations affected by the plan, then seek commitment from the sponsor to enact the plan. Appendix F provides an annotated reuse action plan to serve as a model for your plan.

2.2.2.5 Implement Improvements

The purpose of the Implement Improvements activity is to enact the reuse action plan. In this activity, you will obtain the necessary resources and perform the actions as identified in the reuse action plan. You will assign specific individuals to execute and manage the tasks, ensure that the individuals have the resources they need, and authorize work on the tasks to begin. You will also collect data and monitor the progress of the implementation against the reuse action plan. As with any plan, make adjustments to the plan based on your progress, results, and any problems or opportunities that may arise.

2.2.2.6 Review and Update Reuse Program

The purpose of the Review and Update Reuse Program activity is to determine whether the reuse program has met its objectives and goals, to what extent the program has met the objectives and goals, and whether the program should address additional objectives or goals. At this point, you may choose to close the reuse adoption effort (e.g., because the program met its objectives and successfully institutionalized reuse). Or, you can cycle back to the Understand Context activity and continue the process (e.g., because the program has not met its objectives, making it necessary to replan, or because you wish to further improve the organization's reuse capability).

The Reuse Adoption process supports evolutionary implementation but does not require it; remember, one size does not fit all. You can use it to identify stages of implementation that allow you to reduce startup costs and build up your capability over time while providing for some early return on investment.

2.3 USING THE REUSE ADOPTION PROCESS

When using or deciding whether to use the Reuse Adoption process, keep the following thoughts in mind:

- Adopting reuse may be less linear and orderly than the process illustrated in Figure 2-1. The dynamics of your situation will influence the application of the process. However, the process is flexible. You may need to perform some of the activities in parallel, skip activities and repeat, or iterate the activities.

- You can exit the process at any point if you decide either it is not appropriate to continue or there is no need to continue.

- The resources and time required to perform the process will vary with the scope of the effort. The greater the scope and impact on the organization, the more resources will be required.

Completion of the first four activities could take anywhere from a few weeks to a year depending on the scope of your program. The Implement Improvements activity will likely consist of multiple, 1-year stages to coincide with your annual planning and budgeting.

- Your organization may already have a process improvement, technology transfer, or planning process in place that could be applied to institutionalizing reuse. In this case, you can use concepts or methods from the Reuse Adoption process to enhance your process. Another possibility is that you do not have a process in place but do have a method that can be used to complement or replace a method included in the Reuse Adoption process. For example, you may have a domain assessment method that you already understand and can use in place of the domain assessment method provided in this guidebook.

- Adopting reuse involves many people and organizations with differing needs, concerns, and viewpoints. The Reuse Adoption process is not a mechanism for controlling the behavior of your organization; rather, it is a guideline. It can help you comprehend the myriad issues involved in institutionalizing reuse and take steps that will lead to success. The process can augment your good judgment, but it is not a substitute.

Adopting reuse is a complex and challenging task. The Reuse Adoption process can guide you in developing and implementing a plan to institutionalize reuse, a plan that is based on a strong understanding of your situation, reduces the risk of failure, and increases the benefits you can expect from reuse.

3. SPECIFICATION OF THE REUSE ADOPTION PROCESS

This section provides the details of the activities that make up the Reuse Adoption process. Each subsection describes one of the top-level activities shown in Figure 2-1:

- Initiate Reuse Program (Section 3.1)
- Understand Context (Section 3.2)
- Analyze Risks and Select Strategy (Section 3.3)
- Plan Improvements (Section 3.4)
- Implement Improvements (Section 3.5)
- Review and Update Reuse Program (Section 3.6)

Each activity subsection begins with a specification of the activity. The specifications are:

- ***Purpose.*** The reason for performing the activity.
- ***Entrance Criteria.*** Conditions that must be met before an activity can be started.
- ***Tasks.*** Work to be performed in the activity.
- ***Inputs.*** Information used and transformed in the activity.
- ***Controls.*** Entities that constrain or enable the activity or information used to direct performance of the activity.
- ***Outputs.*** Information produced in the activity.
- ***Mechanisms.*** Roles responsible for performing the activity and methods that can be used to perform the activity.
- ***Exit Criteria.*** Conditions that must be met before an activity can be considered completed.

The activity specification also includes subsections for each task of the activity. The task subsections identify the purpose of the task and provide guidance on the task. Steps you may take in performing the tasks are provided within the task guidance; they are identified in the left margin within the guidance section for quick reference. The header of each page includes an icon of the Reuse Adoption process diagram (Figure 2-1). The current activity is shaded to serve as a reminder of the context.

This page intentionally left blank.

3.1 INITIATE REUSE PROGRAM

Purpose Obtain authorization and commitment of resources from a sponsor to begin a reuse program.

Entrance Criterion Reuse champion recognizes a reuse opportunity.

Tasks

- Elaborate reuse opportunities.
- Identify relevant organizational objectives.
- Analyze organizational readiness.
- Develop reuse program plan.
- Establish sponsorship.

Inputs Information that characterizes reuse opportunities (business plans, product plans, market forecasts, improvement plans, technology reports, etc.)

Controls

- Information that characterizes organizational objectives (mission statement, business plans, improvement plans, etc.)
- Sponsor

Output Reuse program plan:

- Reuse adoption objectives
- Schedule of adoption process activities
- Resources

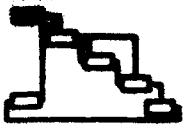
Mechanism Reuse champion

Exit Criterion The sponsor has approved the reuse program plan.

Your first job in the Reuse Adoption process is to get a sponsor's commitment to initiate a reuse program. You can do this by elaborating any reuse opportunities and persuading the sponsor that exploiting these opportunities will help the organization meet its business objectives. Before approaching the sponsor, you should also take into account the organization's readiness; i.e., is this a good time to implement organizational changes for reuse?

To improve your chances of winning sponsorship, you need to provide the sponsor something tangible to which to commit. You also need to give the sponsor some assurance that the effort will be successful. The reuse program plan fills both of these needs. In the reuse program plan you identify what the organization expects to accomplish with the reuse program, how the reuse program will be conducted, and who will participate.

At the completion of this activity, you review the rationale and prospects for reuse in the organization with the sponsor and seek the sponsor's commitment of resources to execute the reuse program plan.



At this point, you are not trying to get the sponsor to commit to an approach for reuse—only the approach for investigating reuse. You should point out to the sponsor that, at the completion of the Analyze Risks and Select Strategy activity, you will seek commitment to a specific approach to reuse.

3.1.1 Elaborate Reuse Opportunities

Purpose

Elaborate the areas where the organization has the opportunity to apply reuse technology.

Guidance

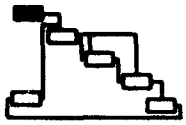
In this task, you elaborate the reuse opportunity that prompted you to take action and look for additional reuse opportunities to help strengthen your case. A reuse opportunity is an occasion when an asset (existing or to be developed) may be used to satisfy a need (current or anticipated). For example, you might anticipate the need for a new line of personal productivity software using hand-held technology that could be based on your existing line of office automation software. To elaborate an opportunity, you should identify both the asset source and the target need.

Recognizing opportunities for applying reuse technology requires an understanding of the organization's current situation and also the enabling prerequisites for successful application of reuse technology. You may find information characterizing reuse opportunities in the organization's business plans, product plans, technology reports, etc. (e.g., similarities between two planned product developments may represent a reuse opportunity). Some of this information may not be documented, so you may have to seek this information from personnel with the requisite knowledge.

At this point, you do not want to do an exhaustive analysis of your reuse opportunities (that will happen later). You do want to identify sufficient opportunities to persuade the sponsor to investigate reuse.

Below is a list of situations that present opportunities. Use this list to identify conditions in your organization in which reuse is appropriate. The first two items are prerequisites for reuse opportunities. The other items are motivating factors that make reuse technology more attractive. You may extend this list based on your own situation.

- ***Defined Product or Component Line.*** Development of reusable assets must be focused on the current and perceived future needs of the end systems that will use them.
- ***Existing Experts.*** You have or can get a staff that understands requirements, designs, implementations, and development processes for the product line.
- ***Existing Software Assets.*** You own or can procure rights to software assets that partially meet existing requirements.
- ***A Long-Term Evolutionary Product.*** A current product is expected to undergo major upgrades and changes over its life cycle.



- ***Customer Demand for Many Versions of Products.*** Similar needs across the customer base cannot be satisfied by a single product but require modifications, extensions, or upgrades.
- ***Customer Need for Prototypes of Product or Component.*** You have a product or component line that is expanding as new, useful applications are identified. However, you need working prototypes to support development of markets.
- ***Reduced Staff Available to Cover Potential Demand.*** The organization is going to downsize and must leverage the knowledge of fewer experts. Or, more positively, increased product demand will outstrip the staff's capacity to deliver new products.
- ***Move to New Product Line.*** The organization plans to adapt existing capabilities and products as a start for entering a new business area.
- ***Customer Demand for Reusable Assets.*** A potential customer wants software that can be upgraded, managed, and applied to new problems.
- ***Parallel Developments Have Similar Components.*** Two or more products at roughly the same development stage have similar functions.
- ***Reduced Time Between Generations.*** The time between technology advances inhibits the development of all-new products, forcing reuse.

3.1.2 Identify Relevant Organizational Objectives

Purpose

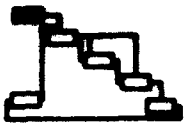
Identify the organizational objectives that can be satisfied by the adoption of reuse technology. These should be specific, recognized objectives.

Guidance

In this task, you start by identifying the broad objectives of the organization as a whole. You may find information characterizing these objectives in the organization's mission statement, business plans, etc. Again, some of this information may not be documented, so you may have to seek this information from personnel with the requisite knowledge. From these objectives, you identify which of the objectives might be supported by reuse technology.

When identifying relevant organizational objectives, you should start from the recognition that organizations must constantly hone their abilities to achieve a small number of objectives that are absolutely essential to their survival:

- A competitive edge on proposals for projects that satisfy organization mission and objectives
- An ability to market, develop, produce, and deliver required performance, on time, within planned cost and quality
- An ability to respond quickly to changing technology and customer needs
- An ability to profit from delivering products and services



Ultimately, you must be able to relate the reuse program to these high-level objectives. Table 3-1 has a more specific set of objectives and a top-level indication of their relationship to reuse technology.

When identifying relevant organizational objectives, you also need to identify the organizational scope, the sponsors, and the sponsors' business motivations.

Table 3-1. Reuse Technology Support for Common Organizational Objectives

Organizational Objectives	How Reuse Technology Supports
• Interoperability • Standardization • Commonality	Objectives imply agreement on commonality of functions and their interfaces. Reusable assets that implement the functions and their interfaces help meet these objectives in a top-down manner by lowering the cost of adhering to the agreements. Additionally, existing, reused assets help meet the objectives in a bottom-up manner by becoming de facto standards that have been proven through field implementation.
• Core product development • Design for growth • Design for maintenance • Modularity	Objectives imply analysis of future needs for developed assets during the creation of a current product. Reuse (domain analysis) technology focuses on this objective, provides methods for meeting it, and places it in a development life-cycle context.
• Use of COTS products • Reengineering of existing products	Objectives imply using existing assets with or without modification. Reuse (domain analysis) technology facilitates early identification of the tradeoffs involved.
• Customer requirements identification • Risk reduction	Objectives imply a need to understand implementation-related knowledge during the early stages of development of new products. Reuse technology leads to the availability of assets that can support rapid development of prototype implementations.
• Retention and leverage of technical expertise	Objective implies capturing technical expertise so that it can be more effectively applied to creating new products. Reuse (domain analysis) technology identifies commonalities and variations in expert solutions to similar problems. Domain implementation technology supports the creation of assets that can be used by others (with less expertise) to solve new problems.
• Quality improvement/ defect reduction	Objective can be met through reuse technology when new needs are met by existing, tested products. Test resources for the new product can be applied to further improvement of the asset quality.
• Productivity improvement • Time to market	Objectives imply reduced labor hours by avoiding work, working faster, and avoidance of rework in the test phase or in subsequent development efforts. Reuse technology supports this objective by allowing development at higher levels of abstraction, i.e., on an asset-by-asset rather than instruction-by-instruction basis.
• Advancing SEI process maturity	Objective requires that you define a reuse program that meets requirements specified in the CMM.

Identify the Organizational Scope

You should identify the management level of the organization for which the reuse program would be implemented.

You should scope the reuse program to the organizational level that is capable of exploiting the identified reuse opportunities. Consider the sample organization in Figure 3-1 and assume that Program A is in early development, Program B is in initial planning, and that the Program B manager recognizes that there is an opportunity to reuse software currently in development from Program A in Program B. To effectively exploit this reuse opportunity requires close coordination between Program A and Program B; thus, the reuse program should be mapped to the Engineering organization.

Now assume further that there is an opportunity to develop a product line based on Program A's software. In this case, the reuse program should be scoped to the Division because it involves the Division's product strategy.

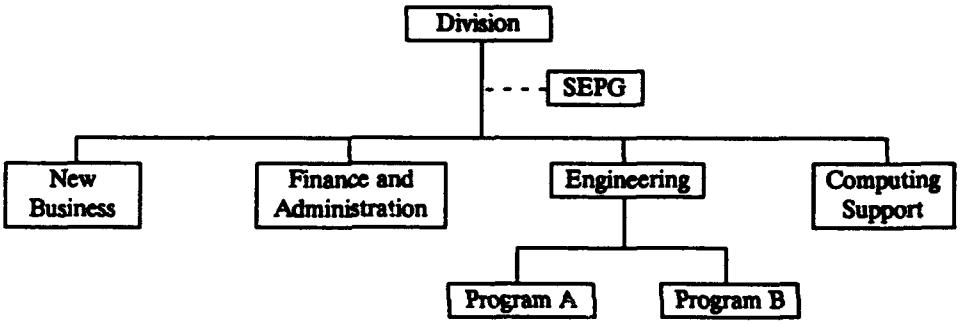


Figure 3-1. Choosing the Organizational Scope

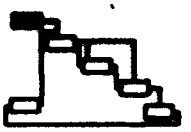
In choosing the scope of the organization, you will need to trade off the benefits of increasing the scope (e.g., for large organizations, there is more opportunity to spread costs by having more applications of the developed assets) versus the drawbacks and risks (e.g., reuse in a large organization may require commitment that is beyond the scope of your influence).

Identify Potential Sponsors

Closely tied to the organizational scope, you should also identify the individuals, offices, or committees that could supply resources to fund and staff the reuse program. There are several different types of sponsors that you may be able to enlist to support the initiation of a reuse program. They are:

- ***Product line managers*** that plan to seek contracts for multiple, similar end products and are motivated to develop core product assets that will provide them with a competitive edge.

- ***Proposal managers*** that are looking for discriminators for a specific contract win. For large programs, the investment in reuse technology might have a payoff on a single contract because it can reduce the costs of developing similar versions of the software; e.g., many pieces of support equipment on the contract may reuse software assets for software development test, hardware qualification, and system acceptance.



- ***Managers of ongoing, long-term programs*** that have requirements to create reusable software assets that can be reused in evolution of the system.
- ***Technology organization managers*** that are responsible for functions or components that are a part of multiple product lines. Here, reuse technology would be applied to creating assets that can be effectively tailored to specific end products.
- ***Process improvement program managers*** that are tasked with formulating changes to business practices that will allow the organization to better meet its objectives. An example is SEPGs attempting to determine how best to perform reuse.
- ***Customer organization managers*** that are procuring and will become owners of the products you deliver. These managers will then have analogous responsibilities and thus similar motivations to the contracting organization managers.

Identify Business Motivations

Clarify the reason the organization should initiate reuse. There are two classes of possible motivation:

- The organization's motivation for adopting reuse is to improve its reuse capability to support general organizational objectives, such as faster time-to-market, improved productivity, or improved quality.
- The organization's sponsor wishes to implement a reuse program to exploit a specific reuse opportunity or has been directed to attempt some aspect of reuse. For example, the sponsor wants to improve a bid on the development of a new system by reusing software developed on another system, or the customer has requested (e.g., via a Statement of Work) use or creation of reuse-related products or processes.

You may find that the sponsor requires estimates of expected payoff. Credible estimates are difficult to generate in the scope of this activity and are more appropriate to generate during the Analyze Risks and Select Strategy activity. However, you can use the Reuse Economics Spreadsheet Model to provide some support for this need (Software Productivity Consortium 1993c).

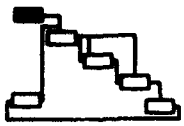
3.1.3 Analyze Organizational Readiness

Purpose

Determine the organization's capacity or readiness for starting a reuse program.

Guidance

You should understand other initiatives or forced changes that the organization is currently undergoing or planning. Because the implementation of reuse technology affects many parts of the organization, the program you identify probably interacts with the other initiatives. Sponsors need to know the relationship. Examples of related activities include strategic planning, process improvement, or TQM initiatives.



Software Productivity Consortium (1993d) suggests you do the following when analyzing the organization's readiness for change:

- Examine prior technology change efforts in the organization—successful change experiences increase the likelihood of acceptance; unsuccessful experiences greatly decrease the chances of success.
- Understand other changes that either have just occurred or might be occurring simultaneously with the reuse program. If possible, try to stabilize other factors while implementing the reuse program.
- Understand the risk associated with implementing the reuse program now versus waiting until later. Will you lose an advantage by not acting now, or can you wait until conditions are better?
- Look at the expectations and culture of the areas of the organization to be affected by the reuse program. Do they perceive a need for the change or is the change being forced on them?

3.1.4 Develop Reuse Program Plan

Purpose

Clearly identify what is to be accomplished with the reuse program, how the program will be conducted, and who will participate.

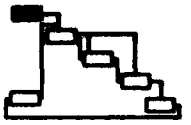
Guidance

The reuse program plan specifies your intentions for a reuse program. Initially, you use the plan as a proposal for seeking sponsorship. It then becomes the guiding document for the reuse program. The reuse program plan is further updated in the Understand Context and the Review and Update Reuse Program activities. The reuse program plan consists of:

- Reuse adoption objectives
- Approach for investigating reuse
- The identification of staff resources (reuse agents, coordinating organizations, and funding) needed to begin the program
- Constraints imposed by the sponsor or others on the solution or the program
- Reuse adoption goals and constraints on the goals, defined in the Understand Context activity

Define Reuse Adoption Objectives

Your reuse adoption objectives should answer the questions of what you want to get out of reuse and why a reuse program is being pursued. The reuse adoption objectives are used to evaluate any alternative reuse adoption strategies that are proposed. You should derive reuse adoption objectives from the organizational objectives that were identified as drivers for the reuse program.



Often, reuse adoption strategies are confused with reuse adoption objectives. Your task is to clearly distinguish the objectives from the strategies. For instance, a sponsor might state that the reengineering of an existing system to meet a new need is the objective. In this case, you should identify reengineering as a potential strategy rather than as the objective. The objective behind the reengineering strategy may be to develop a competitive bid for the new system. If so, this should be the stated objective. Then, in identifying strategies, you have the latitude to seek alternative reuse adoption strategies that may better meet the objective.

Specify Reuse Program Approach

The Reuse Adoption process is an approach for conducting a reuse program upon which you can base your program. In this step, you tailor the Reuse Adoption process to your organization. For example, if you already have a process improvement or technology transfer process in place, then you can adapt the Reuse Adoption process to fit the existing process. You should tailor the Reuse Adoption process after familiarizing yourself with the rest of this guidebook.

You should also indicate the decision points in your reuse program approach. The decision points in the Reuse Adoption process are at the completion of the Initiate Reuse Program, Analyze Risks and Select Strategy, Plan Improvements, and Review and Update Reuse Program activities. You can use the process diagram in Figure 2-1 to explain each activity, the information provided to the sponsor, and the decisions to be recommended.

Identify Resources

Indicate specific organizations involved in the reuse program so that you have backing to seek their review and concurrence during the program effort. These organizations include all those that would be affected by the reuse program. Examples are potential asset developers, maintainers, users, the new-business office, and the contracting agents.

When organizing a group to conduct a reuse program, you should try to achieve a good cross-representation of the organizations that will likely be affected and include the organization's opinion leaders; this will help facilitate buy-in to the group's recommendations.

Figure 3-2 illustrates an example of organizational infrastructure for a reuse program in which senior management participates in the steering committee to oversee the reuse program and to review or approve recommendations. An SEPG can serve as this steering committee. Middle managers chair the working groups staffed by representatives from the affected organizations. The working groups develop the reuse adoption strategies and reuse action plan. As shown in Figure 3-2, you may choose to organize multiple working groups to address specific areas. In this case, there must be close coordination between the separate groups.

You will likely be constrained on the number of staff resources that can be dedicated to this effort. To satisfy this constraint and still have good cross-representation, you can have other individuals provide input or participate as reviewers. The actual resources required for implementing the Reuse Adoption

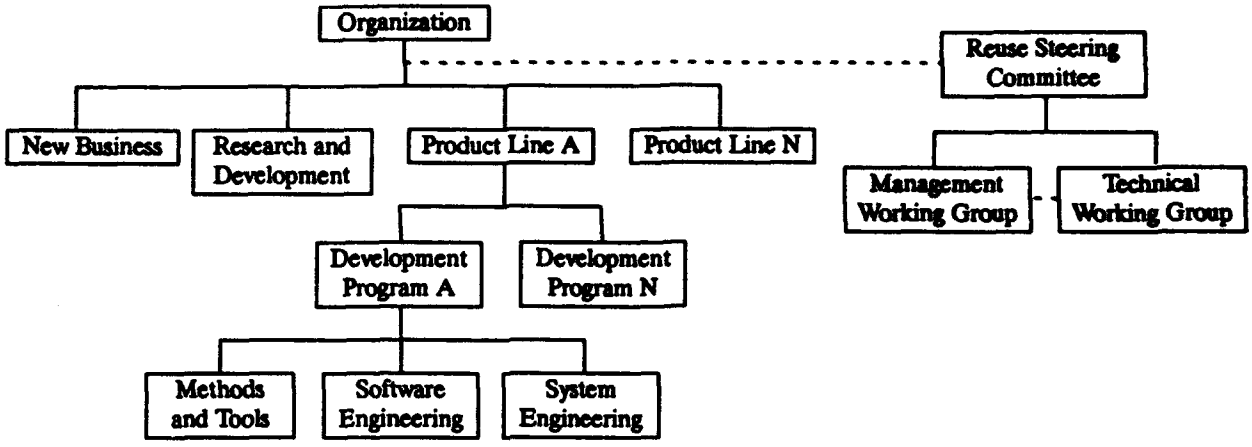


Figure 3-2. Reuse Program Infrastructure

process varies depending on the scope of the reuse program and the complexity of your current situation.

Identify Constraints

You should document any constraints imposed by the sponsor or others on the conduct of the Reuse Adoption process or end result of the Reuse Adoption process, such as a particular business strategy.

3.1.5 Establish Sponsorship

Purpose

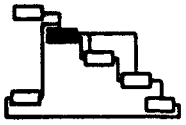
Get a clear, common agreement on the reuse program plan.

Guidance

You will probably confer with the potential sponsors several times before the earlier steps are viewed as completed and the resources are authorized. The material developed in the tasks above will be the basis for obtaining commitment.

Finally, you should make it clear to the sponsor that one possible valid result of this process is that you decide not to implement reuse because conditions in the organization, in the product line, or with the customers may not support reuse. The Understand Context and Analyze Risks and Select Strategy activities that follow will help you determine if, in your situation, the implementation of reuse is likely to be worth the costs.

This page intentionally left blank.



3.2 UNDERSTAND CONTEXT

Purpose	Understand the current reuse situation, establish goals for implementing reuse, and develop strategies for reaching those goals.
Entrance Criterion	The sponsor has approved the reuse program plan.
Tasks	• Assess reuse potential. • Establish reuse adoption goals. • Identify constraints. • Identify alternative reuse adoption strategies.
Input	Current reuse situation, characterized by the organization's mission, structure, personnel, products, customer and supplier relationships, processes, standards, tools, market, and technology base
Control	Reuse program plan (reuse adoption objectives)
Outputs	• Reuse program plan: – Reuse adoption goals – Constraints • Alternative reuse adoption strategies
Mechanisms	• Domain Assessment Model • RCM • Reuse agent
Exit Criteria	• The reuse program plan has been updated to include the reuse adoption goals and constraints and has been reviewed and approved by the sponsor. • Alternative reuse adoption strategies have been identified.

Before you can develop and implement a plan to adopt reuse, you must have a good understanding of the current situation (where you are), then establish realistic goals (where you want to be), and finally produce strategies (how you intend to get there).

In this activity, you assess the organization's reuse potential. This includes an assessment of your organization's business area potential and its reuse capability. Based on the assessment results and the reuse adoption objectives, you then establish reuse adoption goals. The goals state what must be accomplished and by when for you to consider the reuse adoption objectives satisfied. Then you identify any constraints on the reuse program that limit the choice of strategies. You are then in a position to identify alternative reuse adoption strategies to achieve the reuse adoption goals and objectives.

3.2.1 Assess Reuse Potential

Purpose	Determine the potential for reuse in your organization's software business area and its current reuse capability. There are three steps:

- Develop organizational profile.
- Assess the business area potential.
- Assess the organization's reuse capability.

Develop Organizational Profile

Before assessing your organization's business area potential and reuse capability, it is helpful to establish the context for the assessments. The organizational profile is a rough sketch of your organization that establishes the context for the assessment and an initial baseline description of the organization. You do not need to go into great detail; this is only a high-level description. The organizational profile includes:

- ***Organization.*** Mission statement, organization chart, number and type of personnel, and list of subcontractors and suppliers.
- ***Products.*** Identity and description of products and their customers, development phase, size, and development languages.
- ***Business Model.*** Identity of business entities involved in an acquisition process (e.g., developers, customers, subcontractors, etc.) and the contractual or financial relationships among them.
- ***Process.*** Description of major processes (management, development, maintenance, etc.), process diagram, organization process maturity, life-cycle model, and list of development standards.
- ***Environment.*** Identity of methods and tools used to support the identified processes.

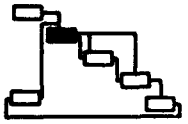
Note that these profile elements are the same as the components of a reuse adoption strategy minus the transition approach. These represent a first-cut description of your organization; the assessments then provide you with a deeper understanding of the organization in these areas. You can also specify the reuse adoption goals in these areas; then the strategies specify how these areas will change.

Assess the Business Area Potential

This step is a precursor to the domain analysis activity that you will perform when you begin to execute a reuse-driven software development process. In this step, you perform a first-cut qualification of the software business areas by examining whether necessary technical, business structure, and organizational conditions exist. In addition to getting a rough understanding of where the best reuse opportunities lie, you use this information later to support economic analysis and the initial domain analysis activities.

You accomplish this step using the domain assessment method described in Section 4 and the supporting Domain Assessment Model in Appendix A. Section 4 provides a qualitative approach for assessing your business area. There is no quick, quantitative method for determining the viability of reuse

- Understand the risk associated with implementing the reuse program now versus waiting until later. Will you lose an advantage by not acting now, or can you wait until conditions are better?
- Look at the expectations and culture of the areas of the organization to be affected by the reuse program. Do they perceive a need for the change or is the change being forced on them?
- Do the projects targeted for adopting reuse have the extra resources



investment in a given software business area. If you have good historical data, then quantitative analysis can be performed later during the Analyze Risks and Select Strategy activity, but it is not practical at this point in the process.

Assess the Organization's Reuse Capability

The purpose of this step is to examine the strengths and opportunities of how your organization currently reuses its software assets. With this knowledge and the knowledge you gain in assessing the business area potential, you will be able to set realistic goals for the reuse program.

To perform this step, use the reuse capability assessment described in Section 5 and the assessment component of the RCM provided in Appendix B. The RCM identifies factors and goals related to an organization's capability to reuse software assets. The factors are subdivided into four broad groups that address the topics:

- How well does the organization utilize reusable assets in the development of end products?
- How well does the organization acquire or develop assets for reuse?
- How well does management facilitate reuse?
- How well does the organization address reuse process issues?

For each of these topics, there is a set of critical success factors, those factors that are most critical to improving reuse capability. For each of the critical success factors, there is one or more critical success factor goals that provide insight into your organization's capabilities.

Although you use the assessment model component of the RCM described in Appendix B to provide an internal assessment of your organization, it does not result in the assignment of an organizational "reuse maturity level." Because of this, the RCM should not be used to compare organizations against each other. It should only be used by an organization to assess and plan for its own reuse process improvements.

3.2.2 Establish Reuse Adoption Goals

Purpose

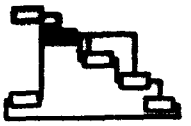
Specify measurable, achievable goals for the reuse adoption objectives.

A reuse adoption goal consists of:

- A description of the modified critical success factor goals to be achieved
- The organizational and domain scope for the reuse effort
- The time frame in which the capability is to be achieved

Guidance

You should choose reuse adoption goals so that accomplishment of the goals implies successful accomplishment of the reuse adoption objectives. In



addition, you use the reuse capability and domain assessment results to ensure that the goals are realistic.

In the Assess Reuse Potential task, you identified how well your organization met the set of critical success factor goals identified in the RCM. These critical success factor goals are organized into stages in the RCM's Implementation Model, and use of the stages is a recommended starting point for identifying the reuse adoption goals.

The stages of the Implementation Model do not represent a mandatory set of transition criteria; i.e., they do not imply that you must first satisfy every goal of the opportunistic stage before advancing to the integrated stage. They only represent a prioritization of the critical success factor goals. The prioritization is based on risk; i.e., to reduce your risk, you should assign a higher priority to the goals of the earlier stages (opportunistic, then integrated, etc.). Also, the stages represent a recommended starting point that must be tempered by organizational objectives, domain suitability, and the near-term needs of your software development programs.

If you have identified several potential domains that are significantly different in their characteristics (e.g., market potential, stability), you should consider developing a separate set of reuse adoption goals for each of the domains. You can use the steps that follow to develop reuse adoption goals for your organization.

Select an Implementation Stage as an Initial Target

Using the findings of the reuse capability assessment, identify where your organization stands with respect to the implementation stages by constructing a table similar to Table 3-2, which is a sample summary of how well an organization met the critical success factor goals. This table allows a visual representation of where your organization's strengths and opportunities are with respect to the implementation stages.

If the findings of the domain assessment indicate that you have a moderate-to-high potential for reuse, then you could select the leveraged or anticipating stage as the target. However, if you have not satisfied a majority of the critical success factor goals in the opportunistic and integrated stages, then you should select the integrated stage as a near-term target with the leveraged or anticipating stage as a long-term target.

If the findings of the domain assessment indicate that you have a low-to-moderate potential for reuse, then you should select the opportunistic or integrated stage as the target.

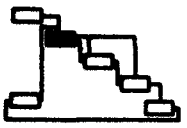


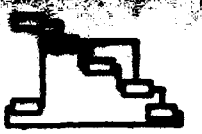
Table 3-2. Using the Reuse Capability Implementation Model to Support Reuse Adoption Goal Setting

Critical Success Factor		Opportunistic	Integrated	Leveraged	Anticipating
Application Development	AA	1■	2□		
	AI	1◪	2◪	3□	4 ⊟
	AE	1◪ 2◪			
	AN		1◪		
Asset Development	NI	1◪ 5◪	2◪	3 ⊟	4 ⊟
	AD	1◪	2◪		
	NS			1◪ 2□	
	CV	1◪	2◪		
	AV				1◪ 2◪
	AR		1◪		
	AQ	1■	2◪ 3◪ 4◪	5◪	
Management	OC	1◪ 2◪ 3◪	4□		
	PD	1◪	3□	2□	4□
	CP	1□	2◪	3□	
	LC	1 ⊟			2◪
	IC		1◪		
Process and Technology	PI	1◪	2◪ 3◪		4◪
	MS	1◪	2◪		3◪
	CI			1◪ 2◪	
	TR	1◪		2◪ 3◪	
	TS	1■	2◪	3◪	4◪
	TI			1◪	2◪

Key:
□ Goal not satisfied
◪ Goal slightly satisfied
◪ Goal moderately satisfied
■ Goal mostly satisfied
■ Goal fully satisfied
⊟ Impact less than moderate
Numbers represent goal identifiers

Identify Initial Reuse Adoption Goal Set

Using this representation, identify critical success factor goals that are needed to attain complete coverage of lower stages. You should identify a group of critical success factor goals as reuse adoption goals for each stage up to and including the target.



The identified goals are your initial, prioritized set of reuse adoption goals, with those at earlier implementation stages having higher priority than those at later stages.

Tailor the Reuse Adoption Goals to Your Needs

From the initial set of reuse adoption goals, you should revise goals and priorities according to the following heuristics:

- ***Add, remove, or modify reuse adoption goals to more closely align them to your organizational and reuse objectives.*** If you have reuse adoption objectives that imply capabilities that are more than one implementation stage above the target, then you should revisit these objectives to ensure that they do not arbitrarily require you to take on unacceptable risk. If the objectives are valid, then you may want to identify an intermediate stage in the transition approach that does not include the objective.

  You should increase the priority of reuse adoption goals that meet or are most closely related to your objectives. You should increase the priority of reuse adoption goals that meet near-term objectives.

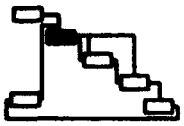
  Similarly, if a reuse adoption goal at the target stage does not relate to your organizational objectives, then you should consider lowering its priority or removing it. However, for stages lower than the target stage, removal of reuse adoption goals may expose you to added risk.

- ***Add or remove reuse adoption goals according to the business potential of the domains in which you develop software.*** Here, use the analysis from the domain assessment. The specific results of the domain assessment may support the addition of higher stage goals or necessitate their removal.

  For instance, if your organization has a strong product line orientation, a stable domain, and a good opportunity for future business, then you should be able to add critical success factor goals above the target stage as long-term reuse adoption goals. However, if your organization does not have a strong product line and you cannot predict the areas in which you will do future business, then you may need to remove some higher stage reuse adoption goals.

- ***Add reuse adoption goals that appear to provide the highest payoff in your current business environment.*** If your organization is committed to reuse and willing to assume a risk, you may be able to apply this rule and ignore the recommended implementation stages. Also, your customer may be willing to assume the risk in anticipation of the benefits that may come to the targeted program.

  Another situation in which you may be able to do this is if you have already developed or acquired reuse technology that makes attainment of the critical success factor goal a low-risk proposition.



In addition to identifying process-related goals, you can also identify goals related to the other aspects of the organizational profile and components of a reuse adoption strategy, such as the product and business model components, to define what is to be accomplished in these areas. Likewise, you can identify both near- and long-term goals to define the evolution of your reuse practices. Figures 3-3 and 3-4 provide examples of some near- and long-term goals.

- Product goal:

  Establish a base of reusable assets for building products.

- Business goal:

  Exploit reuse opportunities in the domain without increasing delivery cost and schedule.

- Process goals:

  - Increase accessibility and awareness of division software assets.
  - Provide mechanisms to ensure that developers look for, evaluate, and verify reusable assets.
  - Win management and staff commitment to practicing reuse.
  - Establish procedures for reuse cost accounting.

Figure 3-3. Example of Near-Term Goals

- Product goal:

  Establish a product line based on reusable architecture and assets.

- Business goal:

  Market the product line commercially.

- Process goal:

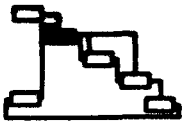
  Establish a leveraged reuse capability.

Figure 3-4. Example of Long-Term Goals

3.2.3 Identify Constraints

Purpose Identify constraints, i.e., any limits on the accomplishment of the reuse adoption goals or choice of strategies. Just as it is seldom useful to identify requirements without thinking of their eventual implementation costs, your identification of goals in the Establish Reuse Adoption Goals task should be performed interactively with the identification of constraints in this task.

Guidance Your constraints derive primarily from the requirements for integrating the reuse program into your existing development efforts.



Common sources of constraints include:

- ***System Engineering Integration.*** Your systems engineering organization may be limited in its ability to provide indications of future or potentially varying requirements for the reusable domain components. Additionally, it may be contractually locked into a development process. This fixed development process, for example, may dictate the timing of the development of requirements and design information in such a way that a reuse-driven software process is not practical.

- ***Budget.*** You must understand the constraints for different parts of the reuse program. For example, prospects for training budgets, independent research and development (IR&D) prospects, and contract funding should be identified. Although on a large program you may be able to recoup any investment costs over the course of the program, you need to understand how the funding has been allocated.

- ***Schedule.*** Existing and upcoming contracts may have hard deadlines for software deliverables. The reuse-driven software process that you put into place will need to be designed to fit into schedule constraints.

- ***Contracts.*** Current contracting guidelines place constraints on ownership of assets. *Acquisition Handbook, Central Archive for Reusable Defense Software* (CARDS 1992); *Proceedings: Software Reuse Legal Issues Workshop* (CARDS 1993b); and Appendix D describe these constraints. You need to examine your anticipated asset creating and asset utilizing software development efforts to determine the particular constraints.

- ***Organizational Policies.*** The policies of your own organization and those of your customers' organizations should be examined to determine if there are reuse-related constraints.

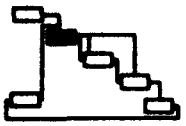
- ***Teaming and Subcontracting Arrangements.*** Agreements made between companies on a single contract may specify particular constraints on how organizations can cooperate on development of assets within a single program.

- ***Access to Assets and Expertise.*** You may be able to get only partial access to the requirements and technology experts on ongoing programs.

In the course of identifying constraints, you need to reexamine your organizational objectives, reuse adoption objectives, and reuse adoption goals to resolve conflicts with the constraints, i.e., to either get the constraints relaxed or to revise your expectations.

3.2.4 Identify Alternative Reuse Adoption Strategies

Purpose Identify potential reuse adoption strategies to meet the established reuse adoption goals and objectives.



A complete reuse adoption strategy should include:

- ***Product Approach.*** What end products will be affected by the reuse effort?

- ***Business Model.*** Who pays to build reusable assets? Who builds them? Who reuses them? What rights do the customer, developer, and reuser have to the assets? Who pays whom for assets that are reused? Who assumes liability for reused assets?

- ***Process Approach.*** What processes, methods, and standards will be used to exploit reuse in developing or maintaining software products, to create reusable assets, and to manage product and asset development? What types of software assets will be produced and reused?

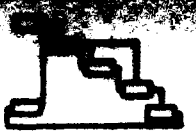
- ***Organizational Approach.*** What organizations will be affected by the reuse program? What organizational structure will be used? Who will manage it?

- ***Environment Approach.*** What tools, automated and nonautomated, will be used to support the development process?

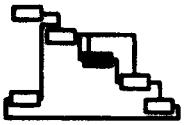
- ***Transition Approach.*** How will changes to the organization, its policies, procedures, processes, methods, standards, and tools be accomplished? What purchases, training, and consulting will be needed? What is the time frame for the changes?

Section 6 provides detailed guidance on the development of reuse adoption strategies.

At the end of this task, you will have developed one or more approaches in each of the six areas. Depending on your situation, these approaches may be independent of each other or they may be tightly linked. You review the alternative approaches with the sponsor at the completion of the Understand Context activity and analyze the alternatives in the Analyze Risks and Select Strategy activity.



This page intentionally left blank.



3.3 ANALYZE RISKS AND SELECT STRATEGY

Purpose Refine, evaluate, and select a reuse adoption strategy for implementing reuse.

Entrance Criteria

- The reuse program plan has been updated to include the reuse adoption goals and constraints and has been reviewed and approved by the sponsor.
- Alternative reuse adoption strategies have been identified.

Tasks

- Assess risks of alternative strategies.
- Analyze organizational impact of alternative strategies.
- Analyze economics of alternative strategies.
- Select a reuse adoption strategy.

Inputs Alternative reuse adoption strategies.

Controls

- Reuse program plan:
  - Reuse adoption objectives
  - Reuse adoption goals
  - Constraints
- Sponsor

Output Selected reuse adoption strategy

Mechanisms

- Reuse agent
- Reuse Economics Spreadsheet Model

Exit Criterion A reuse adoption strategy has been selected and approved by the sponsor.

In this activity, you refine the alternative reuse adoption strategies identified in the Understand Context activity; you may identify additional alternatives as well. Your job is to narrow the list of alternatives to one or a few strategies that best satisfy the objectives and then make a recommendation to the sponsor for a decision. You should seek involvement from all stakeholders to ensure that their issues and concerns with respect to the alternative strategies are addressed.

You can work toward a recommended strategy by iteratively analyzing, refining, and possibly combining or eliminating strategies until you have sufficient confidence in one or a few strategies to make a recommendation.

There are a variety of analytical techniques that you can use to carry out the analysis. This guidebook provides a risk checklist (Appendix C) to assist you in identifying risks. Other possible techniques include trade studies, utility analysis, red teams, prototyping, simulation, role playing, and more.

After the analysis is complete, you present the recommendation, rationale, and supporting analysis to the sponsor and possibly to the individuals whose cooperation is needed for success for a decision.

3.3.1 Assess Risks of Alternative Strategies

Purpose Identify, analyze, and evaluate risks for each alternative reuse adoption strategy to avert risks before they become impediments to the successful implementation of the reuse program.

Guidance Risks are the potential undesired outcomes associated with alternatives, constraints, and other factors and the effects of these outcomes on objectives and goals. Software Productivity Consortium (1992b) states that the following three characteristics are present in any risk:

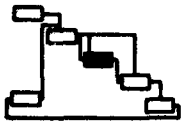
- ***Cost.*** Cost is the unfavorable outcome if an undesired event happens. The outcome must have a direct and negative impact to be considered a risk. If there is no negative impact or loss, then the undesired event is not a risk.
- ***Chance.*** There must be some chance of the undesired outcome occurring for it to be a risk. If there is no chance of the unfavorable outcome occurring, there is no risk. If the chance is 100%, then the outcome of the event cannot be avoided and must be accepted as a constraint.
- ***Choice.*** A risk occurs only if there is a choice involved. Having no choice or no options to avert an outcome means that the unfavorable outcome is beyond your control and should be viewed as a constraint.

For example, suppose a reuse adoption strategy calls for the establishment of a group to collect and catalog assets for inclusion in a reuse library, but the strategy has no provision for ensuring that the product developers use the library. In this case, there is a risk that the assets in the reuse library will not be reused. The cost includes the cost of collecting, cataloging, and maintaining the reuse library assets. The chance is the likelihood that the library will not be used. Choices include accepting the possible outcome by doing nothing or revising the strategy, such as assigning library consultants to work with each project, to increase the chance that the library will be used.

By explicitly identifying and addressing risks early in the planning effort, you gain the opportunity to reduce the chance of undesirable events occurring or the costs if the events occur.

Software Productivity Consortium (1992b) identifies the following steps for assessing risk:

- Identify risks
- Analyze risks
- Evaluate risks



Identify Risks

In this step, you identify any potential risks to fulfilling the reuse adoption objectives and goals. You begin by reviewing the objectives, goals, constraints, and alternative adoption strategies and asking "What could go wrong that would prevent us from achieving our objectives and goals with this strategy?"

In this step, you also categorize the identified risks according to their primary causes; this will improve your understanding of the risk and help you in determining how to address the risk. Software Productivity Consortium (1992b) identifies three primary causes of risk:

- ***Lack of Information.*** There is insufficient information to know with certainty the outcome of the strategy.
- ***Lack of Control.*** There is insufficient control over the organization or over factors influencing the organization to ensure that the strategy will accomplish the objectives.
- ***Lack of Resources.*** There are insufficient resources (e.g., time, money, expertise) to successfully perform the strategy and accomplish the objectives.

In the example of the reuse library, the primary cause is a lack of control. There is no mechanism to influence the developers to use the reuse library; thus, there is a risk that it will not be used.

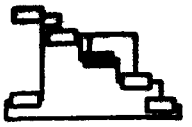
Below are some techniques to help you identify risks:

- ***Risk Source Checklists.*** A checklist is a useful reminder of where to look for risk. There can be generic as well as business-area-specific checklists. Appendix C provides an initial generic checklist of reuse-related risks, and Carr et al. (1993) provides guidance on using risk taxonomies.
- ***Assumption Analysis.*** Optimistic assumptions about the ability to achieve some result undoubtedly lead to risk. Often, you make these assumptions in ignorance about critical issues or by a tendency to be hopeful (Boehm 1989).
- ***Decomposition.*** Looking for poorly defined descriptions of what you must do and how you must do it can reveal areas of low understanding (Boehm 1989).

With some experience, you will find that it is relatively easy to identify risks. However, it is also easy to get carried away in identifying small risks. Try to focus on identifying risks that may have a significant impact on the program.

Analyze Risks

Not all risks are equally critical. To determine which risks are most critical to the program, you estimate the chance (probability) that the risk outcome will occur and the potential cost to the program if it occurs.



In the example of the reuse library, you might consider the chance that the developers will not use the reuse library to be high, based on similar experiences reported in other organizations. You might also consider the cost to be high because the reuse program hinges on the reuse of assets in the library. This suggests that the risk is critical.

Methods for estimating risk vary in their degree of precision. For most organizations starting a reuse program, a high degree of precision is not necessary and probably not feasible. Table 3-3 provides an example of a risk estimation scale. This approach helps you focus on the most severe risks.

Table 3-3. Example of Risk Estimation

Risk	Probability	Consequence	Severity
Developers will not use the reuse library	4	4	16
Program milestone will be missed	2	4	8
Probability, Consequence: 1–very low, 2–low, 3–moderate, 4–high, 5–very high Severity = Probability x Consequence			

Charette (1989) and U.S. Air Force (1988) provide additional methods for analyzing risks.

Evaluate Risks

In this step, you make a determination whether to accept a risk, take action to reduce the risk, or eliminate the reuse adoption strategy because it is too risky. Your decision depends on the organization's tolerance for risk. An aggressive organization might accept more risk at the prospect of gaining greater benefit if successful. A conservative organization might choose to follow a less risky alternative even though there may be less benefit.

Software Productivity Consortium (1992b) identifies the following categories of risk-aversion strategies:

- ***Risk Reduction.*** Strategies in this category attempt to reduce the likelihood of a risk occurring. Risk reduction strategies can be identified by determining the cause, i.e., lack of information, control, or resources, and identifying actions that could increase the information, control, or resources.

- ***Risk Protection.*** Strategies in this category attempt to reduce the cost of a risk if it occurs. Risk protection includes strategies that reduce the impact of the reuse program on the organization, such as "starting small" or initially applying the program to noncritical projects.

- ***Risk Transfer.*** Strategies in this category attempt to reallocate risks to areas or organizations better able to handle them. Risk transfer includes strategies that share risk, such as seeking external funding for

a portion of the program, or subcontracting a high-risk task to another organization.

- ***Risk Contingency Funding.*** Strategies in this category establish a reserve of resources, set aside for use if any of the identified risks occur. The reserved resources are usually time and money, such as allocating more time to high-risk tasks when generating a schedule.

- ***Risk Acceptance.*** Strategies in this category take no action to avert the risk.

In selecting an aversion strategy, you should estimate what the impact of the strategy is on the risk—does it reduce the likelihood or cost to an acceptable level and how much will it cost to implement the aversion strategy. You may even choose to implement multiple aversion strategies. The selected aversion strategy may then become another component of the reuse adoption strategy, or you may choose to implement the aversion strategy before making a decision on an adoption strategy.

In the example of the reuse library, the cause is lack of control. You may be able to increase control and, thus, reduce the likelihood that the library is not used by instituting an incentive program. In evaluating this aversion strategy, you should estimate what impact an incentive program would have on library use. One way you can do this is by reviewing reports on the use of incentives in other organizations. You should also consider how much an incentive program would cost. If it is not acceptable, then you should consider alternative aversion strategies.

3.3.2 Analyze Organizational Impact of Alternative Strategies

Purpose

Determine the organization's capacity or readiness for the changes proposed in each alternative reuse adoption strategy, anticipate resistance to these changes, and identify strategies for reducing resistance.

Guidance

Organizational readiness for change and resistance to change are risk areas. As such, you can assess them as you would any other risk. However, they are included here as a separate task for emphasis because of their importance to the success of a reuse program.

Analyze Readiness for Change

Software Productivity Consortium (1993d) suggests you do the following when analyzing your organization's readiness for change:

- Examine prior technology change efforts in your organization. Successful change experiences increase the likelihood of acceptance; unsuccessful experiences greatly decrease the chances of success.

- Understand other changes that have either just occurred or that might be occurring simultaneously with the reuse program. If possible, try to stabilize other factors while implementing the reuse program.

- Understand the risk associated with implementing the reuse program now versus waiting until later. Will you lose an advantage by not acting now, or can you wait until conditions are better?
- Look at the expectations and culture of the areas of the organization to be affected by the reuse program. Do they perceive a need for the change or is the change being forced on them?
- Do the projects targeted for adopting reuse have the extra resources to learn and use the reuse technologies?
- Does the staff have the required skills to use the reuse technologies? Are there adequate time, resources, and training available to train the users?

Analyze Resistance to Change

Throughout the reuse program you can expect that some users will disagree with what you propose—listen carefully to them. Users often resist change because they feel they will lose power or prestige, they like their current way of doing business, or they would prefer another approach to reuse. Software Productivity Consortium (1993d) suggests the following approaches for countering resistance:

- Involve the users in decisions and performance of relevant adoption tasks.
- Arrange for the reuse champion or experts in the use of the proposed reuse technologies to be permanently or temporarily transferred to the group using the technologies.
- Use management directive ("you will do this") to get people to try the new technologies. This technique should be used sparingly.

3.3.3 Analyze Economics of Alternative Strategies

Purpose

Understand the economic impact of each alternative reuse adoption strategy and determine its economic desirability.

Guidance

Unless your organization has unlimited resources, economics will be a key decision driver for the sponsor, thus making economics an essential part of your analysis. You can expect to be required to provide answers to the following questions for the sponsor:

- How much will it cost to institutionalize reuse?
- How much will it cost to sustain the institutionalized reuse practice?
- How long will it take to institutionalize reuse?
- How long will it take to break even on my reuse investment?

- How long will it take to achieve the first significant reuse?
- What are the expected benefits (e.g., productivity, quality, cost reduction, time to market, market share, competitive advantage)?

Unfortunately, finding answers to these questions is problematic at best. It is particularly difficult if you lack historical data on which to base the estimates. Thus, there is a risk that the estimates will be inaccurate. To reduce this risk, you should seek help from someone in the organization or outside of it who is experienced in cost estimation. The cost sources you should take into consideration, as well as approaches you can use to answer these questions, are described below.

Identify Sources of Cost

The items listed below identify possible sources of cost associated with institutionalizing and sustaining a reuse practice. You can review the sources listed below with respect to the reuse adoption strategies to identify sources of costs in the strategies. The actual sources vary with the strategies. As you identify the sources of cost, you also indicate whether they are one-time costs associated with institutionalization or recurring costs associated with sustaining the reuse practice.

- Management costs
  - Planning, directing, and monitoring implementation of the reuse program
  - Planning, directing, and monitoring asset creation and utilization activities
  - Cost of financing the reuse investment
  - Promotion or advocacy
- Utilization costs
  - Identifying, evaluating, and selecting reusable assets or modeling an application from reusable assets
  - Adapting or generating reusable assets for use in a product
  - Maintaining reused assets over the life of the product
- Creation costs
  - Performing domain analysis and modeling
  - Acquiring, reengineering, or developing reusable architectures
  - Acquiring, reengineering, or developing reusable components
  - Asset verification or certification

  - **Application engineering process support and automation**
  - **Maintaining and enhancing reusable assets**
  - **Establishing and operating a reuse library**
- Support costs
  - Training
  - Consulting
  - Selecting, acquiring, or developing tools for reuse
  - Metrics collection and analysis
  - Process definition and evolution

Analyze Costs and Benefits

You can perform a cost/benefit analysis to determine whether the expected benefits from implementing a strategy outweigh the costs. You can also use the analysis results as a basis for comparing the alternative strategies.

To perform the cost/benefit analysis, list the expected costs and benefits for each strategy. You can use the cost areas listed above to aid in identifying possible costs, then contrast these costs with what it would cost if you did not implement the strategy. You can also identify less tangible but significant costs and benefits, such as the impact on the organization's morale, prestige, or competitive advantage.

Qualitative approaches to estimating costs and benefits can be just as effective as quantitative approaches when comparing costs to benefits and comparing strategies, particularly if you lack historical data. For example, you can rate the impact of the strategy on the cost/benefit factors using a scale (e.g., low, medium, high). For this approach to be effective, you need to be consistent in how you determine the ratings.

Estimate Cost and Schedule

Although qualitative approaches can be effectively used as a basis of comparison, at some point, the sponsor will want to know how much it will cost, quantitatively, so the sponsor can acquire and allocate the needed resources. You can use a work breakdown approach to estimate cost and schedule in support of the sponsor's decision making.

In a work breakdown approach, you recursively decompose a strategy into components, activities, and tasks until you reach a point where you can estimate the cost and duration of each element with some confidence. You can use estimate ranges to reflect uncertainty in the estimates, i.e., provide a pessimistic and optimistic estimate for each element.

Analyze Model Results

Models provide an additional approach to estimating economics factors. You can use the Consortium's reuse economics models to estimate the cost of

developing systems with reuse, ROI, break-even number of systems, the effect of reuse on software quality, and productivity and to evaluate incremental investment strategies.

For example, you can use the equation shown below to estimate the ROI. You can estimate the parameters of the equation using historical data, the estimates from the approaches described above, and the results of the domain assessment.

$$ROI = \left[\frac{N \cdot E \cdot (C_{VN} - C_{VR})}{C_{DE}} - 1 \right] \cdot 100$$

where:

- N = Expected number of applications (versions, products, etc.) to be produced by the organization
- E = Efficiency of the library infrastructure; the ratio of the amount of reused code in an application system to the available reusable code
- C_{VN} = Unit cost per code size of new code developed for an application system
- C_{VR} = Unit cost per code size of reusing code from the reuse library in an application system
- C_{DE} = Unit cost per code size of the investment in the reuse program

Software Productivity Consortium (1993c) provides a detailed description of the Consortium's reuse economics models.

You can use the Consortium's Reuse Economics Spreadsheet Model to estimate economic factors, like ROI, and to perform a sensitivity analysis of the results (Software Productivity Consortium 1993c). Sensitivity analysis is useful when input values are uncertain. It can help you discover if variation within the range of uncertainty would change your decisions.

Figure 3-5 illustrates the economics spreadsheet input for a sensitivity analysis. Each row represents a variation of one or more parameters in the sensitivity analysis. Figure 3-6 illustrates the economics spreadsheet output for a sensitivity analysis. It shows how the ROI and break-even number of systems vary for differing values of library efficiency. The spreadsheet model is based on Microsoft Excel.

3.3.4 Select a Reuse Adoption Strategy

Purpose Select a reuse adoption strategy that satisfies the organization's reuse adoption objectives and goals as a basis for developing a reuse action plan for implementing reuse.

File Family Graphs Window

Normal

A22 =1+ModelsDB.xls!A21

ModelsDB.xls

	A	B	C	D	E	F	G	H
1	FamilyNo	N	R	CDE	CVN	CVR	ST	SS
2	Family Number	Number of Application Systems	Proportion of Code Reuse	Unit Cost of Reuse Program Investment	Unit Cost of New Code	Unit Cost of Reused Code	Size of Reuse Library	Average # of Applicat Systems
4	1	1	0.30	7.50	5.00	0.50	30.00	100
5	2	1	0.50	7.50	5.00	0.50	50.00	100
6	3	1	0.70	7.50	5.00	0.50	70.00	100
7	4	2	0.30	7.50	5.00	0.50	30.00	100
8	5	2	0.50	7.50	5.00	0.50	50.00	100
9	6	2	0.70	7.50	5.00	0.50	70.00	100
10	7	3	0.30	7.50	5.00	0.50	30.00	100
11	8	3	0.50	7.50	5.00	0.50	50.00	100
12	9	3	0.70	7.50	5.00	0.50	70.00	100
13	10	4	0.35	7.50	5.00	0.50	35.00	100
14	11	4	0.75	7.50	5.00	0.50	75.00	100
15	12	4	0.95	7.50	5.00	0.50	95.00	100
16	13	4	0.35	7.50	5.00	1.00	35.00	100
17	14	4	0.75	7.50	5.00	1.00	75.00	100
18	15	4	0.95	7.50	5.00	1.00	95.00	100
19	16	4	0.35	7.50	5.00	1.50	35.00	100
20	17	4	0.75	7.50	5.00	1.50	75.00	100
21	18	4	0.95	7.50	5.00	1.50	95.00	100
22	19	5	0.30	7.50	5.00	0.50	30.00	100
23	20	5	0.50	7.50	5.00	0.50	50.00	100
24	21	5	0.70	7.50	5.00	0.50	70.00	100

Figure 3-5. Reuse Economics Spreadsheet Input Window

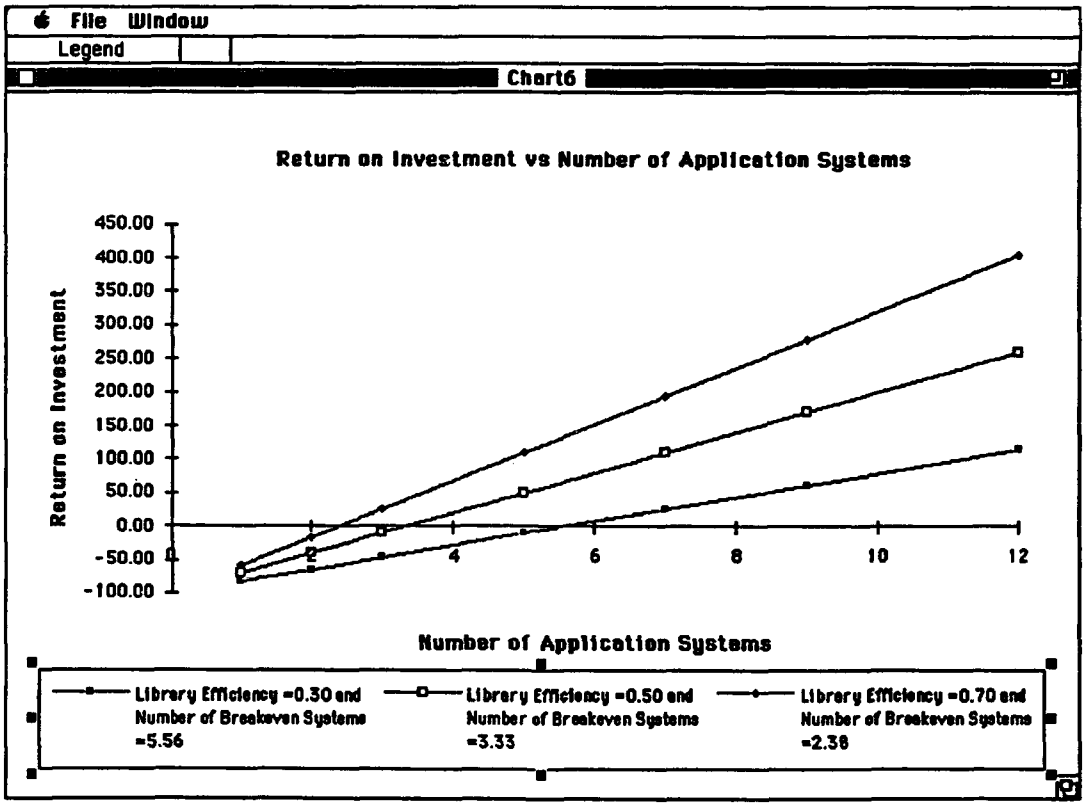


Figure 3-6. Return on Investment vs Number of Application Systems Graph Window

Guidance

At some point, you must bring together the analyses, weigh the results, make a recommendation, and seek approval from the sponsor on what strategies to pursue. These can be accomplished in the following steps:

- Evaluate alternative strategies.
- Make a recommendation.
- Seek and publicize approval.

Evaluate Alternative Strategies

In this step, you use the analysis results to compare and rank the alternative reuse adoption strategies. The comparison should be based on the following factors:

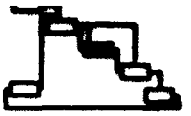
- The extent to which the strategy meets or exceeds the reuse adoption objectives and goals.
- Compliance with constraints.
- Level of risk—is it acceptable, can it be reduced to an acceptable level?
- Organizational readiness—will the organization accept and support the strategy?
- Expected costs and benefits—do the benefits sufficiently outweigh the costs, do the expected costs and schedule comply with any budget and time constraints?

You can use a variety of mechanisms, varying in formality and precision, to compare and rank the alternatives. Possibilities include:

- ***Rating Scales.*** Rate each alternative on a predefined scale. Table 3-4 provides an example.
- ***Voting.*** Provide each member of the group with a fixed number of votes that they can apportion over the alternatives.

Table 3-4. Example of a Rating Scale

Evaluation Criteria	Alternative Reuse Adoption Strategy Ratings		
	Strategy A	Strategy B	Strategy C
Meets goals and objectives	4	5	5
Complies with constraints	5	4	2
Risks are acceptable	4	4	2
Organization will buy in	4	5	3
Benefits outweigh costs	3	4	4
Rating scale: 1-decidedly inadequate, 2-inadequate, 3-borderline, 4-adequate, 5-highly adequate			



- ***Quantitative Analysis.*** This includes techniques, such as the Analytic Hierarchy Process (Saaty 1980), and tools, such as Expert Choice (Forman et al. 1990) based on the Analytic Hierarchy Process, that assign numerical weights and scores to qualitative and quantitative factors to produce a composite score.

Use the technique with the degree of precision that best meets your needs, is within your ability to provide precision, and will be acceptable to the sponsor.

Make a Recommendation

In this step, you use the evaluation and analysis results to make a recommendation on how to proceed with the reuse program. Possible outcomes include:

- One strategy is a clear winner that you can recommend.
- There are multiple winning strategies to choose from but with varying costs and benefits. If you are unable to reach a decision, then present the winning strategies to the sponsor for a decision.
- There are one or more acceptable-to-borderline strategies but no clear winning strategies. In this case, you may decide to do further analysis, go back to the Understand Context activity to revise the strategies (e.g., by scaling down, narrowing the scope, or staging the implementation), or recommend that multiple strategies be pursued in parallel until a winner can be selected.
- There are no acceptable strategies. In this case, you may need to go back to the Understand Context activity to revise the objectives and goals and search for new strategies, defer the reuse program until a later time when conditions might be better, or recommend that the reuse program be canceled.

Seek and Publicize Approval

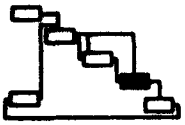
For the recommended strategy to succeed, you need the support of the organizations to be affected by the strategy and the approval of the sponsor. To accomplish this, you should provide ample opportunity for the affected organizations to review and comment on the recommendation. Based on the review results, you may determine that you need to revise the recommendation.

After you determine that you have adequately addressed any review comments, you present the recommendation to the sponsor to obtain approval. The presentation should include:

- A description of the recommendation
- The rationale for the recommendation
- The impact of the recommendation on the organization
- The estimated cost and time frame to implement the recommendation

You may also want to identify the alternative strategies that were rejected and the rationale for rejecting them. However, a good rule of thumb is to keep the presentation brief and to the point. After you have obtained approval for the strategy, you should publicize the approval and strategy to the organizations to keep them informed and prepare them for the changes ahead.

This page intentionally left blank.



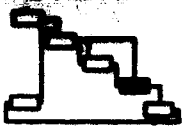
3.4 PLAN IMPROVEMENTS

Purpose	Develop a plan to implement reuse based on the selected reuse adoption strategy.
Entrance Criterion	A reuse adoption strategy has been selected and approved by the sponsor.
Tasks	• Identify transition actions and staffing. • Identify equipment and facilities changes. • Identify success criteria and data collection requirements. • Develop schedule and resource profile. • Obtain commitment to implement.
Input	Selected reuse adoption strategy
Controls	• Reuse program plan: – Reuse adoption objectives – Reuse adoption goals – Constraints • Sponsor
Output	Reuse action plan
Mechanism	Reuse agent
Exit Criterion	The sponsor and the managers responsible for implementing the reuse action plan commit to implementing the plan.

During this activity, you identify actions and schedule resources for putting the selected reuse adoption strategy into effect. The reuse action plan is a transition plan; i.e., you include actions for putting the product, business model, organization, process, and environment approaches of the adoption strategy in place, e.g., training people in reuse techniques, documenting policies and procedures, and acquiring tools. You do not include actions for developing the domain or building a system; these will be planned by the appropriate domain or project managers. Rather, your planned actions result in the appropriate domain or project managers taking the necessary steps to execute the organization's approach to reuse. Hence, you derive many of the actions for the reuse action plan from the transition approach of the adoption strategy.

Areas for you to address in the action plan include transition actions, equipment and facilities changes, success criteria and data collection requirements, schedule, and resources. The following sections describe these in detail. Appendix F provides an annotated reuse action plan.

You can use a number of common techniques in support of this activity, such as Gantt charts, the Critical Path Method, and the Program Evaluation and Review Technique (PERT), and tools, such as Microsoft Project.



3.4.1 Identify Transition Actions and Staffing

Purpose Identify actions required to implement the selected reuse adoption strategy, dependencies between actions, and resources required to implement the actions.

Guidance In this activity, you identify the actions required to implement the selected reuse adoption strategy. Action descriptions should include the:

- Purpose and description of work to be performed
- Inputs and expected results or artifacts
- Resources and personnel required
- Estimated cost and duration
- Dependencies on other actions (e.g., via PERT diagrams)

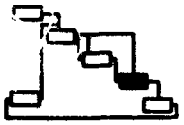
Table 3-5 provides an example of a description for an action.

Table 3-5. Example of an Action Description

Develop Training Materials	
Action	Develop training materials on the reuse-driven software process and supporting tools for program management and technical staff.
Deliverables	2-day course and course workbook
Effort	3 staff months
Start and Completion Dates	April 1 and May 31
Personnel	Education and training department Consulting support
Charge Code	1001-974

A rule of thumb for writing action descriptions is to identify them at about 1 to 3 staff-months of effort. This level of detail is usually appropriate to allow estimation of resource requirements but not so restrictive that it limits flexibility during implementation. Additionally, the actions should be identified following organizational lines to the lowest level practical to allow the clearest assignment of the actions.

Transition actions are those actions required to implement the reuse adoption strategy. Transition actions may include:



- ***Risk Aversion Strategy.*** The risk analysis from the Analyze Risks and Select Strategy activity is likely to identify specific risks that you can avert before you commit to the full reuse effort. Typical risks that can be averted in the transition phase are ensuring that the staff is able to practice methods that are required to perform the process and ensuring that the target domain is suitable for development of domain assets.

- ***Pathfinder Efforts or Pilot Projects.*** These are specific means for risk reduction that have been successfully used in introducing new technology (Redwine, DelFosse, and Spencer 1992; Elkin, Gardner, and Ireland 1991). In these efforts, the technology is applied on a small scale before its use in a larger project. The pilot effort allows for revision and tuning of methods and automation support before their use by the project as a whole. Additionally, participants in a pathfinder effort can act as trainers or leaders in the later use by the entire project.

- ***Customer Approval and Required Contract or Funding Changes.*** If the new business model requires a change to the normal contracting procedures or if it requires specific recognition by the customer of the use or development of proprietary software, then you need to devote resources to getting the approval.

- ***Establishment of the Data Collection and Metrics Programs.*** You need to put the mechanisms in place that will be used for establishing the value of a reuse program. Those mechanisms are identified in Section 3.4.3.

- ***Training.*** You need to identify and implement training in the reuse-driven software development process and methods, the particulars of the software development environment, and the programmatic issues related to the adoption plan.

- ***Documentation.*** Implementation of the adoption strategy may require changes to organizational documents, such as policies, procedures, standards, and guidelines.

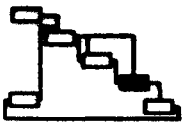
- ***Policy Change Implementation.*** You may need to dedicate effort to working through official channels to change organizational policies. If under contract, you need to examine the terms and provisions of the contract.

3.4.2 Identify Equipment and Facilities Changes

Purpose Identify equipment and facilities required to implement the reuse activities.

Guidance Typical items that should be considered for this category are the:

- Development environment tools that are required (e.g., code management systems or component library mechanisms)



- Infrastructure (e.g., networked workstations) that allows for cooperative development of assets

Note that the reuse adoption strategy does not necessarily require any additional equipment over and above that required for traditional software development. For example, library mechanisms are not required for many reuse-driven software development processes (e.g., those defined in Software Productivity Consortium [1993b]).

3.4.3 Identify Success Criteria and Data Collection Requirements

Purpose Provide a quantitative basis for determining whether the reuse activities are successful. Identify mechanisms required for collecting the data.

Guidance There are three primary reasons it is important to quantitatively measure the success of the reuse efforts:

- Your ability to improve a process depends on your ability to measure it.
- Quantitative measures help the sponsor and the organization make the right decisions on continued funding or expansion of the reuse activities.
- Establishing a metrics program now provides a basis for more accurate cost/benefit analyses for future reuse adoption efforts.

You can establish a metrics program using the goal, question, metrics paradigm (Basili and Weiss 1984). The goals are the reuse adoption objectives you defined in the Initiate Reuse Program activity and the reuse adoption goals you defined in the Understand Context activity. The questions are the questions that must be answered satisfactorily to indicate that the goal has been met. The metrics are the specific quantities that you collect to determine the answers to the questions. Table 3-6 provides an example.

Table 3-6. Example of Goal, Question, and Metrics

Goal	
Increase accessibility and awareness of division software assets among the developers.	
Question	**Metrics**
Do all of the developers have access to the division's assets?	Number of developers that have electronic access to the assets from their office
Is it easy for a developer to access the division's assets?	Average number of steps to retrieve an asset (for a representative sample of assets)
Do all of the developers know what assets are available that may be relevant to their work?	Number of developers that browsed the assets in the past month, 6 months, year

You should strive to identify metrics that are easy to collect and unobtrusive to the users. For instance, labor utilization metrics can sometimes be collected through time sheets that are already in use for charge control. In addition to identifying the data to be collected, you should also identify when, how, and by whom it is collected. You then baseline the metrics by determining their current value.

Depending on the organization's current measurement activities, you may need to develop the metrics and data collection techniques in concert with an overall measurement program for the organization. Software Productivity Consortium (1992c) provides guidance in the general area of measurement.

3.4.4 Develop Schedule and Resource Profile

Purpose Schedule the actions to satisfy dependencies, resource requirements, and constraints. Identify levels of resources (money and people) required and available over the term of the schedule.

Guidance Schedules should identify start and completion dates for all actions, should be linked to the related milestones in the application projects that are being supported, and should identify the major decision or review points for the reuse program implementation.

In developing the schedule, you need to take into account the cost and duration of the actions, availability of resources for performing the actions, dependencies between the actions, and any "hard" deadlines for action completion. You can use the cost and schedule estimates developed for the selected reuse adoption strategy in the Analyze Risks and Select Strategy activity.

You develop the resource profile to show the level of resources required versus the level of resources available over the term of the schedule. This information can help identify where you need to make adjustments in the schedule to accommodate gaps. It helps identify where you need to acquire additional resources and also serves as input to the organization's budget planning. To support the organization's budget planning, you should identify current-year as well as out-year resource requirements.

3.4.5 Obtain Commitment to Implement

Purpose Obtain commitment to implement the plan.

Guidance As in the previous activities, you should have the reuse action plan reviewed by the organizations affected by the plan. In this case, you could also have all of the managers, who have a role in implementing the plan, "sign off" on the plan to indicate their approval and commitment. You then submit the plan to the sponsor for review and approval. In addition, you seek the sponsor's commitment to go ahead with the implementation. You may have to iterate through the tasks of this activity several times before you obtain the sponsor's commitment.

After the sponsor commits to the plan, he should demonstrate his commitment to the rest of the organization. For example, he could distribute a memorandum to the organization that states the importance of the reuse program to the organization, states his commitment to implementing the reuse action plan, and solicits the support of the organization in implementing the reuse action plan. The sponsor should also demonstrate his commitment by providing the resources called for in the plan.

3.5 IMPLEMENT IMPROVEMENTS

Purpose Enact the reuse action plan.

Entrance Criterion The sponsor and the managers responsible for implementing the reuse action plan have committed to its implementation.

Tasks

- Obtain resources and perform work.
- Monitor progress against the reuse action plan.

Input Current reuse situation, characterized by the organization's mission, structure, personnel, products, customer and supplier relationships, processes, standards, tools, market, and technology base

Control Reuse action plan

Output Improved reuse situation, characterized by an improvement in the organization's structure, personnel, products, customer and supplier relationships, processes, standards, tools, market, and technology base

Mechanisms

- Reuse agent
- Users

Exit Criterion All actions in the reuse action plan have been closed.

In this activity, you obtain the necessary resources and perform the actions identified in the reuse action plan. You assign specific individuals to execute and manage the tasks, ensure that the individuals have the resources they need, and authorize work on the tasks to begin. You also collect data and monitor the progress of the program against the reuse action plan. As with any plan, you should expect to make adjustments to the plan based on progress, results, and any problems or opportunities that may arise.

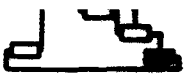
3.5.1 Obtain Resources and Perform Work

Purpose Obtain the necessary resources and perform the actions as identified and scheduled in the reuse action plan.

Guidance In this activity, you obtain the resources identified in the reuse action plan (equipment, personnel, funding, etc.) and allocate the resources to the defined tasks to initiate work. The responsible individuals then perform the tasks. Procedures for obtaining and allocating resources vary with each organization; use the procedures established for your organization.

Here are some things you can do to facilitate a successful implementation:

- Ensure that individuals understand their assigned tasks by reviewing the task descriptions with them, showing them how their tasks relate to other tasks, and explaining how their effort contributes to the overall effort.



- Maintain communication with the implementation team, end users, and management; meet periodically, provide status to management, and publicize the results.

3.5.2 Monitor Progress Against the Reuse Action Plan

Purpose Monitor progress against the reuse action plan and make any necessary adjustments.

Guidance To maintain control over the progress and direction of the reuse program, you need to monitor progress against the plan. Progress is monitored against the plan to ensure that the plan is enacted as specified, i.e., that tasks are initiated and completed as scheduled.

Here are some steps you can take to monitor progress against the reuse action plan:

- Collect the data identified in the reuse action plan.
- Compare actuals to estimates (e.g., task start time, task completion time, task resources).
- Identify new indicators of risks:
  - Is someone associated with the program becoming unhappy?
  - Is the latest total cost estimate much different than the last estimate?
  - Do the key program indicators (usage, cost, schedule, resource utilization) vary significantly from the projected indicators?
- Adjust the plan to account for the actual progress. This could involve adding new tasks, deleting tasks, rescheduling tasks, reallocating resources, etc. If significant changes to the plan are necessary, you should cycle back to the Plan Improvements activity.

3.6 REVIEW AND UPDATE REUSE PROGRAM

Purpose	Review progress of the reuse program against the reuse adoption objectives and goals and update the reuse program plan to reflect actual progress.
Entrance Criterion	A key milestone in the reuse action plan has been reached.
Tasks	• Monitor progress against the reuse program plan. • Update reuse program plan. • Obtain commitment to proceed.
Input	Improved reuse situation, characterized by an improvement in the organization's structure, personnel, products, customer and supplier relationships, processes, standards, tools, market, and technology base
Controls	• Reuse program plan • Sponsor
Output	Update reuse program plan.
Mechanism	Reuse agent
Exit Criterion	The reuse program plan has been updated and approved by the sponsor.

In this activity, you collect data and monitor the progress of the program against the reuse adoption objectives and goals documented in the reuse program plan. As with any plan, you should expect to make adjustments to the reuse program plan based on progress, results, and any problems or opportunities that may arise.

Upon completion of key milestones, you should make a determination whether the objectives have been met, to what extent they have been met, and whether there are additional objectives or goals that should be addressed. At this point, you may choose to close the reuse program, e.g., because the objectives have been met and reuse has been successfully institutionalized, or you can cycle back to the Understand Context activity and continue the process, e.g., because the objectives have not been met, making it necessary to replan, or because you wish to further improve the organization's reuse capability.

3.6.1 Monitor Progress Against the Reuse Program Plan

Purpose	Monitor progress against the reuse adoption objectives and goals.
Guidance	To maintain control over the progress and direction of the reuse program, you need to monitor progress against the goals. Progress is monitored against the goals to determine whether the program is fulfilling the reuse adoption objectives. It is possible that the effort can proceed as planned but not fulfill the objectives.

Here are some steps you can take to monitor progress:

- Collect the data identified in the reuse action plan.
- Assess whether the reuse adoption goals have been met or will likely be met upon completion of the reuse action plan.
- Assess whether reuse adoption goal satisfaction will fulfill the reuse adoption objectives.

3.6.2 Update Reuse Program Plan

Purpose Update the reuse program plan to reflect any changes in reuse adoption objectives, goals, or strategies based on program performance.

Guidance Based on the assessment of the program's progress against the reuse adoption objectives and goals, you take the appropriate action:

- If you determine that the objectives or goals will not be met upon completion of the reuse action plan, then you can:
  - ***Adjust the reuse action plan.*** When you determine that there is nothing wrong with the strategy, but the plan is not successfully implementing the strategy. In this case, you cycle back to the Plan Improvements activity to revise the plan or develop a new plan that will better meet the goals.
  - ***Adjust the strategy.*** When the program's performance indicates that the reuse adoption goals will not be met with the current strategy. In this case, you cycle back to the Analyze Risks and Select Strategy activity. You may either revise the current strategy or choose an alternative strategy.
  - ***Adjust the goals.*** When you determine that no reuse adoption strategy will adequately meet the goals. In this case, you cycle back to the Understand Context activity to redefine the goals.
  - ***Relax the objectives.*** When you determine that the objectives cannot be met as stated.
  - ***Terminate the program.*** When it is not acceptable to adjust the objectives.
- If you determine that the reuse adoption goals have been met, then you can:
  - ***Close the reuse program.*** You have successfully institutionalized reuse and can now let it operate on its own.
  - ***Continue to monitor the organization's reuse practice to ensure that the organization continues to meet the reuse adoption goals and objectives.***

- ***Start with the successfully institutionalized reuse practice as the current baseline and continue to improve upon the practice.*** In this case, you cycle back to the Understand Context activity to reassess your situation and define new goals.

After you determine an appropriate course of action, you should update the reuse program plan to reflect any changes.

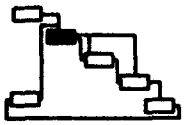
3.6.3 Obtain Commitment to Proceed

Purpose Reaffirm sponsorship and obtain commitment to continue the program according to the reuse program plan.

Guidance As in the previous activities, you should submit the updated reuse program plan to your sponsor for review and approval. In addition, you seek the sponsor's commitment of the resources necessary to go ahead with the program.

This page intentionally left blank.

4. DOMAIN ASSESSMENT



The domain assessment method supports the Assess Reuse Potential task of the Understand Context activity.

Purpose

Understand the potential for reuse in your business area to help determine how much to invest in reuse and where to focus your investment.

Entrance Criterion

The sponsor has committed the resources to perform the domain assessment.

Tasks

- Organize the domain assessment team.
- Identify specific product domains to assess.
- Assess domain factors.
- Develop assessment findings.
- Develop supporting material.
- Report domain assessment findings.

Inputs

- Organizational profile
- Current reuse situation, characterized by the organization's product plans, market information, existing assets, product family requirements, domain history, technology trends, and standards

Controls

Reuse adoption objectives

Outputs

- Domain assessment findings
- Supporting material
- Findings presentation
- Domain assessment report

Mechanisms

- Domain Assessment Model
- Domain experts

Exit Criterion Domain assessment findings and supporting material have been reviewed and approved by the sponsor.

Domain assessment is designed to provide enough information to make rational business decisions without the extensive commitment of resources needed for detailed study of the domain (e.g., domain analysis procedures defined by the Consortium, STARS, SEI, etc.). The Domain Assessment Model is defined in Appendix A. It focuses on the attributes of a domain that normally favor reuse. The selected team can perform the assessment in less than a day assuming they are familiar with the Consortium domain assessment. However, the final step, developing supporting material, will require whatever time is necessary to satisfy the sponsors' needs. If the assessment team is not familiar with the method, they should be trained in it before performing the assessment.

4.1 ORGANIZE THE DOMAIN ASSESSMENT TEAM

Purpose Select people who have the knowledge and experience necessary to assess the domain factors and to organize them into a team.

Guidance To ensure an effective and credible assessment, you must have the right people and they must be adequately prepared. The major steps you take to complete this task include selecting the assessment team members, scheduling the assessment, and training the assessment team members.

Select Team Members There are three roles to be filled when organizing an assessment team: facilitator, assessor, and scribe.

- ***Facilitator.*** The facilitator is responsible for conducting the assessment in a timely, effective, and impartial manner. The facilitator ensures that the discussion focuses on the issues, everyone has a fair chance to speak their views, the discussion leads to results, and the assessment keeps to its schedule. The facilitator does not make judgments regarding the domain. The facilitator should be knowledgeable in domain assessment and the Domain Assessment Model and be experienced in leading a group.

- ***Assessor.*** The assessor is responsible for making technical judgments on the domain and the organization's assets with respect to the factors identified in the Domain Assessment Model, identifying the reuse opportunities, identifying constraints that must be faced, and developing the findings. Collectively, the assessors should be knowledgeable in the domain and organization's assets.

- ***Scribe.*** The scribe is responsible for taking notes on the assessment discussions and results.

Presumably, the managers understand the domains to be studied well enough to select appropriate people. In difficult cases, it may be necessary for experts to clarify the domain definition (as in the next task) and identify the kinds of expertise needed to evaluate it. We suggest four sources of experts that you may use to build an assessment team:

- ***Project Staff.*** People who have worked on existing versions of the product or related systems.
- ***Personnel Outside the Project but Within the Organization.*** Same qualifications as project staff.
- ***People Hired for Their Expertise.*** People who worked on similar or related systems while with a previous employer.
- ***External Consultants.*** Consultants who have substantial knowledge of the target domain.

You usually have from four to eight individuals participating as assessors; any more makes it difficult to achieve consensus, any less limits the representation and may cause people to question the validity of the results. You should try to select the assessment team so that:

- The team provides a good cross-representation of the products you are assessing. You can, and are encouraged to, include a mix of management and technical staff.
- The choice of team members will facilitate buy-in to the assessment results from the remaining organization, i.e., you want to include opinion leaders.

You should have the same team members participate throughout the assessment. However, you may want to call in specialists for certain issues.

Selecting the assessment team members also requires that you get the team members to commit to conducting the assessment—getting this usually depends on the schedule.

Schedule the Assessment

In scheduling the assessment, you should allot 1 day to doing the assessment and developing the findings. You should also schedule 1 hour for presenting the assessment findings to the sponsor.

In addition to scheduling the team members' time, you need to schedule the room and any necessary audiovisual equipment.

Train the Assessment Team

To ensure that the assessment team is adequately prepared for the assessment, the team members should be instructed on the:

- Purpose of the domain assessment
- Domain assessment procedures
- Expected use of the assessment results
- Domain Assessment Model

All assessment team members should participate in this training. The training could be conducted in a classroom format (usually 2 hours) or via reading material.

4.2 IDENTIFY SPECIFIC PRODUCT DOMAINS

Purpose

Identify and describe the specific domain or domains to be assessed.

Guidance

The domain or domains of interest are generally defined during the initial phase of a reuse adoption effort. This should be further elaborated by asking the assessment team what areas have the greatest opportunities to apply reuse. The objective in this task is to establish clear definitions of what is being assessed and should be completed before the team sits down to do the assessment. There are several possible situations:

- ***A Single Domain.*** This is desirable when the domain is of reasonable size and a top level analysis is sufficient.
- ***Several Domains.*** This is appropriate when an organization has several product domains and some evaluation is needed to help identify the relative benefits of reuse in each.
- ***Several Subdomains.*** It may be desirable to perform the analysis below the top level for a domain to get more detailed information. Various ways of breaking up a domain are discussed below.

Defining the domain to be analyzed is critical to keeping the assessment team focused on the products to be produced. Domains have been variously defined as:

- A coherent business area
- A functional area covered by a family of systems
- A functional area in which similar requirements exist across systems

A good place to begin in defining a domain is often the organization chart (see Figure 4-1). One reason for doing this is to ensure that the organization that is building a reuse program has the charter for the domain. This is necessary to keep full control over the process of developing and using the assets.

Keep in mind that domain assessment is not restricted to domains that represent complete systems sold to external customers. Your product could just as well be components or tools, and the customer could be other divisions of your own organization.

It may be desirable to work below the top level domain to get a more detailed description of your reuse opportunities. You might pick the next level of detail along one of several dimensions:

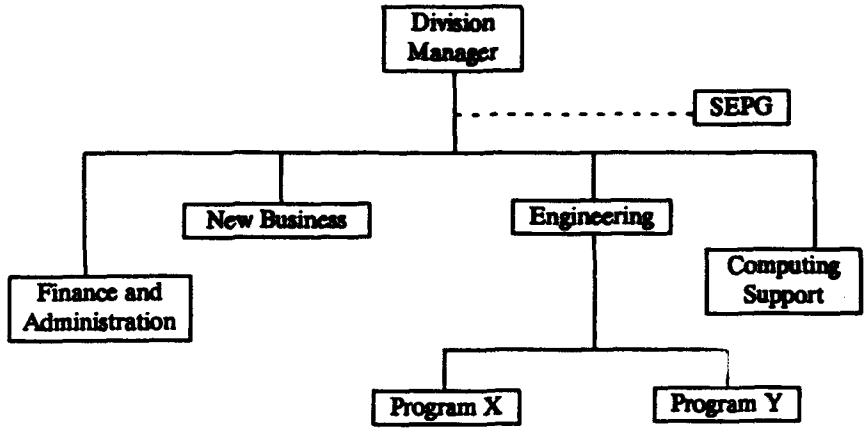


Figure 4-1. Organization Chart

- ***Subdomains (Specializations of the Prime Domain).*** This could be appropriate if you have several candidate subdomains for targeting the product family and want to determine which offers the most potential for reuse. An example from the aircraft domain is shown in Figure 4-2.

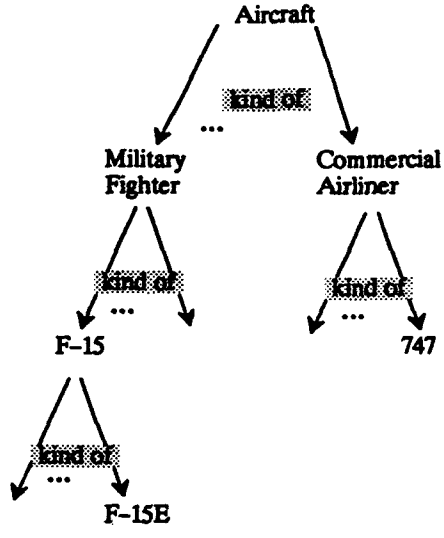


Figure 4-2. Domain Decomposition by Kind-of Relation

- ***Components of the Domain.*** This will provide more information on the opportunities and risks in each principle component of the domain. See Figure 4-3 for an example.

- ***Horizontal Domains.*** These are usually subsystems that occur in a wide variety of different products, but the subsystem itself can be a profitable product (e.g., Data Base Management Systems [DBMSs] or graphical user interfaces [GUIs]).

- ***Work Products.*** This could be appropriate if some work products have unusual reuse constraints. If principal variation is the target platform,

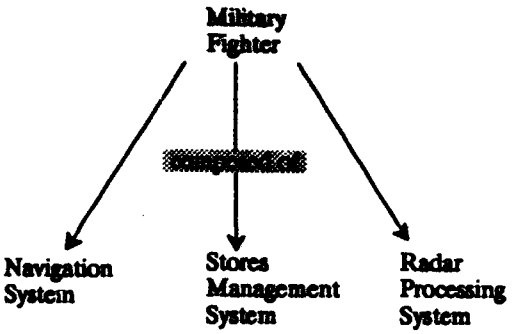


Figure 4-3. Domain Decomposition by Composed-of Relation

then code reuse may not be feasible, but opportunities for extensive reuse may exist for design and other work products.

4.3 ASSESS DOMAIN FACTORS

Purpose

Develop a set of findings characterizing the reuse potential for the selected domains.

Guidance

The team should next assess each of the five factors: market potential, existing assets, commonalities and variabilities, domain stability and maturity, and standardization. A factor consists of a list of attributes that a domain would likely exhibit or the preferred relationship of your organization to the domain if reuse is to be successful. In other words, the attributes define conditions that enhance the probability of successful reuse (see Appendix A for a complete definition of factors and attributes). As team members, you should base your evaluation on your understanding of the current situation and future trends.

To complete this task, you repeat the following three steps for each factor:

- Assess domain factor attributes.
- Tabulate the assessment.
- Develop preliminary findings.

You have the option of iterating these steps for each factor one at a time or completing each step for all of the factors before moving on to the next step. The first approach gives the facilitator the opportunity to explain each factor as you proceed through the assessment. The second approach gives the facilitator the opportunity to tabulate the assessment off-line, thus maximizing group discussion time. You should take this into consideration when scheduling the assessment.

Assess Factors

The facilitator begins this step with an explanation of the factor and its corresponding attributes to ensure that the assessors understand. The facilitator should explain the meaning of the attribute in the organization's context.

When the assessors understand the attributes, they then assess each attribute on the extent to which their domain exhibits the specified attribute on a scale of 1 to 5 (1–not exhibited, 5–fully exhibited).

At this point, you do not want the assessors to discuss the attributes (except to clarify their understanding of the attribute). The purpose is to prevent anyone from being unduly influenced by a peer. You can give the assessor the option of changing an assessment during the discussion.

Figure 4-4 illustrates how factors and attributes are presented to reviewers, and Table 4-1 shows how they would record their individual responses (see Appendix G for the recommended individual scoring form). In this case, the assessor indicated that the first attribute is mostly exhibited, indicating a potential opportunity. For the second attribute, the assessor indicated that it is not exhibited, indicating a constraint on the feasible reuse approaches.

Commonalities and Variabilities

Commonalities in Behavior, Structure, and Implementation

CR-1. A large portion of the products can be built with common components.

Comments: _____

_____

CR-2. A common architecture for the domain is feasible to enhance reuse.

Comments: _____

_____

Figure 4-4. Example of Factor Assessment

Table 4-1. Example of Individual Response

Attribute Identifier	Domain			
	A	B	C	D
	Exhibited: 1–not, 2–slightly, 3–moderately, 4–mostly, 5–fully			
Commonalities and Variabilities				
CR-1	4	5	4	
CR-2	1	4	4	

Tabulate the Assessment

After the assessors complete their assessment of the attributes, the facilitator collects and tabulates the assessment results. Table 4-2 shows one possible way to tabulate the assessment results. One row in the table is used for each domain assessed. The domains are identified in the second column. The values represent the average score for the entire team for each factor in a given domain. The attributes under each factor are identified in the left column (only one factor is shown in the sample table). The complete form is in Appendix G.

Table 4-2. Example of a Reuse Assessment Summary (One Factor)

Attribute Identifier	Domain	Assessor Scores				Total	Mean
		E	E	E	E		
Commonalities and Variabilities							
CR-1	A	4	5	4	5	18	4.5
	B	5	5	5	5	20	5.0
	C	5	5	4	4	18	4.5
	D						
CR-2	A	1	4	4	5	14	3.5
	B	4	3	5	4	16	4.0
	C	5	5	5	5	20	5.0
	D						

Develop Preliminary Findings

After the assessment results have been tabulated, the facilitator uses these results to stimulate group discussion. The facilitator or scribe then records the assessors' comments. Flip charts are very useful in this respect because they can be posted around the meeting room so everyone can see the discussion notes.

Specific actions that the facilitator may take to stimulate the discussion include:

- Look for potential strengths where the average attribute satisfaction is partially satisfied and more. Ask the assessors to describe why they believe these attributes are satisfied.
- Look for constraints where the average attribute is moderately satisfied or less. Ask the assessors to describe what needs to be accomplished to overcome these constraints.
- Look for wide variances in the individual assessors' scores. Ask the assessors to explain their views.

When all of the factors in the group have been discussed and everyone has had an opportunity to present their views, review the discussion notes and try to pick out the reuse opportunities and constraints. This can be readily accomplished by annotating the discussion notes.

4.4 DEVELOP ASSESSMENT FINDINGS

Purpose

Condense and prioritize the findings of the domain assessment.

Guidance

When the basic assessment is completed, you should identify the most important attributes to consider in building the reuse strategy. Build an initial list by asking each team member to select his ten top attributes. Then reduce the list through discussion and consensus.

Next, your team should group, merge, and prioritize results. Craft a set of statements to summarize the conclusions as illustrated in Figure 4-5.

Subdomain A

- Potential:

  Subdomain A is the largest of the three and exhibits the greatest potential for reuse.

- Contributing factors:
  - Technology has matured sufficiently to provide cost-effective systems.
  - COTS products are available for many functions (e.g., DBMS).
  - Some assets are available from current products.
  - Most customer requirements can be met with standard proprietary functions.
  - High commonality exists for the data processing portion of product line.
- Constraint:

  Customers may want to utilize existing hardware.

Figure 4-5. Example of a Summary of Reuse Potential

The summary should identify:

- Reuse opportunities
- Factors that support these opportunities
- Constraints that must be faced in building a reuse plan

A few cautionary notes on crafting summary statements:

- Avoid sweeping statements (e.g., no, not). Such statements are usually unjustified and of little help; e.g., this product will never require more than four megabytes of memory.
- Avoid veiled recommendations. The purpose of the summary is to identify opportunities and constraints, not to present solutions. An example will illustrate the problem:
  - Poor: There is a common requirement for sequential, unbounded components. (This implies a specific product and the language in which it is written.)
  - Better: Many of the required data structures have a common form, which can be implemented with standardized components.

- Avoid issues that have no possible recommendation. Such issues cannot be addressed in the reuse adoption strategy, so there is no point in listing them.

You should also present your assessment data in some form. The summary table illustrated in Table 4-2 is one possibility. However, you may want a less detailed presentation form such as the Kiviat diagram shown in Figure 4-6. One can quickly comprehend the relative strength of each factor for the domain assessed from this type of diagram. Such a diagram is easily prepared by averaging the mean scores for all the attributes under a given factor and plotting that value on the corresponding axis of the diagram. Drawing and shading the polygon completes the job.

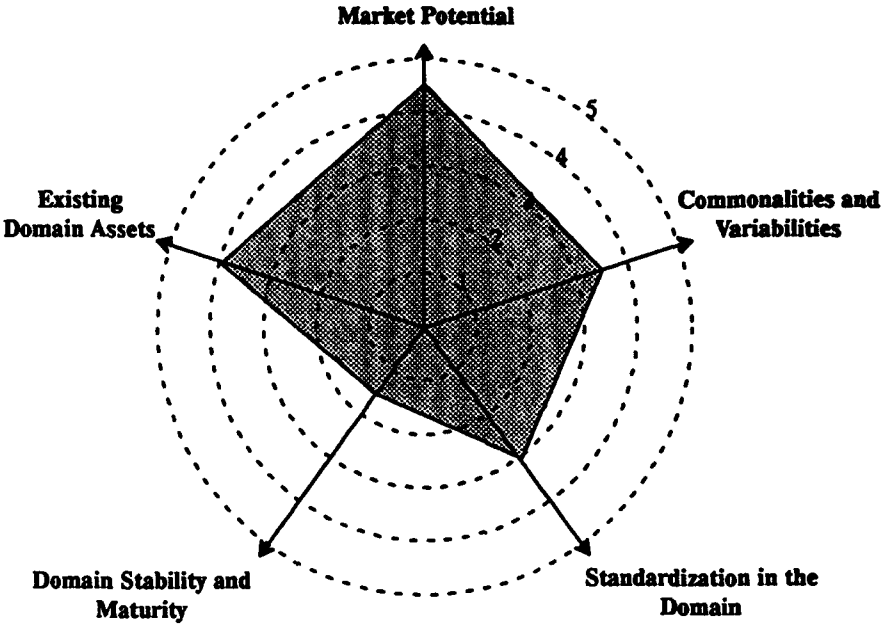


Figure 4-6. Example of a Domain Assessment Profile

4.5 DEVELOP SUPPORTING MATERIAL

Purpose

Document the factual basis for key findings of the domain assessment so that a rational and supportable case is available to management. This task also provides material for the product approach and other aspects of a reuse adoption strategy.

Guidance

The results of the domain assessment are used to identify the areas in which supporting material is needed. The process for developing the supporting material consists of five steps corresponding to the domain assessment factors as illustrated in Figure 4-7.

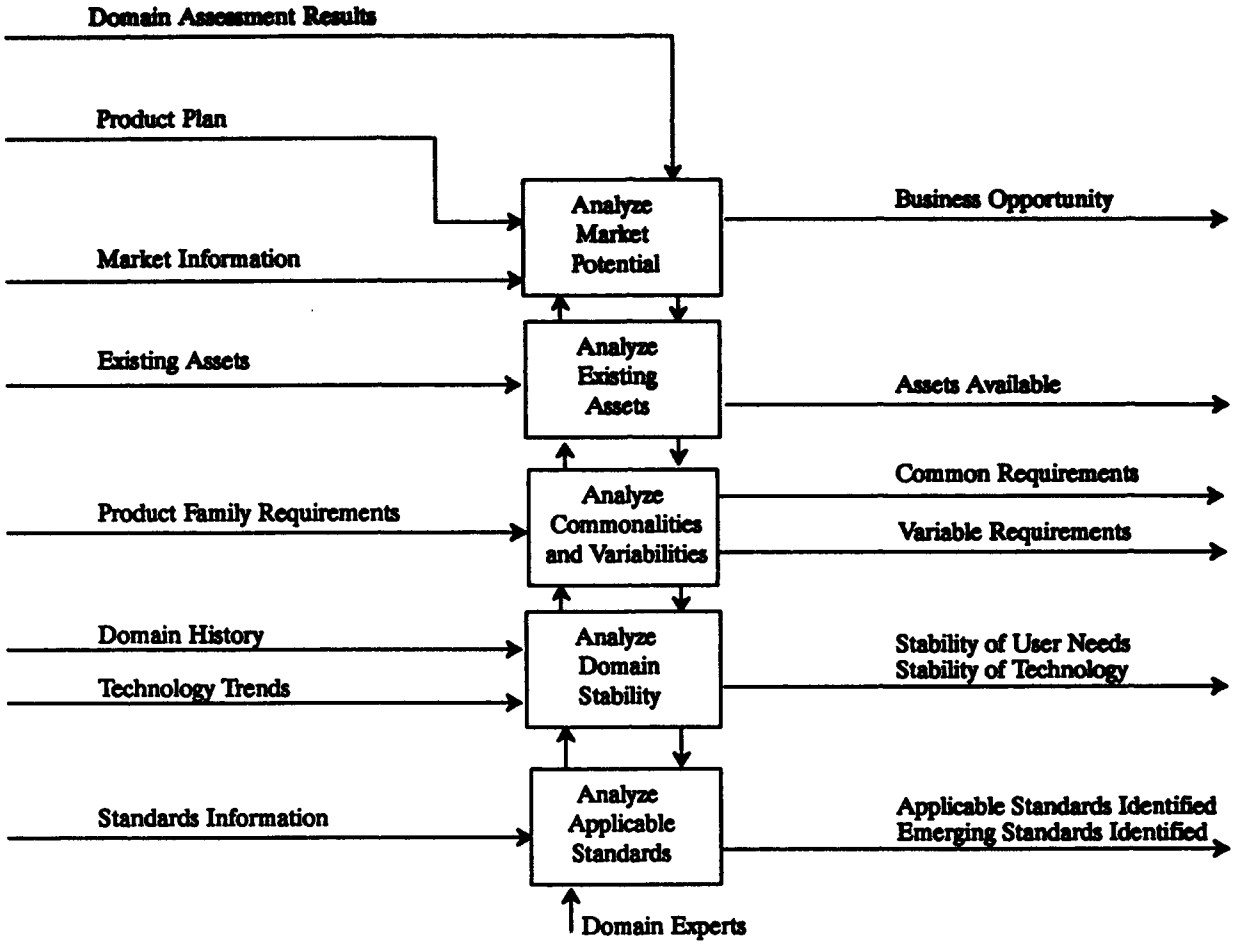


Figure 4-7. Process for Developing Supporting Material

The goal of this task is to create a concrete basis for management decisions. It is not intended to produce a system design, but some of the material should form the basis of the reuse adoption strategies proposed. In particular, they are intended to provide a definition for the product approach that will become the basis for the product design during the implementation stage.

You should use the results of the domain assessment to identify the key questions to be answered in each step and the depth of analysis required in each before assigning them to individual staff members. Otherwise, there will be a tendency to gather detailed information for each step regardless of what is needed to make decisions.

4.5.1 Analyze Market Potential

Purpose Estimate the magnitude of the business opportunity and identify the market factors that are important to designing the product line.

Guidance You should examine your market situation and product plans to develop a good definition of the business opportunity and the benefits to be gained by

applying reuse. You also need to examine the market situation to identify those factors that can be exploited in applying reuse to the product line.

Identify Product Situation

The product situation is described by the long-term plan for the product family and the point in that plan at which reuse is to be introduced. The plan includes products in development, anticipated enhancements to current products, and planned additions to the product family. Your analysis should include:

- ***Future Product Requirements and Expectations.*** These are the opportunities expected for future domain business acquisition and application of the products. Expected opportunities for development of products in the domain should be augmented by any additional business the organization expects to get because of competitive advantages provided by leveraging the domain assets.

- ***Products Currently in Development.*** New products should be assessed according to their stage in the product development life cycle. Domain development can be applied most effectively when products are in the concept definition phase. When the domain development is started early, it can have a payoff even if only a single system is built from the domain, especially on large programs that have evolving requirements. For product developments that are in detailed design or later phases, domain development can still provide benefits. Here, the benefit is realized if changes are expected in current and later stages of development caused by an evolving understanding of, for example, functional requirements, hardware implementations, operating concepts, system resource constraints, sizing and timing, schedule and cost limitations, and evolutionary growth.

- ***Existing Product Base in Modification.*** The objective of a reuse effort in this situation should be to improve the development process. Existing assets would almost certainly be used when a product or product line is being revised. However, the organization would not receive some of the benefits of reuse because it made no investment in asset creation. Because the product evolution process requires knowledge known only to a few engineers, it is probably not automated to any extent. With the addition of domain engineering, the process can be made explicit and standardized. This standardization should allow some parts of the process to be automated, thus providing for faster product update times.

You must estimate whether application of reuse technology reduces costs, reduces development times, improves quality, or in some other way improves your long-term plan. You also need to decide how reuse adoption can be worked into the long-term plan for maximum benefit and with minimum impact and risk.

Identify Market Situation

Consider the market situation. Expectations for future systems should be forecast in a three-step process as recommended by Jaworski et al. (1990):

- Estimate the total number of systems that could be sold, regardless of technical or cost constraints.
- Estimate the total number of systems that are expected to be sold by all producers, i.e., the percentage of the first estimate that customers will actually purchase.
- Estimate the number of systems that your organization will be able to sell, i.e., the expected penetration into the market identified by the second estimate.

You need to back up the above estimates with enough information to provide a solid basis for making business decisions. This could involve a substantial market research effort if large financial investments are at stake. However, you only need to gather enough information to convince yourself and management that the opportunity justifies the cost of the proposed effort.

You also need to consider the impact of the situation on your reuse plans. Reuse can be used to increase the number of versions produced or reduce the cost of a given number of versions. Either approach can be used to differentiate your product from the competition (thus possibly increasing market share) or make the product attractive to a larger number of potential buyers. Finally, a market that is growing implies a market that will demand new versions.

Your present customer base can be important. If that customer base is representative of the market in general, your assets and understanding of market needs are probably good. Otherwise, you should identify any significant gaps that may exist in your assets.

4.5.2 Analyze Existing Assets

Purpose

Identify existing assets (either on hand or available commercially) that can be applied to building the planned products.

Guidance

You need to review the assets available for reuse, the quality of those assets, and the capabilities of your organization in the domain. The less that needs to be generated during each production cycle, the higher the potential return. The use of existing assets can be a major factor.

Identify Availability of Existing Software

Documentation from previous product developments in the domain can be used as a basis for developing an understanding of the assets available. Requirements, design, source code, test documents for the software, and system and hardware specifications and any management documents, such as development and sustaining engineering plans, are all assets in domain development. Documents that the organization itself has developed and, to a lesser extent, those available from other sources should be considered.

Identify existing software that exhibits reuse characteristics (such as having already been reused) and adaptability. If your organization has been using specific work products (e.g., requirements, designs, code) in many applications, then it is likely that the same products can be used in future systems. These are obvious candidates and are just the beginning in identifying reusable assets.

You should identify and count these similar components so that a potential reuse benefit can be quantified. The number of identified components, their respective sizes (e.g., pages of requirements, lines of code), and the number of times they have been reused provides an indicator of value for assessing reuse potential.

Code analysis tools provide an alternative to manual inspection of potentially reusable components. The UNIX environment, for example, provides several tools to trace function calls and extract call structures. A component being called from several points in a system is said to exhibit high internal reuse. Components with a high internal reuse factor should also be identified and quantified to help you in assessing reuse potential.

Determine the Quality of Software Assets

Quality can have several meanings, ranging from well-documented and reliable software to software that provides customer satisfaction. In the case of reuse, quality is a measure of how well software meets its own specified requirements. The quality of your current software is, in most cases, proportional to its salvage potential, i.e., the money saved by reengineering the existing asset rather than building one from scratch.

High-quality software may not be reusable in other domains if the requirements are different. Similarly, it may not be reusable in the same domain if the requirements change. Thus, in addition to the quality of the asset, you should also assess its adaptability to current needs.

Identify Individuals and Organizational Expertise

The domain expertise embodied in your personnel and organization is necessary to create and apply the reusable assets in producing products. You need to identify these assets:

- Individuals with domain expertise are essential to creating useful assets. You should estimate cumulative staff effort worked in the domain and the number of available engineers.
- Organizational expertise is the organizational equivalent of individual expertise. The structure and policies of an organization can be a big asset if designed for reuse in the domain.
- The best domain developments are those for which an existing organizational charter closely maps to the scope of the domain to be developed. Your domain management effort will be much easier if ownership and control of the domain products do not cross organizational boundaries.

4.5.3 Analyze Commonalities and Variabilities

Purpose

Estimate what portion of the product features will be common across the family and the extent of the variable requirements.

Guidance

Common components of a product are the essence of reuse. You should look for common requirements, structure, and behavior across the anticipated product family. The existing product family is a good place to start to assess the extent of commonality. Look for commonality in designs and documentation as well as in code. If a common architecture is feasible, then exploiting commonality will probably be easier.

Variation is what differentiates individual members of the product family. You should examine the product family requirements for variation. You need to determine how much variation can be accommodated by each member of the family. You could produce new versions on demand or versions that each cover an enumerated set of variations. The former is driven by customer demand, while the latter is based on marketing and product design. Domain assessment can help in making the basic decisions on what approach to use in handling variations. Domain analysis can be used during development to provide the additional detail needed to design an effective approach for accommodating variation across the product family and future product evolution.

Judge whether the portion of each product that can be built from common elements is sufficiently high to justify basing the product family on reuse.

Identify Product Line History and Expectations

You should understand and make explicit the kind of software business you are currently practicing or planning to pursue. In most cases, organizations tend to believe they are involved in one-of-a-kind systems mainly because they have to comply with new requirements each time. It is not until they perform a careful analysis of their products that they identify many of the commonalities. These commonalities are the basis of a successful reuse program.

Your analysis should not be superficial. You should at least perform some initial cuts to domain analysis activities, such as the domain definition activity described in Software Productivity Consortium (1993b). Other methods for domain analysis, e.g., Feature-Oriented Domain Analysis (Kang et al. 1990), and Prieto-Diaz and Arango (1991), can also be used. Wartik and Prieto-Diaz (1992) provide a comparison of several domain analysis approaches.

The effort invested in this task should be just enough to determine whether significant differences or advantageous similarities exist among applications. One man-month or a team working for a week should be sufficient for a commonality analysis assuming that detailed and quality information is available or that experts are available. To reduce uncertainty in your analysis, you should have at least one expert participating in the analysis and have the results reviewed by additional experts.

Assess Behavioral, Structural, and Implementation Aspects

You should assess variability and commonality from the following viewpoints:

- ***Behavioral Requirements***. Assess the commonality of the behavioral requirements (i.e., on the end user's view of the system) and the relative priorities of the requirements.

- ***Software Structure***. Assess the commonality permitted in the implementing software architecture; i.e., how much do the variabilities force differences in the software architecture?

- ***Supporting Hardware and Software Implementation***. Assess the level of dependency of domain components on the supporting hardware and software components on which the application software is to be implemented.

Estimate Commonalities and Variabilities From Existing Systems

One way to estimate commonalities and variabilities is by comparing a number of existing systems. This comparison should be made at the design or requirements levels because code similarities are rare. Also, legacy systems usually have been built based on a given hardware architecture; i.e., their designs were hardware driven. In determining variability, you should look for essential features, not differences, in implementation details. Essential features are parts or characteristics of individual systems that must be present.

A suggested rule of thumb to determine if variations are large or small in a domain consists of the following steps. First, identify the essential features of existing applications. Next, separate the essential features that are recurrent through several applications from those that are unique, and compute their percentages with respect to the total essential features identified. A larger percentage (more than 50%) of invariant features indicates small implementation variations.

The resulting percentage is a preliminary indicator to help you assess reuse potential. More detailed figures would be obtained during domain analysis.

Estimate Viability of Commonalities and Variabilities

To estimate viability of the commonalities and variabilities in applications, you must analyze how different the systems your organization has built are from what you expect to build in the future. Reuse is viable when the allowed variability among applications is such that the cost of developing reusable components plus the cost of reusing them in new instances is less than developing systems from scratch for the expected number of applications.

Develop a list of features for the planned system family. Then evaluate its viability by answering the following questions:

- Do the commonalities correspond to the essential needs of the target customers? Are they really interested in systems that do everything implied by these commonalities, or may the scope be reduced? Conversely, are required features missing?

- Do the variabilities represent the real issues that determine whether a system addresses the individual needs of the target customers? Are there inconsequential variations that you can constrain without losing business? Are there important differences in the needs of target customers that are not captured in the variabilities defined?

- Are the commonalities and variabilities implementable? Can your organization build such systems? Does your organization have a good understanding of the problems such systems are intended to solve compared with your competition?

4.5.4 Analyze Domain Stability and Maturity

Purpose

Identify the features of the system that will likely be revised over time and estimate how soon those changes will be required.

Guidance

Lack of domain stability is both a problem and an opportunity. Rapid changes in customer needs or underlying technology for a product would appear to make reuse ineffective. However, where components can be identified that are not likely to change (i.e., be common across many future versions), reuse can reduce the cost and time required to produce each of the many required versions. On the other hand, complete stability would eliminate the need for new system versions. Stability is closely related to the maturity of the domain. New domains have notoriously unstable requirements. Beyond domain maturity itself, there are two kinds of stability that must be examined: stability of user needs and stability of the technology on which the system is based.

Assess Domain Maturity

You need to understand the maturity of the domain to define an appropriate reuse strategy. As a domain matures, more features become stable or standardized. Four stages in the life of a domain can be identified:

- ***Conception.*** The first products are introduced for the domain.

- ***Elaboration.*** The number of products and producers grows. Variety increases as the needs of the users are explored.

- ***Expansion.*** All the developers that want to be in the domain rush their products to market. Products and users become numerous.

- ***Consolidation.*** Unsuccessful products and producers begin to disappear from the market. The domain becomes well defined. Further diversification is much less likely.

Reuse can be applied to speed up the development of new releases and to reduce the cost of additional versions. The first option is most useful during the early phases of the domain life cycle when keeping ahead of the competition is essential, while the second option is most useful in the later phases when meeting the needs of specialized users is important.

It is important to assess where you are with respect to a domain life cycle and how long the life cycle is expected to be. Being in the early stages of domains with short life cycles, for example, may be more advantageous than being in the last stages of large and complex domains that feature long life cycles.

There are several factors that affect domain life cycle. If the domain is complex or broad in scope or if domain expertise is in short supply, the life cycle will tend to be long.

Assess Stability of User Needs

The stability of user needs in your domain is a major factor in planning for the future variations of your product. There are several suggested ways to assess change in customer needs: (1) look at changes in past products over time, (2) project the customer's future needs as far as possible, and (3) project future demands of the customer's environment.

The first assessment can be accomplished by examining the specifications of past members of the product line. You should identify the rate of change in your customer's needs for products in your business area. One way to do this is to review past change requests and Requests for Proposals (RFPs). Rapidly changing customer needs make it difficult to predict what assets will be reused or should be developed for reuse. In that case, a large investment in reuse may not be justifiable.

The second assessment may be accomplished by obtaining information from users of the current product. Try to identify possible future changes that you expect 6 months, 1 year, or even 3 years from the present. You can then consider the needs to be stable for the time period you feel you can predict confidently.

The third assessment can be handled like the second only if the customer anticipates changes in his environment. Otherwise, you should look at the environments in which your customer operates to determine what demands are placed on him that might translate into new needs. If the customer's time constraints are shorter than your product development cycle, you should look for the sources of change in the customer environment (e.g., government regulators).

Summarize your results by listing the requirements of the system that are expected to change and the expected pace of that change.

Assess Risks of Technology Evolution

Technological change is the other source of instability. You should assess the rate of change in the underlying technology for the domain and the sensitivity of the assets to those technologies. If you expect much change, you need to design your assets to isolate them as much as possible from the anticipated changes. You can also keep many of your assets abstract enough to be unaffected by the changes. This is relatively easy for some products, such as designs and documentation.

In assessing reuse potential, a rapidly changing technology, at a minimum, will require more investment in domain analysis, and it may possibly prevent

successful reuse. To reuse successfully within a rapidly changing technology, you will need to develop abstractions of the system where the technology-dependent implementation details can be isolated and interpreted as variabilities. So, dependency on a highly volatile technology calls for more up-front analysis work and less reliance on reusing from existing systems.

Thus, your first step is to identify the technology dependencies in your domain. Your domain can be classified as technology dependent if changes in technology substantially affect your software applications. Then you can estimate the rate of change and its predictability in the same manner you estimate changing needs: by looking at the past and predicting the future. Possible sources of information for identifying upcoming technology changes include trade journals, vendors, and conferences.

Again, summarize your results by listing the requirements or system features that are expected to change with the underlying technology and the expected pace of that change.

4.5.5 Analyze Applicable Standards

Purpose

Identify existing standards in the domain (especially those that could be the basis for reusable components), features of the product family that are or could be the subject of future standards, and features that can be profitably standardized within the product family.

Guidance

Standards do not generally define a product. Rather, they define certain individual features of a product (e.g., interfaces or recording formats). Software corresponding to a standard feature can be fixed as a reusable component and may be available commercially. Some features of a new domain will usually be covered by existing standards, and new standards will evolve for additional features. The relationship between standards and domain maturity is discussed below. As a domain matures, more of the product features are covered by standards. In planning your reuse strategy, you need to identify applicable existing standards, features that will likely become standards, and features that you can standardize to your advantage.

Identify the Type and Level of Standardization

Standardization in a domain may be characterized by type or level. The type of standardization refers to what is standardized; it could be an entire system or a piece of one. The level refers to the authority of the standard. This is closely related to the weight that the organization behind it carries. Domain standardization is related to maturity and formality of the domain.

There are at least three types of standard:

- ***System Architecture.*** Architectural standards tend to appear in domains that are mature, stable, and formalized. Compilers are an example of a mature, well-understood domain. A compiler is not usually developed from scratch. You either buy one or use compiler generation technology, such as Lex and YACC, to create one. Such domains are

apt to be highly commercialized, and generation tools may be available.

- ***Subsystems.*** Certain subsystems are expensive to build and appear in many domains. When the requirements across many domains are established, commercial products are usually introduced (e.g., operating systems, database managers, and graphical user interfaces). Such subsystems may be thought of as constituting a horizontal domain.

- ***Components and Interfaces.*** This type of category includes packages of mathematical routines, device drivers, and other component level software. These are usually available commercially as separate software packages (e.g., statistical packages) or bundled with associated hardware (e.g., printer drivers).

The scales for the level of standardization are:

- ***Official.*** These standards are established by consensus among many companies and published by national or international industry groups, such as the Institute of Electrical and Electronics Engineers, the American National Standards Institute, or ISO. An example is the networking standards that conform to the ISO/Open System Interconnection reference model. Users of an official standard construct systems by using standardized interfaces to the capabilities provided by the domain. The economic forces that create standardization at this level often support commercial development of related domain-specific software products. These define highly reusable components and may be available commercially.

- ***De Facto.*** These standards may be defined by a market leader or simply be practices recognized by the industry. These also define highly reusable components. You could attempt to establish a new standard if there is sufficient justification. Here, there are commonly understood ways of solving problems in the domain that are published in journals and textbooks. Elements of common solution techniques include agreement on key design drivers, ordering of design and implementation decisions, and the form of solutions. Solution techniques are generally agreed to and practiced in similar fashion by developers at different companies. As standardization increases at this level, commercial tools are often developed and become de facto standards in the domain.

- ***Organizational.*** These standards are defined by and for your organization. This is a good way to create your own reuse opportunity. They can be applied to most features of your product. However, if the feature is a candidate for official recognition, you need to consider whether you will press to get your standard accepted or redefine your product later to conform to whatever standard becomes accepted.

Based on circumstances, successful organizational standards are sometimes guarded by proprietary mechanisms and sometimes publicized to promote organizational status. As a result, development of automated environments may be done in-house (usually by tailoring commercial tools) or in collaboration with commercial vendors.

Assess Future Potential for Developing Standards

To what extent can definitions of software applications in your domain be adopted as industry standards? Although difficult to quantify, a sole indication that the domain has potential for standardization may be a sufficient justification to start a reuse program.

You should identify the degree to which your organization can influence the development of standards. If your company is, for example, a leader in developing software applications for a particular domain and that domain is becoming increasingly important for a large community of users, then your company has a good chance of influencing the definition of standards for that domain. If, on the other hand, your company or organization is involved in developing software systems for mature domains, it is more likely that your company will be following a set of standards rather than participating in defining such standards.

Reuse can also be applied in small or niche markets. Look for a domain that is small enough that your organization alone can standardize and exploit it effectively

In some cases, company-wide rather than industry-wide standardization is all you want. For example, Hewlett Packard wanted to standardize its instrument's command processing interfaces (Martin, Jakoway, and Ranganathan 1991) so that it could provide a consistent look and feel across a product family; this was viewed as enhancing market appeal and user satisfaction.

Identify Related Standardization Efforts

A new standard may cause obsolescence of existing software. Failure to recognize and react to the emergence of broad-based standards is a risk to the expected value of domain assets that you develop. For instance, some organizations focused on developing reusable components for management information systems before the emergence of database and networking standards. The emergence of the standards decreased the value of the assets that were developed because the components could not be used in more modern computing environments without extensive modification. By foreseeing other emerging standards, you can define your software assets to minimize this effect.

4.6 REPORT DOMAIN ASSESSMENT FINDINGS

Purpose

Document the assessment and supporting material for subsequent use and reference.

Guidance

When you have reached consensus on the findings, prepare and conduct a presentation on the assessment and findings for the sponsor and other key

members of the organization that either you or the sponsor believes should be in attendance. This is a first step in getting buy-in to the assessment results. You need this buy-in to effectively take any action on the assessment results.

Present Findings

The findings presentation should include:

- Assessment objectives
- Assessment scope: the domains assessed and the assessment team members
- Conduct of the assessment: a brief overview on the assessment procedure and your general impressions on its effectiveness
- Domain assessment findings
- Next steps: a suggested schedule for completion of the written report, reuse adoption strategy development, and action planning

One of the assessors should make the presentation (usually the reuse agent responsible for organizing the assessment). You should conduct a dry run of the presentation with the assessment team before the presentation to the sponsor. All members of the assessment team should attend the presentation to support the presenter and answer questions.

Document the Assessment

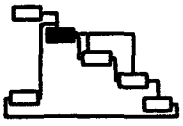
To complete the assessment, document the assessment and findings in a written report. This report provides a permanent record of the assessment. The assessment report should cover the same topics as the findings presentation but should include additional explanatory and supporting material. Appendix E provides an annotated outline for a combined domain and reuse capability assessment report.

To complete the report, you can divide the writing responsibility among the team members, then integrate the results. To ensure accuracy, you should give the team members a chance to review and comment on the draft. After any changes have been incorporated, you can distribute the report to the sponsor and any other parties from whom you will need support.

As an additional validation measure, you can also discuss the findings with other key members in the organization who could not participate in the assessment.

The supporting material developed should be included in summary form at least. Material for each step may be included with the assessment results for the corresponding factor. Indicate the relationship of the analysis and assessment results. Also include any conclusions reached from combining them.

5. REUSE CAPABILITY ASSESSMENT



The reuse capability assessment method supports the Assess Reuse Potential task of the Understand Context activity.

Purpose

Gain an understanding of an organization's process with respect to reuse sufficient for planning improvements—identifying process strengths and improvement opportunities.

Entrance Criterion

The sponsor has committed to performing a reuse capability assessment.

Tasks

- Organize the reuse capability assessment team.
- Identify the processes to assess.
- Assess the organization's process.
- Develop assessment findings.
- Report reuse capability assessment findings.

Inputs

- Organizational profile
- Current reuse situation, characterized by the organization's process, methods, tools, structure, and skills

Controls

Reuse adoption objectives

Outputs

- Reuse capability assessment findings
- Findings presentation
- Reuse capability assessment report

Mechanisms

- RCM
- Reuse capability assessment team

Exit Criterion

Reuse capability assessment findings have been reviewed and approved by the sponsor.

The reuse capability assessment is a collaborative, self-assessment technique that brings together your managers and engineers, producers, and consumers of reusable assets to assess your organization's

ability to practice software reuse. Your objectives are to gain a fuller understanding of the situation and to reach consensus on what needs to be done to exploit your reuse opportunities. Accomplishing this assessment will put you in a good position for developing realistic goals and effective strategies.

Begin the assessment by first identifying who should participate in the assessment, prepare the team, and make any required logistical arrangements. After you have organized the assessment team, conduct the assessment using the RCM (Appendix B). The RCM helps you focus the assessment on the issues critical to effective software reuse. The outcomes of the assessment are a shared understanding of your organization's process with respect to reuse and a set of findings: process strengths and improvement opportunities. Then report the findings to the sponsor and use the findings as a basis for establishing reuse adoption goals. Appendix G provides the worksheets used in the assessment and illustrated in this section.

5.1 ORGANIZE THE REUSE CAPABILITY ASSESSMENT TEAM

Purpose

Select and prepare individuals for conducting the reuse capability assessment.

Guidance

To ensure an effective and credible assessment, it is essential that you have the right people and that they are adequately prepared. The major steps you take to complete this task are selecting the assessment team members, scheduling the assessment, and training the assessment team members.

Select Team Members

There are three roles to be filled when organizing an assessment team: facilitator, assessor, and scribe.

- ***Facilitator.*** The facilitator is responsible for conducting the assessment in a timely, effective, and impartial manner. The facilitator ensures that the discussion focuses on the issues, everyone has a fair chance to speak their views, the discussion leads to results, and the assessment keeps to its schedule. The facilitator does not make judgments regarding the organization's reuse capability. The facilitator should be knowledgeable in the reuse capability assessment and the RCM and be experienced in leading a group.

- ***Assessor.*** The assessor is responsible for making judgments on the organization's process with respect to the critical success factors identified in the RCM, identifying the organization's strengths and improvement opportunities, and developing the findings. The assessor should be knowledgeable in the organization's process, policies, procedures, and standards.

- ***Scribe.*** The scribe is responsible for taking notes on the assessment discussions and results.

You will usually have from four to eight individuals participating as assessors; any more makes it difficult to achieve consensus, any less limits your representation and may cause people to question the validity of the results. You should try to select the assessment team so that:

- The team provides a good cross-representation of the processes you are assessing. You can, and are encouraged to, include a mix of management and technical staff.

- The choice of team members will facilitate buy-in to the assessment results from the remaining organization, i.e., you want to include opinion leaders.

The organizational profile should provide sufficient context to aid you in selecting team members. If it does not, you can identify the processes to be assessed in more detail as described in Section 5.2, then use that identification to select a team with a good cross-representation.

Normally, you have the same team members participate throughout the assessment. However, you have the option of having different members participate during the different assessment topics. A likely breakup is to have more technical staff participate when discussing the factors associated with application and asset development and more management staff participating when discussing the factors associated with management, process, and technology. In this case, you should still try to have a core group of at least three people who participate in the entire assessment. This approach enables a larger cross-representation and reduces the amount of time that any one person must commit to the assessment.

Selecting the assessment team members also includes that you get the team members to commit to conducting the assessment; getting this commitment usually depends on the schedule.

Schedule the Assessment

In scheduling the assessment you should allot a half day for discussion of each group of assessment factors (application development, asset development, management, and process and technology) plus a half day or more for developing the findings. You should also schedule 1 hour for presenting the assessment findings to the sponsor.

You have some flexibility in scheduling in that the half-day factor discussions need not be done contiguously—you could do one per week, for example. However, you need to weigh this flexibility against the possible negative effects of a protracted assessment, such as loss of continuity and momentum.

In addition to scheduling the team members' time, you need to schedule the room and any necessary audiovisual equipment.

Train the Assessment Team

To ensure that the assessment team is adequately prepared for the assessment, the team members should be instructed on the:

- Purpose of the reuse capability assessment

- Reuse capability assessment procedures

- Expected use of the assessment results
- RCM

All assessment team members should participate in this training. The training could be conducted in a classroom format (usually 2 hours) or via reading material.

5.2 IDENTIFY THE PROCESSES TO ASSESS

Purpose

Identify the processes that will be the subject of the assessment and review them with the assessment team to ensure that the team has a common understanding of the processes that they are assessing.

Guidance

Ideally, you should identify processes that:

- Correspond to the scope of your reuse program established in your reuse adoption objectives. For example, if you are implementing a division-wide reuse program, then you should assess your division-wide processes.
- Encompass management, producer, and consumer roles. In many organizations just beginning to practice reuse, the producer and consumer processes are often not distinguished from one another as in carryover reuse, in which the products of one development become the reusable assets for the next development.
- Include software development, maintenance, and acquisition processes if applicable.

However, there are cases when you might want to narrow the focus of the assessment, including:

- If you have no formally defined, organization-wide process, you could select a set of individual projects that are representative of your organization's process as a basis for the assessment.
- If the domain assessment results indicate areas of high reuse potential, you could focus the capability assessment on the processes corresponding to those areas to determine how you might best exploit those high-potential opportunities.
- If your organization does not have ownership of a process, you could focus on your interface to that process. This could be the case in corporate-wide reuse programs or in government acquisition agencies. In a corporation or acquisition agency, you may have limited or no control over the individual division's or contractor's processes. In these cases, you can focus on the processes you own and your ability to facilitate or influence the processes you do not own.

When identifying the processes to be assessed, you should try to be as specific as possible. You can elaborate on or trim the organizational profile as necessary to support this identification. The important thing is to make sure that everyone is assessing the same process.

5.3 ASSESS THE ORGANIZATION'S PROCESS

Purpose

Solicit discussion on the organization's process, focusing on the issues critical to an effective and efficient reuse practice to identify process strengths and potential improvement opportunities.

Guidance

The RCM identifies a set of factors critical to achieving effective and efficient reuse. The factors are divided into four groups: application development, asset development, management, and process and technology. The factors are defined in terms of one or more goals that describe what needs to be accomplished to address that factor (see Appendix B for a complete definition of the factors and goals).

The assessment team uses the critical success factors and corresponding goals to focus and stimulate the discussion. The findings then are derived from the discussion.

To complete this task, repeat the following three steps for each group of critical success factors:

- Assess critical success factor goals.
- Tabulate the assessment.
- Develop preliminary findings.

You have the option of iterating these steps for each critical success factor one at a time or completing each step for all of the factors before moving on to the next step. The first approach gives the facilitator the opportunity to explain each factor as you proceed through the assessment. The second approach gives the facilitator the opportunity to tabulate the assessment off-line, thus maximizing group discussion time. You should take this into consideration when scheduling the assessment.

Assess Critical Success Factor Goals

The facilitator begins this step with an explanation of the critical success factor and its corresponding goals to ensure that the assessors understand. The facilitator should explain the meaning of the goals in the organization's context; the organizational profile is very helpful in this respect.

When the assessors understand the goals, they then assess each goal on two items:

- The extent to which their organization meets the specified goal on a scale of 1 to 5 (1–not satisfied, 5–fully satisfied)

- The expected impact on the organization's reuse capability from fully satisfying the stated goal on a scale of 1 to 5 (1–no positive impact, 5–high positive impact)

At this point, you do not want the assessors to discuss the goals (except to clarify their understanding of the goal). The purpose is to prevent anyone from being unduly influenced by a peer. You can give the assessor the option of changing a goal assessment during the discussion.

Figure 5-1 provides an example of an assessment for the Asset Awareness and Accessibility factor. In this case, the assessor indicated that the first goal is mostly satisfied and has a high impact, indicating a potential strength. For the second goal, the assessor indicated that it is not satisfied and would have a moderately high impact, indicating a potential improvement opportunity.

Asset Awareness and Accessibility

AA-1. Developers are aware of, can find, and have access to any relevant reusable assets and external sources of assets.
Comments:

Not — Moderately Satisfied — Fully
☐ 1 ☐ 2 ☐ 3 ☑ 4 ☐ 5

No — Moderate Positive Impact — High
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☑ 5

AA-2. Developers are aware of and reuse assets that are specifically acquired or developed for their application.
Comments:

Not — Moderately Satisfied — Fully
☑ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

No — Moderate Positive Impact — High
☐ 1 ☐ 2 ☐ 3 ☑ 4 ☐ 5

Figure 5-1. Example of a Critical Success Factor Assessment

Tabulate the Assessment

After the assessors complete their assessment of the critical success factor goals, the facilitator collects and tabulates the assessment results. Tables 5-1 and 5-2 illustrate two possible forms for tabulating the assessment results.

Table 5-1. Example of Assessment Tabulation

Goal Identifier	Assessor Scores					Total	Mean
	S/I	S/I	S/I	S/I	S/I		
Application Development Factor Goals							
AA-1	4/5	3/4	4/4	5/5	3/5	19/23	4/5
AA-2	1/4	1/4	1/3	3/4	1/4	7/19	1/4
AI-1	3/3	4/5	2/4	2/3	3/3	14/18	3/4
AI-2	2/4	1/4	2/3	1/4	2/3	8/18	2/4

Table 5-2. Example of a Reuse Capability Profile

Critical Success Factor		Opportunistic	Integrated	Leveraged	Anticipating
Application Development	AA	1■	2□		
	AI	1◪	2◪	3□	4 -◪-
	AE	1◪ 2◪			
	AN		1◪		
Asset Development	NI	1◪ 5◪	2◪	3 -◪-	4 -◪-
	AD	1◪	2◪		
	NS			1◪ 2□	
	CV	1◪	2◪		
	AV				1◪ 2◪
	AR		1◪		
	AQ	1■	2◪ 3◪ 4◪	5◪	
Management	OC	1◪ 2◪ 3◪	4□		
	PD	1◪	3□	2□	4□
	CP	1□	2◪	3□	
	LC	1 -◪-			2◪
	IC		1◪		
Process and Technology	PI	1◪	2◪ 3◪		4◪
	MS	1◪	2◪		3◪
	CI			1◪ 2◪	
	TR	1◪		2◪ 3◪	
	TS	1■	2◪	3◪	4◪
	TI			1◪	2◪

Key:
□ Goal not satisfied
◪ Goal slightly satisfied
◪ Goal moderately satisfied
■ Goal mostly satisfied
■ Goal fully satisfied

-◪- Impact less than moderate

Numbers represent goal identifiers

Table 5-1 shows how each of five individuals assessed the satisfaction (S) and impact (I) of each goal; this is useful in bringing out contrasting opinions. The table also provides the average satisfaction and impact for each goal.

The capability profile in Table 5-2 shows how the goals map to the stages of the RCM Implementation Model. Each row corresponds to a critical success factor. Each numbered box corresponds to the critical success factor goals. The column of the box indicates which stage the goal is in. The boxes are shaded to indicate the extent to which the goals are satisfied. This table

provides a good visual picture of the goal satisfaction and helps the group focus on the most important issues: the unsatisfied goals of the early stages.

If the results of the domain assessment indicate a limited potential for reuse, you can further focus your discussion to the goals in the opportunistic and integrated stages because pursuing the higher stage goals may not be justified.

Develop Preliminary Findings

After the assessment results have been tabulated, the facilitator uses these results to stimulate group discussion. The facilitator or scribe then records the assessors' comments. Flip charts are very useful in this respect because they can be posted around the meeting room so everyone can see the discussion notes.

Specific actions that the facilitator may take to stimulate the discussion include:

- Focus the discussion on the goals that have an average impact of moderate and more.
- Look for potential strengths where the average goal satisfaction is partially satisfied and more. Ask the assessors to describe how they are satisfying these goals.
- Focus the discussion of improvement opportunities on unsatisfied goals in the earlier stages of the capability profile.
- Look for potential improvement opportunities where the average goal satisfaction is partially satisfied and less. Ask the assessors to describe what needs to be accomplished to satisfy these goals.
- Look for wide variances in the individual assessors' scores. Ask the assessors to explain their views.

When all of the factors in the group have been discussed and everyone has had an opportunity to present their views, review the discussion notes and try to pick out the key strengths and improvement opportunities. This can be readily accomplished by annotating the discussion notes.

5.4 DEVELOP ASSESSMENT FINDINGS

Purpose

Achieve consensus on the organization's key strengths and improvement opportunities.

Guidance

A finding is the assessment team's view of the organization's key process issues with respect to effectively and efficiently practicing reuse. A finding consists of a group of related strengths or improvement opportunities and the associated benefits expected from implementing the opportunities. Figure 5-2 provides an example of a finding.

Limited Accessibility

- **Strength:**

  Many developers have and use their own personal libraries of software that they had developed.

- **Improvement opportunities:**

  - Increase accessibility and awareness of division software assets.
  - Provide mechanisms to ensure that developers look for, evaluate, and verify reusable assets.

- **Benefits:**

  - Increased likelihood that available assets will be reused
  - Reduced cost and schedule

Figure 5-2. Example of a Finding

Consensus on a finding means that there is a general agreement among the team members and no strong minority dissention; i.e., most of the members support the finding and those that oppose it have had a fair chance to state their views and influence the consensus but are prepared to support the finding.

Specific actions that the team can take to achieve consensus on findings include:

- Group, merge, and prioritize the preliminary findings into categories. The critical success factors constitute a possible grouping.
- Limit the number of findings to five to nine. This will allow the organization to better focus its actions.
- If there is strong disagreement, try to find a third option that encompasses both views.
- Set aside issues that you are having difficulty in resolving and return to them later.
- Avoid sweeping statements (e.g., no, never, not, always); there will usually be an exception.
- Avoid issues that have no solutions.
- Avoid veiled recommendations. At this point, you want to get buy-in to the opportunity; including a recommendation in the opportunity runs the risk of having the opportunity rejected because someone does not like the recommendation.

When you identify benefits for an opportunity, you should identify benefits that address the sponsor's motivations, e.g., productivity, quality, image, risk, cost, competitiveness, etc.

5.5 REPORT REUSE CAPABILITY ASSESSMENT FINDINGS

Purpose

Ensure that the sponsor and other members of the organization understand their key process strengths and opportunities with respect to reuse. Provide a written record of the assessment for future reference and use.

Guidance

When you have reached consensus on your findings, prepare and conduct a presentation on the assessment and findings for your sponsor and other key members of the organization that either you or your sponsor believe should be in attendance. This is a first step in getting buy-in to your assessment results. You will need this buy-in to effectively take any action on the assessment results.

Present Findings

Your findings presentation should include:

- Assessment objectives
- Assessment scope: the organizations assessed and the assessment team members
- Conduct of the assessment: a brief overview on the assessment procedure and your general impressions on its effectiveness
- Reuse capability assessment findings
- Next steps: a suggested schedule for completion of the written report, reuse adoption strategy development, and action planning

One of the assessors should make the presentation (usually the reuse agent responsible for organizing the assessment). You should conduct a dry run of the presentation with the assessment team before the presentation to the sponsor. All members of the assessment team should attend the presentation to support the presenter and answer questions.

Document the Assessment

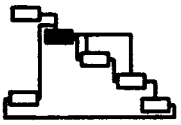
To complete the assessment, document the assessment and findings in a written report. This report provides a permanent record of the assessment. The assessment report should cover the same topics as the findings presentation but should include additional explanatory and supporting material. Appendix E provides an annotated outline for a combined domain and reuse capability assessment report.

To complete the report, you can divide the writing responsibility among the team members, then integrate the results. To ensure accuracy, you should give the team members a chance to review and comment on the draft. After any changes have been incorporated, you can distribute the report to the sponsor and any other parties from whom you will need support.

As an additional validation measure, you can also discuss the findings with other key members in the organization who could not participate in the assessment.

This page intentionally left blank.

6. REUSE ADOPTION STRATEGY DEVELOPMENT



The reuse adoption strategy development method supports the Identify Alternative Reuse Adoption Strategies task of the Understand Context activity.

Purpose Develop a strategy to meet the established reuse adoption goals and objectives.

Entrance Criterion Reuse adoption goals have been established.

Tasks

- Develop product approach.
- Develop business model.
- Develop process approach.
- Develop organizational approach.
- Develop environment approach.
- Develop transition approach.

Inputs

- Organizational profile
- Supporting material

Controls

- Reuse adoption objectives
- Reuse adoption goals
- Constraints

Output Reuse adoption strategy

Mechanism Reuse agent

Exit Criterion The reuse adoption strategy addresses all strategy components.

Figure 6-1 shows a general flow in the development of the adoption strategy components. Note that there will be interaction and iteration in development of the approaches. However, the three primary concerns will be the product approach, business model, and process approach. The environment approach depends mostly on the process. The organizational approach should be developed to fit the three primary concerns. Finally, the transition approach should come last and determine any

intermediate steps in implementing the other approaches. You should expect to develop all of these approaches more or less in parallel, with initial emphasis on the areas that most closely relate to the reuse adoption objectives, goals, and constraints.

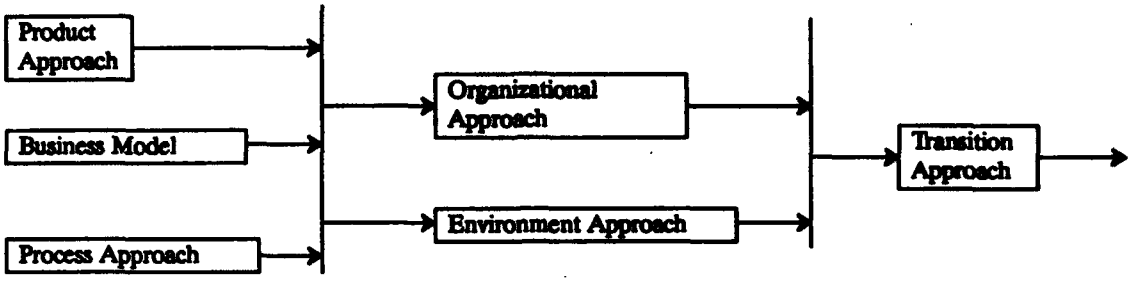


Figure 6-1. Flow in Development of the Alternative Reuse Adoption Strategies

6.1 DEVELOP PRODUCT APPROACH

Purpose

Identify the reusable assets (both existing and planned) that will be the basis of the planned products, and define a strategy for accommodating the variations expected across the product family.

Guidance

Start with the inventory of existing assets and the market potential analysis that was identified during the Assess Reuse Potential task. You will need to modify these broadly scoped analyses to the specific scope that your sponsor is interested in.

If existing products are to be the source of assets, you must consider the organization's ownership rights (see Appendix D and Section 6.2).

Identify Reusable Assets

Prepare a description of assets that will form the basis of the reuse program. You should begin with the common requirements identified during requirements analysis (Section 4.5.3). Identify the reusable assets that could meet these requirements. Determine how each asset can be obtained. Your primary options are:

- Extract from existing assets.
- Reengineer existing assets.
- Acquire assets from outside source.
- Build or subcontract the needed assets.

Standards also need to be examined here. The common requirements that correspond to existing or emerging standards can be satisfied with assets acquired by any of the methods described above. Assets for emerging standards may go through several iterations but are included here because they should soon become part of the stable asset base. Organizational standards are considered in the next step.

You must identify sources for all the assets you need. There are three possibilities: assets that can be extracted from existing software, assets that can be adapted from available material, and outside vendors. Anything that is not available from one of these must be developed new. You are in a position to estimate the cost of obtaining all the needed assets after reviewing existing software.

Reusable assets should include documentation of specifications, design assumptions, design decisions, and usage instructions. Existing material and staff expertise are both important in adapting or building reusable assets. If you decide to build the assets, it is advisable to train the developers in designing and documenting software for reuse.

You should also consider the evolution of the reusable assets in your product approach. It is often not feasible or desirable to build up a complete set of reusable assets before the development of the first product. A more prudent approach is to build up your set of reusable assets over the course of several product developments. This approach spreads out your reuse investment and mitigates risk.

Define Strategy for Product Family

Defining a strategy for producing a product family is something like designing a production line for automobiles. The aim is to define a process for producing all the variations that customers want while keeping production cost low.

One key is to find efficient ways to accommodate variation. There are at least three ways to do this: build a set of modules to cover the range of requirements, build an adaptable module to meet a range of requirements, or define a standard interface for which modules can be built as needed. A mix of these strategies can be used to cover the complete set of variable requirements. The strategy must not only accommodate known variations but also projected evolution in user needs and supporting technologies.

Variabilities in current requirements were identified during requirements analysis (Section 4.5.3) and future variabilities were predicted during the analysis of domain stability (Section 4.5.4). The strategy for your product family needs to cost effectively address as many of these variabilities as possible. Another factor that was obtained in analysis of domain stability is the probable duration of the current situation, i.e., the time before the product family itself will have to be redesigned.

Some requirements may be addressed by several competing standards (e.g., IBM, Apple, and Amiga disk formats). This forms an enumerated set and would most likely be handled by acquiring an asset corresponding to each and selecting the appropriate one in building a given version.

A product architecture can be the most powerful unifying concept for a product family and point the way to a production process. For example, where variability can be fully enumerated, a domain engineering process can be applied and the production process automated (Software Productivity

Consortium 1993b). In either case, organizational standards should be considered here. They can give the product family a consistent look and feel or standardize internal features of the system to reduce the cost of developing future versions.

The product approach portion of a reuse adoption strategy is basically a list of key ideas derived from the above analysis. An example of a product approach is shown in Figure 6-2.

- Reengineer assets from existing products.
- Acquire a commercial DBMS.
- Acquire a development tool that will generate versions of the product for all target platforms.
- Acquire assets to support all targeted file formats for color graphic images.
- Develop assets to support a standard look and feel for the product line.

Figure 6-2. Example of a Product Approach

6.2 DEVELOP BUSINESS MODEL

Purpose

Identify an approach to contracting that provides for the profitable management, development, use, and maintenance of reusable assets. Specifically, you should identify the following:

- Who pays to build reusable assets?
- How is the investment in reusable assets recovered?
- How are the costs of reuse accounted for?
- What ownership rights do the customer, developer, and reuser have to the assets?
- How is the price to the reuser or end customer determined?
- Who has responsibility for maintaining and upgrading the reusable assets over their life cycle?
- Who assumes liability for reused assets?

Guidance

Developing a business model is probably one of the most difficult tasks in establishing a reuse program. Currently, legal guidelines, including ownership, liability, patent law, copyright law, licensing agreements, and royalties, are not mature. The net result is that software reuse, except within some narrowly defined bands, cannot be practiced in standard legal formats that average software engineering organizations understand and can repeat for many customers. Except in these narrowly defined bands, you should get the support of legal counsel to negotiate the issues with your customer.

The earlier in the system development life cycle that you begin to establish the business model, the more likely you are to be successful. Further, if you begin a large development contract after the RFP cycle for the engineering and manufacturing development phase, it is unlikely that systematic reuse will happen on that contract.

Although most of the legal and contracting community is not prepared to resolve the issues, there have been a number of recent studies that identified the problems and recommended solutions. Appendix D is a summary of the results of those studies. Additionally, there are ongoing government and industry working groups attempting to create and implement solutions.

CARDS has developed a handbook that specifically addresses DoD software acquisition issues (CARDS 1992). This handbook is aimed primarily toward government program managers and their support personnel. The goal of the handbook is to encourage software reuse during the acquisition and maintenance portions of the life-cycle process. The handbook is also appropriate for contractor use because it identifies business model strategies, recommends customer evaluation criteria, and recommends RFP wording.

You should take the following steps in developing a business model:

- Understand the plans and constraints of your customers.
- Select a funding mode.
- Ensure a commitment to domain management.

Understand the Plans and Constraints of Your Customers

Your customer's needs constrain the acceptable business models. For instance, if the customer is buying a product that is expected to have a long operational life with many modifications and upgrades, the customer probably wants to procure unlimited rights to the software to ensure maintainability.

On the other hand, if the software is not expected to be maintained and is an off-the-shelf product, then the customer may be willing to buy the software with restricted rights.

Select a Funding Mode

You must select a funding mode before you can analyze your potential payoffs and risks. There are basic modes for which the rules are reasonably clear:

- You develop proprietary reusable assets entirely on your own funding and then sell the assets unaltered to the DoD with restricted rights. This is the typical funding mode for off-the-shelf software such as database packages. It requires that you take all the investment risk, but, if your customer is willing to procure the copies under a restricted rights agreement, you can sell each copy of the software that you deliver.
- You develop reusable assets entirely on the customer's funding. The customer owns the assets at completion of the development effort.

Thus, the assets can be used by third parties. In this model, the customer assumes the risk for the reusability of the assets that you develop, but your return is limited to the single contract plus any advantage that you have on future use of the asset resulting from your familiarity with the asset.

Between these two extremes, there are many variabilities that provide for sharing the ownership and risk between the customer and developer. Unfortunately, these fall into the range where the open legal issues exist and must be worked out on a case-by-case basis. In these situations, retention of ownership rights by the software developer depends on an accounting and configuration management system that keeps track of the assets according to the agreements made.

Consider, as an example, a situation in which the developer can develop assets under a contract to produce an initial system; i.e., the customer is buying a custom system with the understanding that the developer has the right to sell the product or derivatives of it to others. This situation is diagrammed in Figure 6-3.

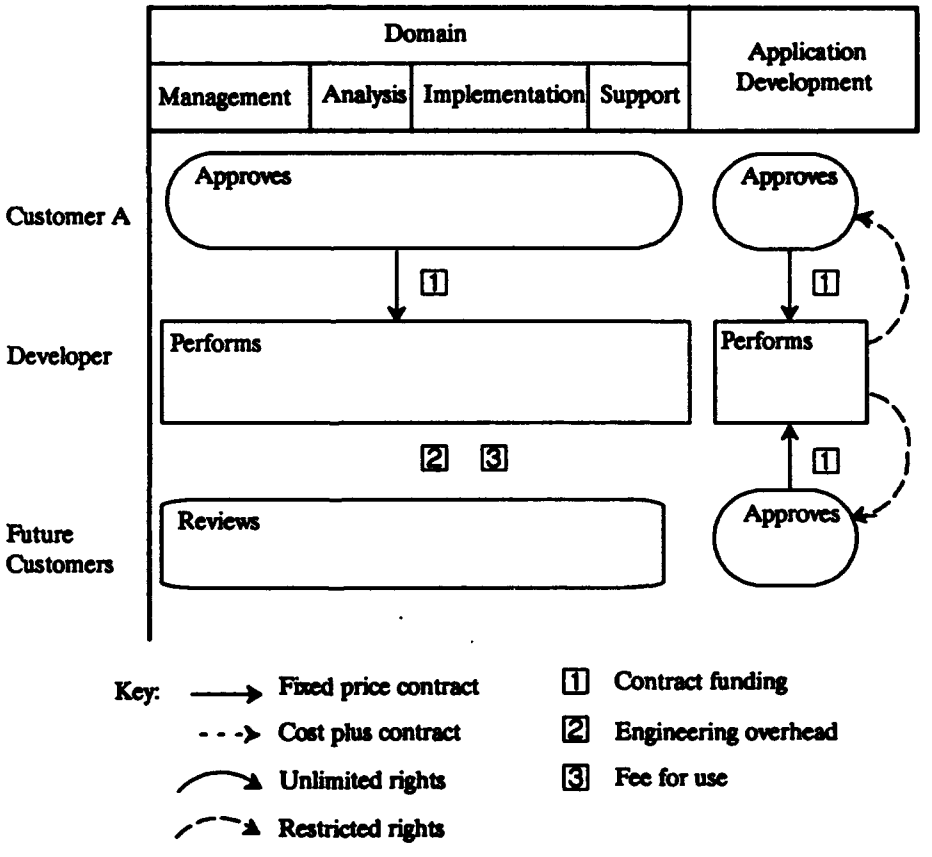


Figure 6-3. Example of a Business Model

In the example, the developer initially performs the domain activities under a fixed price contract to customer A. The developer also builds the application for customer A under a fixed priced contract, and customer A has restricted

rights to the software. The developer continues to build up the domain assets using engineering overhead funds and fees received for use of the assets. The developer can then build applications for future customers based on its domain assets under fixed price contracts, which include fees for use of the domain assets.

Ensure a Commitment to Domain Management

As a final step in the development of a business model, you must ensure that your model allows for maintenance and continuity of the reusable asset as a business area asset; i.e., the assets must be controlled by a party (a "domain manager") whose economic interest is in the asset's usefulness across all the application projects, rather than a single one. This domain manager role can be in conflict with the program or project management role on a single application. The domain manager must be able to trade off the costs and benefits of proposed changes to the domain assets (or exceptions from their use) that individual application projects propose. Alternatively, you might use a committee to manage the domain assets.

6.3 DEVELOP PROCESS APPROACH

Purpose

Define processes, methods, and product standards for reuse-driven software development. To complete the process definition, you must identify:

- The types of software assets to be produced and reused (e.g., requirements, designs, code, test cases)
- How the assets are documented for the customer (e.g., DOD-STD-2167A [DoD 1988]), supplier, and internal users
- How the software development effort integrates with the overall system development

Guidance

You should tailor your development approach to your reuse capabilities as defined during the reuse capability assessment, building on the strengths of your existing processes, not a whole replacement. There are several aids that you can use in defining a reuse-driven software development process for your organization. Beginning at the most abstract level, they are:

- The *DoD Software Reuse Initiative: Vision and Strategy* (DoD 1992c) outlines a long-term vision for a "process-driven, domain-specific, architecture-centric, library-based way of constructing software" and identifies some of the cornerstone products that support the vision. Each of the military services has been tasked with developing an implementation plan for the document.
- The *STARS Reuse Concepts—Conceptual Framework for Reuse Processes* (STARS 1992b) provides a framework for classifying and describing reuse processes. It provides a hierarchy of process activities that can be used as a framework for expressing your development process.

- The *Reuse-Driven Software Processes Guidebook* (Software Productivity Consortium 1993b) identifies a family of reuse-driven software processes (Synthesis) that produce the types of products defined in DoD (1992c).

Software Productivity Consortium (1993b) further defines two specific members of the Synthesis family that have been tailored to the opportunistic and leveraged stages of the RCM. The opportunistic stage is a recommended version of process that is easiest to integrate into today's common software development practice. The leveraged stage process model is a more aggressive approach to reuse that more directly approaches the intent of DoD (1992c).

The representation of processes in Software Productivity Consortium (1993b) is still abstract enough to allow some latitude in the definition of the software development methods and representations for requirements, architecture, designs, and code and work products. For example, the process defines specific steps that you take in creating a software architecture and defines why and when the architecture is used, but it does not specify any particular notation or format for the architecture. Similar latitude is provided to the practicing organization in the selection of requirements and design methods.

Figure 6-4 shows a Synthesis process. It consists of two distinct subprocesses: Application Engineering and Domain Engineering. These processes take on a number of different forms depending on the reuse capability stage. For example, the processes defined for the opportunistic stage are shown in Figures 6-5 and 6-6, respectively. Note that the reusable assets are contained within an entity called the domain implementation.

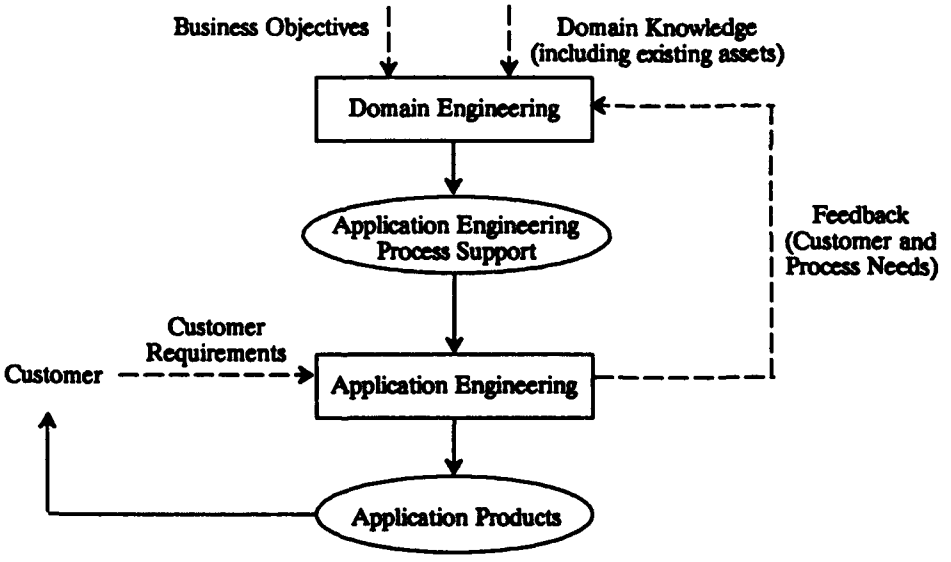


Figure 6-4. A Synthesis Process

The typical activities involved in the software development process with reuse are identified using this example. Activities like those in the waterfall model

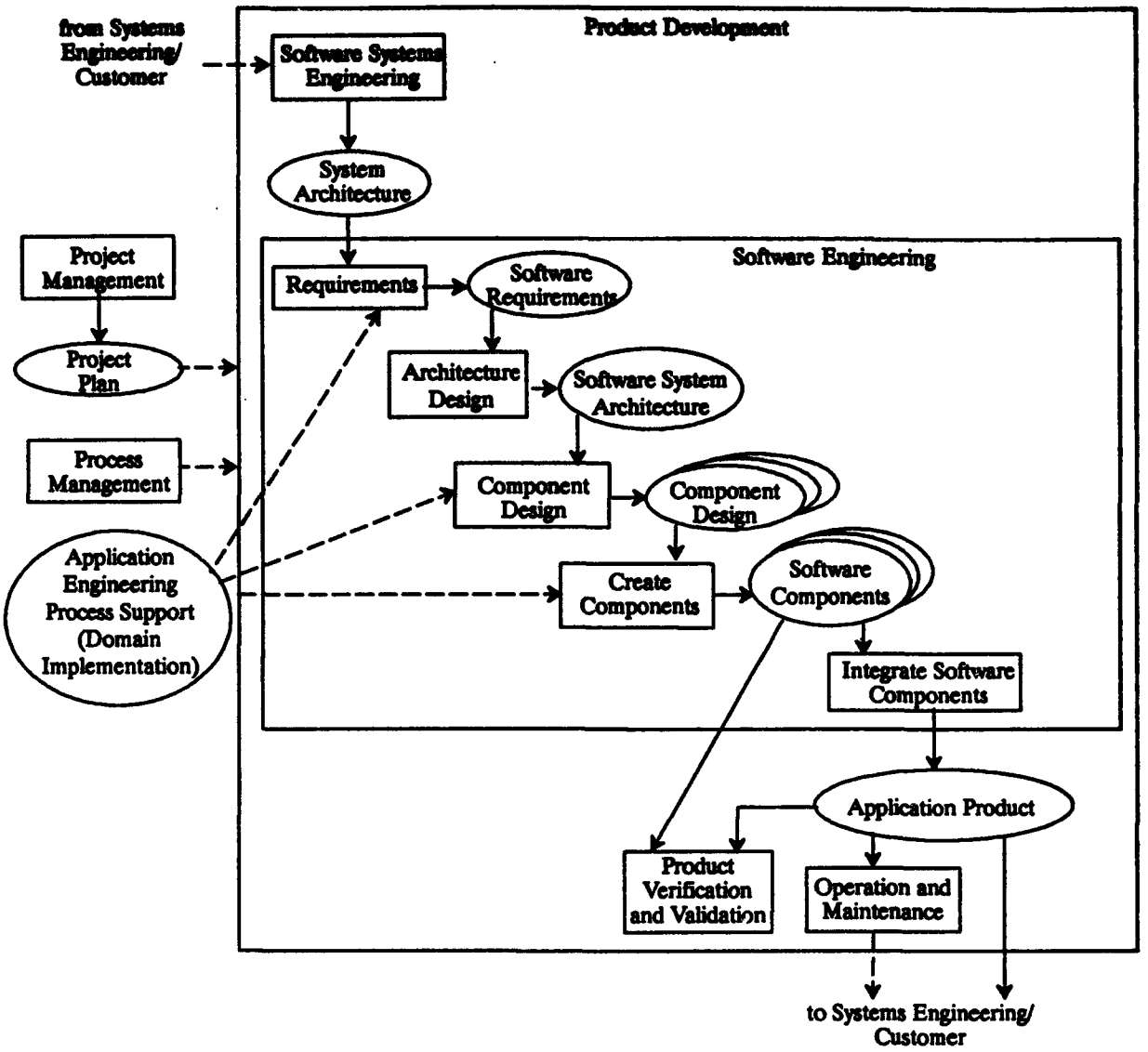


Figure 6-5. A Simple Example of an Application Engineering Process for Opportunistic Reuse

are found embedded in the Application Engineering Process, namely, software requirements, software design, software implementation, and verification and validation. However, all these activities are simplified by drawing on the reusable assets available in the reuse library.

The reuse library is constructed and maintained as part of the Domain Engineering process. Domain engineers work with domain knowledge, which includes existing systems, reusable assets currently in the reuse library, personal experience, and other available materials. Domain engineers first identify the domain, identify opportunities for reuse in terms of work products that application engineers build, then specify and implement families of work products. Opportunistic Synthesis stresses identification of existing assets rather than creating new ones for addition to the library.

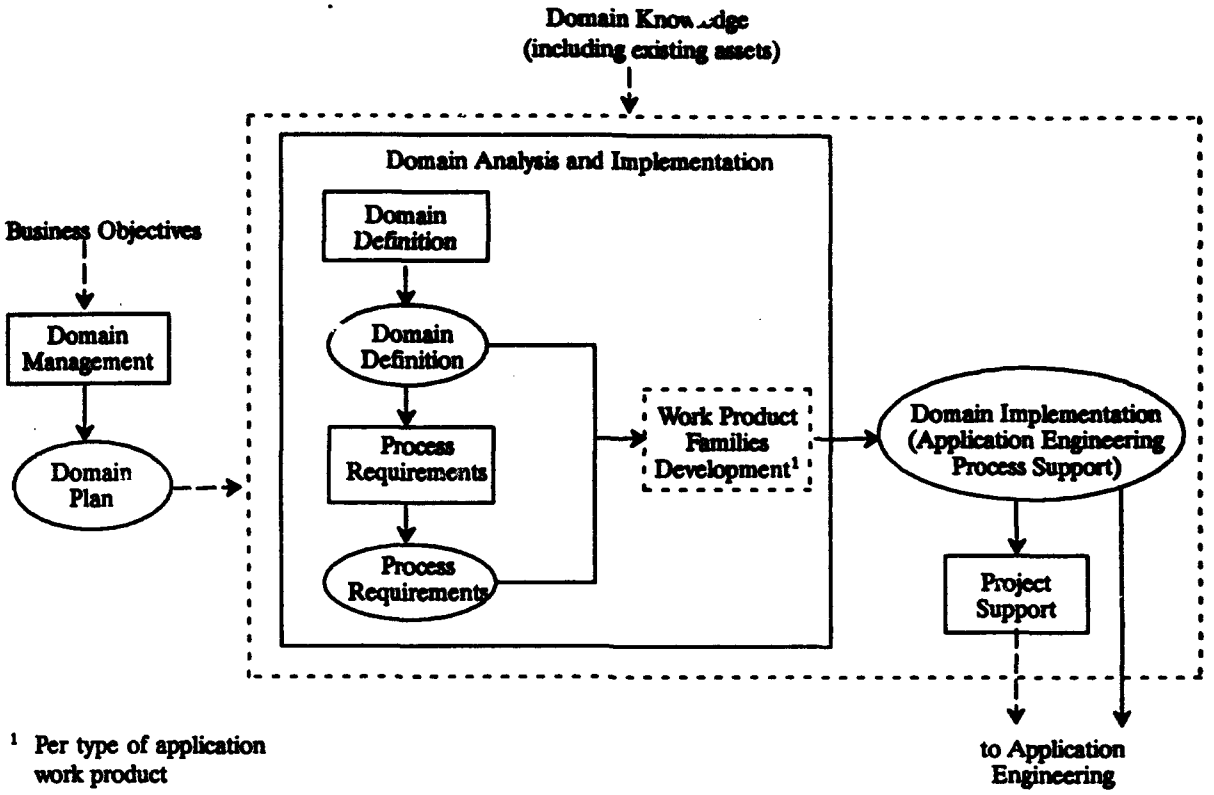


Figure 6-6. A Domain Engineering Process for Opportunistic Reuse (Simplified)

Different approaches to automation are available. For example, generators could be used. In that case, domain engineers would likely build or buy the generator itself and add it to the assets of the organization, while application engineers would use it to build products for use.

Commonly practiced software development activities can be related to a Synthesis process. However, those activities should be adapted to the context in which they will be executed. A general correspondence of activities to the process is:

- ***Requirements Analysis.*** Requirements analysis is an application engineering activity that works with existing requirements as a baseline. When practiced as a domain engineering activity, requirements analysis is a part of domain analysis; i.e., the characterization of the requirements for a complete set of planned products or business area.

- ***Design.*** For Application Engineering, the focus is on adapting an existing design. For Domain Engineering, the emphasis is creating an adaptable system architecture and the specification for each of its reusable components.

- ***Implementation.*** Implementation in a reuse process can take several forms:

- ***Creating.*** Products that are not available from the asset library must be created (developed from scratch). Application engineers will produce new products to meet the needs of a customer that cannot be met with existing assets. Domain engineers, on the other hand, produce new products to add to the asset library for future use. You should choose between these two alternatives based on expected future demand.

- ***Reengineering.*** Reengineering may be employed by domain engineers to produce assets that can be used in a variety of products.

- ***Tailoring.*** An existing asset may be tailored by an application engineer to meet customer needs. Domain engineers may also tailor existing products to be added to the reuse library.

- ***Generation.*** Generators would most likely be used as part of the Application Engineering process, especially if it is highly automated. Generators may be built by domain engineers to provide tool support for part or all of the Application Engineering process.

- ***Black-Box Reuse.*** Black-box reuse represents the limiting case where a part can be used without modification, though it may be necessary to specify parameters to be applied at compile, link, or run-time. This is an Application Engineering activity.

- ***Verification and Validation.*** Reviews and formal methods may be applied during development while various test methods are applied in the final steps. The verification and validation activity applies to complete systems and to components for the library; i.e., it applies to both Application and Domain Engineering.

The reuse library would contain the reusable components for a product family. Components should cover requirements, design, code, documentation, and test data. Generators and test tools should be included also, if available. Reference data and process guidance should be included as an aid to application engineers. It may be desirable to maintain a separate library for each product family to simplify management and use. Another library containing other existing software and components that can serve as a resource for domain engineering is also desirable. The domain implementation bubble shown in Figures 6-5 and 6-6 would produce these library materials along with Application Engineering information (e.g., domain-specific process guidance).

You should decide what assets you want to create along with the process for creating and using them. You may first think of code reuse. This may be of limited benefit unless there is a well-thought-out strategy for applying reuse. Other reusable products should be considered (e.g., requirements, designs,

test cases). Some combination of these may provide the best payoff for your business.

Documentation of your assets should also be carefully planned. What are the customer's documentation requirements (e.g., DOD-STD-2167A)? What documentation do you need from suppliers for purchased components? What components documentation is needed so that application engineers can use the assets most efficiently?

Finally, you should consider how the software development process integrates into the larger context. Who specifies the business objectives to guide the development of assets? How are contracts obtained and forwarded to application engineering? How is the finished product handled? What support is needed (e.g., for integration with hardware or for the customer)?

6.4 DEVELOP ORGANIZATIONAL APPROACH

Purpose

Identify which changes are necessary to the responsibilities of each part of the organization and changes to the structure of the organization.

Guidance

To develop an organizational approach, start with your current organizational hierarchy, identify the tasks that will be performed as part of the reuse-driven software development process, and make a responsibility mapping between the two. Consider whether library and asset management should be assigned to an existing element of the organization or a new element created for the purpose.

If you are designing a reuse program under the review or sponsorship of a particular customer (e.g., a DoD program office) rather than as an internal effort, you must identify and plan organizational responsibilities for the customer organization also. You must have a common understanding of the planned relationships between the two organizations that will exist as a result of the reuse effort. However, the following guidance assumes that your scope is only for your own organization.

Figure 6-7 is a generic organizational structure. It depicts the parts of the organization that may be affected by a reuse program. To identify the changes that you require in organizational responsibilities, you should begin by constructing a similar chart for your organization. Your chart should adjust the scope of the organizational structure to the level that you intend to apply the Reuse Adoption process.

After you have tailored the organizational structure and functional responsibilities to your own organization, you should create a matrix, similar to Table 6-1, that cross references your organization with the major activities of the reuse-driven software process that you identify in Section 6.3. For each element in the table, you can identify the responsibilities of the organizational elements. The table is an expected organizational responsibility chart for an organization that is using a member of the Synthesis family of reuse processes

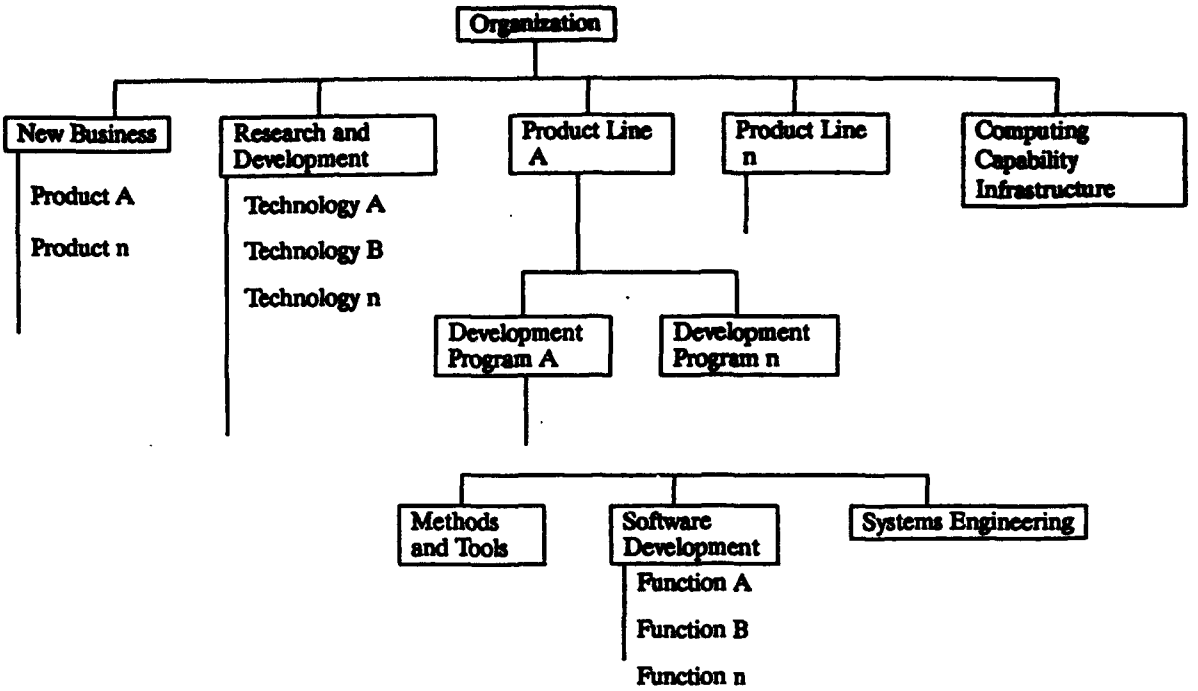


Figure 6-7. Generic Organizational Structure for Planning Reuse Program Impacts

(Software Productivity Consortium 1993b), has a strong product line, has an advanced level of reuse, and is able to assign some of the cost of reusable component implementation to its end-product sales or contracts.

If you implement a reuse-driven software development process that is not a member of the Synthesis family, you can use the Conceptual Framework for Reuse Processes (CFRP) created by the STARS program (STARS 1992b). The CFRP is intended to provide a unifying framework for specifying all reuse processes and provides a process hierarchy that is also appropriate for use in the same manner as Table 6-1.

Table 6-1. Mapping of Organizational Roles to Synthesis Reuse Process Activities

Organizational Function	Domain Engineering				Application Engineering
	Domain Management	Domain Analysis	Domain Implementation	Project Support	
New Business	R	—	—	R	—
Research and Development	—	—	RD	—	—
Product Line	SRDUAM	SRDM	SUAM	SDM	—
End Product	—	S	S	S	S
Systems Engineering	—	RUA	R	—	RA

Table 6-1, continued

Organizational Function	Domain Management	Domain Analysis	Domain Implementation	Project Support	Application Engineering
Methods Group	—	R	R	—	M
Software Development	—	RAU	DA	RUA	D

Key: S Sponsors; U Uses; R Provides Requirements; A Approves; D Develops (or Performs); M Maintains

A further description of the organizational components and potential reuse relationships and impacts follows.

- ***New Business.*** This function is responsible for identifying future customer needs, relating them to capabilities that the organization possesses, and positioning the organization to provide the best solution to customer needs.

  The new business function can become involved in the reuse program in several ways. It can:

  - Identify future requirements and constraints that can be accommodated in the current development of software assets.
  - Forecast the demand for products in terms of new or existing core capabilities or core products that should be developed. It can identify the potential economic benefits from software assets that support capabilities and products.

- ***Research and Development.*** This group is responsible for development of products and technologies that expand the core competencies of the organization. These core competency domains are candidates for reusable asset development if common interface layers can be identified, i.e., if they can be characterized and used as subdomains.

  The Research and Development group can contribute to the reuse effort by:

  - Providing domain assets for core competency areas
  - Identifying requirements on domain implementation efforts that will allow them to use the Research and Development organization's assets

- ***Product Line (Core Product).*** This management group is concerned with specific product lines. It can contribute to the reuse effort by:

  - Sponsoring product-line investments in reusable assets
  - Championing organization-wide process standards and guidelines

- Performing domain analysis for the product line
- Managing product line domains
- Developing tools to assist in domain-specific reuse
- Maintaining product-line assets
- Assuming responsibility for identifying reuse opportunities within the product line
- Providing funds to sponsor reuse training and tool acquisition
- Running incentive programs that can get the reuse mind-set started
- Promoting reuse within its division

- ***End Product (or Project).*** This management group is concerned with producing a specific product or completing a specific contract. It can contribute to the reuse effort by:
  - Sponsoring Domain Engineering activities that are identified or implemented for the first time on that end product
  - Recognizing reuse opportunities within specific projects
  - Promoting reuse on its project
  - Training its staff to use reuse-oriented methods
  - Establishing policies and procedures to enable and encourage reuse

- ***Systems Engineering.*** This implementation group is concerned with design of systems and specification of the software portion of the systems. It can contribute to the reuse effort by:
  - Recognizing system design options that permit reuse of existing assets
  - Identifying domain commonalities and variabilities to support the domain analysis
  - Using the domain analysis products as a basis for specifying systems and software requirements

- ***Methods Group.*** This implementation group performs a supporting role and is not directly concerned with delivery schedules. They conduct an ongoing effort to improve the development process and provide direct application engineering support for the tools and methods used on the project. They can contribute to the reuse effort by:

  - Identifying better methods and tools for implementing the reuse process
  - Working with developers and system engineers to improve reuse tools, methods, and processes in support of Domain Analysis, Domain Implementation and Application Engineering
  - Maintaining the project's application engineering tools and environment

- ***Software Development.*** This implementation group is concerned with producing the software and related products. It can contribute to the reuse effort by:
  - Adopting reuse methods and design principles
  - Understanding and contributing to domain analyses
  - Implementing reusable domain components
  - Developing applications by using existing assets

- ***Computing Capability Infrastructure.*** This group supports the above and is not shown in Table 6-1. This group is the central maintainer of software assets and computing equipment belonging to the organization. It can support the reuse effort by:
  - Adding facilities to accommodate repositories of reusable assets
  - Maintaining an organization-wide repository of reusable assets
  - Providing communications between staff members and external repositories
  - Acquiring reusable assets that can be used across the organization's various domain engineering efforts
  - Publicizing the assets available in the repository
  - Aiding staff members in locating sources of assets and source material for asset development

6.5 DEVELOP ENVIRONMENT APPROACH

Purpose Identify how tools and automation are configured to support the software process.

Guidance

The process approach describes the activities but says nothing about the environment in which they are carried out. The software development environment shown in Figure 6-8 is intended to do so. The software development environment should be structured so that it supports the software development process. You should avoid allowing the particulars of the environment to drive the process. You may also want to adopt an open environment so that new tools can be added over time. However, this is not always possible because development environments, including CASE tools, are often mandated by organizational or customer policy.

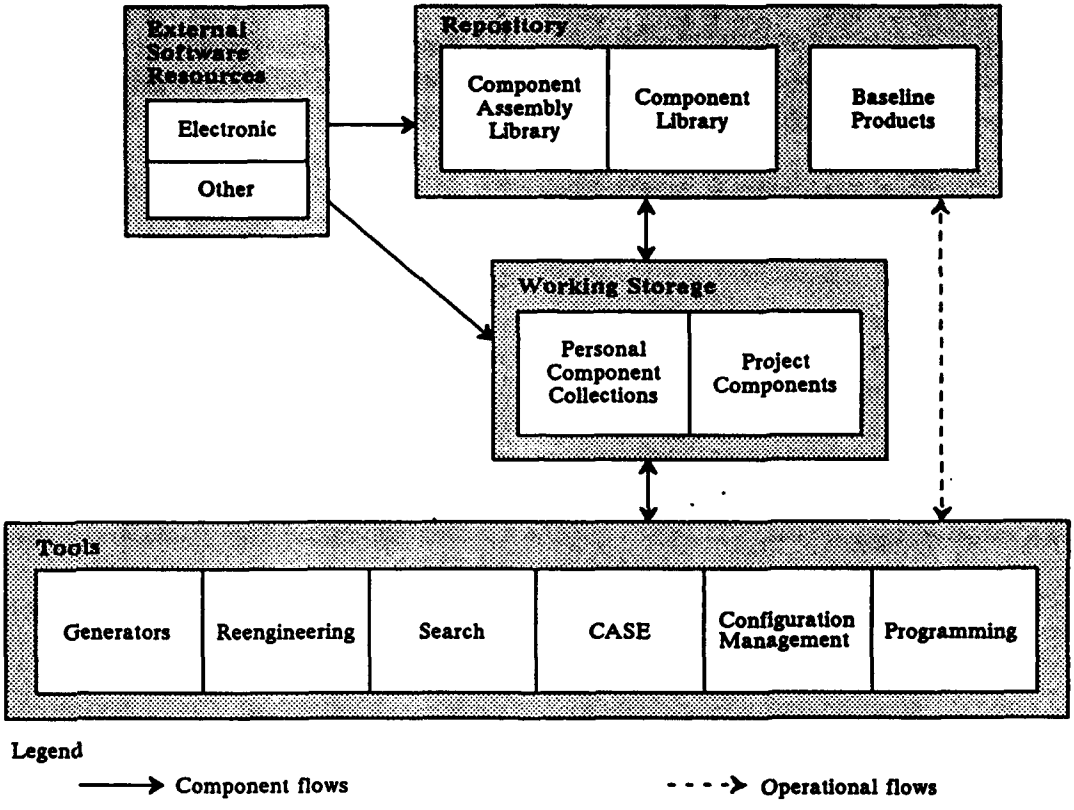


Figure 6-8. Software Development Environment

You may also need to trade off between having a preferred software development process with little tool support and having an alternative development process with good tool support. You need to consider the existing software development environment and the preferred methods and tools of the software development team.

The software development environment includes the four basic features found in any software development organization: tools with which to develop software, working storage for saving intermediate products generated by each individual developer, a repository for completed products, and external software resources. The subcategories listed are not all inclusive or limited to a specific reuse process, but they identify various reuse capabilities that are discussed below.

Tools are aids for the developer. The most basic set is the group labeled programming tools. The second group of tools is for the exacting task of configuration management. Today's environments also include tools for CASE. The remaining categories are specifically for code reuse. The search tools support the developer in locating and retrieving existing software artifacts that are available for reuse from a repository. Reengineering is one way to produce reusable components. Tools are available to assist in the process. Automatic generation of code is reuse in the sense that the design work is already done and code fragments have been built in or a dedicated set of components has already been prepared to support the generator. The selected set of tools must be coordinated with the software development process to achieve effective reuse and may itself be a major asset of the organization.

Working storage is essential for software development. At a minimum, it must accommodate working files for software in development. In addition, most programmers keep objects previously developed that may be useful or objects obtained from external sources. However, the best source of components is the repository.

The repository is a major resource for the organization. All the objects previously developed and put under configuration management should be here. In addition, the organization may acquire software components specifically for general use (reuse) on current and future projects. They may be purchased from commercial vendors, obtained from the public domain, or developed in-house. Two classes are distinguished: general components, which perform a specific function, and component assemblages, which are designed to work together and have common properties. These can also be domain-specific, adaptable components or work products linked to a process for developing a particular family of applications.

External software resources are extensive today. These include commercially available subsystems (DBMSs, GUIs, etc.), component assemblies (e.g., Booch parts), and tools (e.g., teamwork). High-quality components are purchased for specific projects or general use and may be added to the repository. Extensive libraries of free or inexpensive software (ASSET, COSMIC, etc.) are also available. Numerous collections are accessible over the electronic network, though quality varies widely. Selected free components are often added to personal collections where they serve as a source of ideas for new products.

Relating a Synthesis process to the software development environment varies with the process chosen. In the opportunistic case, domain engineers identify and catalog baseline products for work product families, while application engineers may need a full range of tools to complete development of the application. In a highly leveraged process, domain engineers should build an Application Engineering environment to support production (Software Productivity Consortium 1993b). However, the other services are available for use in any given Application Engineering process.

Again, the development environment should be designed to support the development process selected. The following categories generally describe existing reuse-related environment components that may meet your process needs:

- ***Working Storage.*** Each software developer generally has his own workspace whether it is on a mainframe computer or a workstation.

- ***Project Components.*** These are ordinary software artifacts that are in some stage of development or revision.

- ***Personal Component Collection.*** This may contain any software artifact that the developer believes may be useful for building future products and chooses to retain.

- ***Reengineering Tools.*** These are used for converting existing code to reusable assets. We include here any tools designed for browsing code, retrieving structure, or restructuring code. Such tools can be used to extract components that have broad application or can be modified to be adaptable and portable.

- ***Code Generators.*** These generate code from a declarative specification or a domain-specific higher level language and support rapid response to new requirements at low cost.

- ***Search Tools.*** These aid users in identification and retrieval of components and related documentation that is normally stored in a repository (database system). The retrieval process may or may not be visible to the user.

- ***Programming Tools.*** These include general tools for code development: editors, compilers, builders, and debuggers.

- ***CASE Tools.*** These include tools for design, simulation, testing, and performance evaluation.

- ***Configuration Management Tools.*** These assist in keeping track of versions and the status of products in development.

- ***External Resources.*** These include any accessible source from which software or tools may be obtained. From such resources tools can be purchased and added to the environment.

  Electronically accessed resources include any of the numerous on-line repositories, such as RAPID, or informal collections that are publicly available.

  A related area is electronic communication, including mail and file transfer capabilities. This permits fast and accurate distribution of

information or assets and can be a big help in rapidly resolving problems in applying assets or components from external sources.

- ***Repositories.*** These are organized, shared collections of information about business and software. They are bases of knowledge that can support software development and maintenance.
- ***Baseline Products.*** These are configuration managed, delivered products.
- ***Component Library.*** The component library contains components that typically are used individually.
- ***Component Assemblies.*** Component assemblies may be available for building systems or subsystems. These are designed to work together and an appropriate subset would be used to implement a given requirement.

6.6 DEVELOP TRANSITION APPROACH

Purpose

Identify the major stages that your organization passes through in reaching its goals. Identify:

- How the changes are accomplished
- Intermediate steps and major decision points
- Time frame of the change

Guidance

A transition approach is needed to identify intermediate steps required to get to the implementation of the strategy. Your transition approach should lay the groundwork for each transition step so that people accept and support it, the technology does not need to be forced upon the receiving organization, and the likelihood of the success of each increment of technology transition is high. Software Productivity Consortium (1993d) focuses on providing information to help you do this.

Develop Acceptance and Support

One of the first steps in any transition program should be to prepare your personnel. Guest lectures, optional employee courses, and small studies are some of the ways the new technology can be introduced in a nonthreatening way. Involve personnel in the planning process. Let people know they are participating in advancing production capabilities. All this can and should be done well before the main transition effort.

Identify Transition Stages

You should base your transition approach on the reuse goals and constraints identification task identified in Sections 3.2.2 and 3.2.3. Some of the key steps that could determine needed transition stages are:

- Removal of constraints that conflict with goals that derived from organizational objectives

- Gaining a sufficient level of staff experience with the process to scale up to use on contracted end products
- Getting the first customer who is willing to support use of a reuse-driven process

Some of the elements of transition approaches that have been successful for application of the reuse technology described in Software Productivity Consortium (1993b) and that are probably equally applicable to other software development processes are:

- Start with your most promising domain. Use the domain that rated best when you conducted the Assess Reuse Potential task in Section 3.2.1. Because getting a new technology to work the first time is difficult even under ideal circumstances, find areas where the technology is likely to work best. Make sure that you have people with domain knowledge and expertise available.
- Begin with a pilot or shadow project that is not in the main line of your product development schedule. You need to commit two or three of your domain experts to the task.
- Work iteratively, with cycles being about 3 months in duration. During the first iteration, work at a high level to describe the domain as a whole.
- During the second iteration, focus on a specific subdomain that is appropriate to the size of the staff commitment. Cycle through the entire process so that the staff gets a feel for each of the activities.
- Review the products of each cycle with the staff from the organization that will be the eventual receiver of the technology. This provides the benefits of providing feedback on the usefulness of the products under development. It also begins to get some buy-in to eventual use of the products by the receiving organization.
- Continue tuning your process and product through successive iterations until the entire organization is ready to practice the technology.

An alternative is to implement a reuse program directly in a suitable project. This has been done successfully, but certain conditions need to be met:

- The long-term reuse benefits must be substantial.
- The technical leaders need to have a clear vision of how reuse should be applied in the project and be dedicated to making it work.
- Management must share that vision or at least be willing to support the decision of its technical staff.

Identify Transition Incentives

Incentives provide a means to stimulate immediate organizational and individual interest in making the reuse program work. Incentives are most useful at the start of a reuse program. As described here, they are temporary measures that should be most closely associated with the transition approach of your reuse program.

To properly assess the set of incentives that is more effective in a particular reuse adoption program, Table 6-2 presents a basic incentives framework that characterizes different types of incentives for promoting reuse.

Table 6-2. Basic Reuse Incentives Framework

Types of Reuse Incentives		
	Individual	**Organization**
Asset Creation	A	B
Asset Reuse	C	D

- ***Asset Creation Incentives for Individuals (Type A).*** Among the most used incentives are recognition programs and awards, cash for assets, promotions, and training programs. The objective is to create enough interest in the individuals so that they will be willing to invest the extra time and effort to design architectures and interfaces that are more general and to create software that is more modular, better documented, and more thoroughly tested.

  - ***Recognition programs*** are usually implemented through a local newsletter and by employee-of-the-month award programs. The objective is to provide public recognition to software developers who have gone out of their way to create and contribute reusable assets. Peer recognition is perceived as extremely valuable, especially in large organizations, but it is an incentive that must be accompanied with other, more tangible rewards.

  - ***Cash for assets*** can be an effective incentive, especially in the early stages of a reuse program. However, you should take extra precautions to ensure that the incentive is defined in such a way that it cannot be abused by developers. To do this, the incentive must reflect the value of the asset provided rather than a less direct measure, such as the number of lines of code in the asset.

  - ***Promotions*** are effective motivators, but they must be used cautiously. There is an upper limit to promotions. A promotion usually changes a person's perspective on his job. The objective in reuse adoption is to change a person's way of developing software, from coding lines to integrating components.

  - ***Training and education*** are essential to enable employees to participate in the other incentives programs. Training should

include teaching employees how to use the reuse library, how to create reusable assets, and how to reuse. Students should be formally exposed to the software development process practiced by the organization and to the reuse process. Training should be portrayed as a privilege and as a means to enable students to participate in other incentive programs.

- ***Asset Creation Incentives for Organizations (Type B).*** Organizational incentives include incentives aimed at motivating developers within the organization to create reusable assets and incentives to improve the organization as a whole. The former focuses on management; the latter incentives focus on market opportunities and investment opportunities.

  - ***Market opportunities*** emerge when reusable assets are created for multiple applications in the same domain. Marketing reusable assets is just another business that can bring revenue to the organization. Market opportunities can be internal, among divisions or projects, or external, across organizations. An internal market for reusable assets may be extremely beneficial for a large organization where different divisions can reuse assets from each other. The main benefits in this market are in cost savings and quality improvement. An external market brings direct revenues to the organization.

  - ***Investment incentives*** are based on the concept that reusable software assets are actually capital assets. The same motivation an organization has to invest in capital assets should be applicable to software assets. A company's worth may increase significantly if it invests in creating reusable assets for a particular market niche (i.e., domain). This improves the company's bidding capacity in contracts and gives it a definite competitive advantage. But like any other investment, it takes time to accrue value. A proprietary reuse library is a mechanism to keep the organization's software portfolio.

  - ***Management incentives*** to generate reusable assets include some restructuring of the management bonus program and some organizational structure changes. A management incentives program should include rewards for contributing to the organization's worth through creation of reusable assets. A two-track reward program is possible for managers. Managers should be rewarded by the number of reusable assets contributed to the library, and they should also be rewarded by timely product delivery.

- ***Asset Reuse Incentives for Individuals (Type C).*** Several approaches can be taken to motivate individuals to reuse software. Use many of them concurrently. Work practice incentives, product perception incentives, and reward incentives are a few of the incentives of this type.

- ***Work practice incentives*** place emphasis on creating groups or teams to design and develop systems. Group work increases communication, promotes sharing of ideas, and stimulates reuse. Individual work should be discouraged. Even for small tasks, teams of at least two individuals should be assigned.

- ***Product perception incentives*** include assigning development of complete systems to a single team rather than partitioning systems into small components for individual development. Another product perception incentive is to follow a reuse-driven process. If the same team participates in all facets of the development process, its opportunity for reuse is enhanced. Thus, a well-defined, revised, and accepted reuse-driven process facilitates the software development process.

- ***Reward incentives*** for reusing software have some overlap with incentives for creating reusable assets. Reward incentives include recognition, promotion, and bonuses. A reuser-of-the-month award is recommended as part of a recognition program. Team rewards can also be given on the basis of how a team consistently follows the reuse process. Managers should also be rewarded for promoting reuse practice in their projects.

- ***Asset Reuse Incentives for Organizations (Type D).*** The incentives to organizations for reusing assets include higher productivity, improved quality, reduced maintenance, increased saving, and opportunities to win new contracts. The bottom line for these incentives is financial benefits. Organizational incentives should originate at the top management level. Top management must be the ultimate promoter of an institutionalized reuse program.

  - ***Productivity, quality, maintenance, and savings*** are interrelated and are a natural direct effect of reuse. A formal reuse program can bring a sustained increase to an organization in all these factors. Organizations such as Toshiba, NEC, Hitachi (Cusumano 1991), GTE, Fujitsu, and Raytheon (Prieto-Diaz 1991) have demonstrated that institutionalized reuse programs bring substantial increase in productivity, quality, and savings and reduce maintenance.

  - ***Opportunities to win new contracts*** result when a reuse program has been in place and an organization is able to quickly put bids together from reusable components. A reuse program provides not only the reusable components but the documentation associated with those components, such as reuse metrics, cost and maintenance histories, and previous applications records. This information allows an organization to create realistic proposals and to outbid competitors.

APPENDIX A. DOMAIN ASSESSMENT MODEL

The potential for reuse in your business area is a key decision driver when implementing a reuse program. Understanding your potential will help you determine how much you should invest in a reuse program and where you should focus your investment. You assess your business area potential within the Understand Context activity. You then use this understanding to establish reuse adoption goals that are commensurate with your potential and to develop reuse adoption strategies that exploit your potential. You will also use information regarding your potential in the Analyze Risks and Select Strategy activity as input to your economic analysis.

The purpose of this appendix is to provide a Domain Assessment Model that will help you assess your potential for reuse in a particular business area. There are five known factors that must be considered in assessing the reuse potential in a domain. Namely, market potential for products, existing domain assets, commonalities and variabilities between systems in the domain, domain stability and maturity, and domain standards. These factors identify features of the domain and of the relationship of your organization to the domain that will permit successful reuse. These are discussed in the following sections.

This appendix provides a general model for obtaining a quick, qualitative assessment. A set of concrete attributes is associated with each factor. These are designed so that a team of experts familiar with the product domain and your organization can estimate the potential for reuse in the target domain and produce a summary that you can use in developing reuse strategies.

A.1 WHY IS A DOMAIN ASSESSMENT MODEL NEEDED?

Potential benefits to an organization using a Domain Assessment Model include:

- Lower the risk to implementing a reuse program. The model provides a characterization of the domain so that the reuse potential can be realistically evaluated and the risks identified.
- Systematize the identification of reuse opportunities. The model identifies the reuse opportunities. These relate to cost, schedule, quality, productivity, return on investment, customer satisfaction, and competitiveness.
- Provide the basis for selecting appropriate reuse technologies and tools. The model addresses issues such as variability, stability and standards which are important factors in selection of reuse technologies and tools for improving the development process.

A.2 INTRODUCTION TO THE DOMAIN ASSESSMENT MODEL

The Domain Assessment Model is to be used by an organization to gain an understanding of the reuse potential for the target product domain and the position of the organization with respect to that

domain. The Domain Assessment Model is designed to be used by persons familiar with the target product domain and the assets that the organization already possesses.

The domain assessment is designed to be used to assess or reassess a domain of interest for its reuse potential. The results can be used to plan reuse adoption strategies or in-depth studies of benefits and risks most relevant in the situation that the organization faces.

Domain assessment has its roots in domain analysis. Briefly, the latter is an engineering approach for identifying the functional requirements for a family of products, the context in which they operate, and structures that enable software reuse (Prieto-Diaz and Arango 1991, Jaworski et al. 1990, and Software Productivity Consortium 1993b). Domain assessment examines a domain from a business viewpoint to evaluate the potential for profitably applying reuse.

A.3 THE DOMAIN ASSESSMENT MODEL

The model includes five factors that are important to successful reuse in a particular domain. Within each factor, one or more attributes are identified that describe favorable reuse circumstances. This list of factors is organized in the order in which they would probably be analyzed and the order of impact on reuse feasibility. For example, if market potential is deemed to be poor, one might drop the idea of introducing reuse in most circumstances. On the other hand, lack of standards in the domain may have little impact. The domain factors are:

- Market Potential
- Existing Domain Assets
- Commonalities and Variabilities
- Domain Stability and Maturity
- Standardization in the Domain

Each factor is defined by one or more attributes that characterize a domain. An attribute is a descriptive statement about a domain or the organization's relation to the domain. Each factor is cast as a positive statement that describes a favorable reuse situation. For example, under Quality of Assets, there are two attributes:

AQ-1. Packaging existing assets for reuse will be much less expensive than developing reusable assets from scratch.

AQ-2. Assets are adaptable to meet expected market needs.

The purpose is to identify characteristics of existing assets that will favor applying them as the basis of a reuse program. If the attribute is not exhibited in the target domain, reuse may still be feasible. It simply identifies a risk or constraint that must be met in the reuse program.

A.3.1 Market Potential

The potential for systems to be built in the domain is critical in determining whether the domain engineering investment should be made. Company goals, current contracts, past developments,

technology trends, and defense budget forecasts are a few of the indicators you can use to estimate how many versions or models of your system your company will be developing.

There are two components of the situation that you need to consider. One is the internal production situation or production plan for your organization. The other is the market situation in which you will sell these products. Both of these will influence how reusable assets will be obtained and applied.

Product Situation

The product situation is the product development activity in which software reuse is to be introduced. Typical situations are products in development, anticipated enhancements to current products, or planned additions to the product family.

Attributes

PS-1. New planned products could benefit.

PS-2. Products currently in development could benefit.

PS-3. Products ready for enhancement could benefit.

PS-4. The value of using reusable assets is high for the given product situation.

Supplemental Information

You need to identify which of the above product situations (PS-1 to PS-3) matches the circumstances under which you are considering introducing reuse. Consider the example of a large fighter aircraft production contract in its initial stage (PS-1). Reuse was identified as a cost saver. The developers recognized that numerous components of the system would be tested using the same automatic test equipment and that the software that controlled the automatic test equipment had many common elements. If these could be developed in advance, then the component developers would not have to write their own.

You are asked to consider the value of reuse in your situation on attribute PS-4. Software reuse technology can be used to reduce costs, reduce development times, improve quality, or in some other way improve your long-term prospects. You must identify how it will help in your product situation and estimate the magnitude of the benefits.

Market Situation

The market situation is the strength of the demand in the market and its expected growth. Specifically, it can be described as the expected sales to this market and your share of it. One of your organizational goals may also be to expand your market share through reuse. Finally, if that customer base is representative of the market in general, your assets and understanding of market needs are probably good. Otherwise, there may be significant gaps in your assets that need to be addressed. The bottom line is that the market demand for your products must be sufficient to provide good return on the investment in reuse.

Attributes

MS-1. There is substantial demand in the given market.

MS-2. The current customer base is representative of the market.

MS-3. Increased market share will result from investment in reuse.

MS-4. This market is expected to grow.

Supplemental Information

The demand for versions of your product rather than gross sales is more important when examining reuse. For example, assume that each manufacturer has unique requirements in production management software. In this case, the demand for versions would appear to be good. If, however, all your current customers produced fiberglass products, your software may not have strong appeal to small engine makers because your current customers are not representative of the larger market. On the other hand, reusing current assets to produce management software targeted at small engine makers constitutes an opportunity to expand your share of the overall market.

It should also be obvious at this point that you must have a clear definition of your market when performing an assessment. The market under consideration could be narrow, as in "those now using current versions of our product," or broad, as in "anyone who needs to perform our product's function."

A.3.2 Existing Domain Assets

Reusable assets are the foundation of a reuse program. You need to consider existing assets that are reusable or that can be converted for reuse. These are hard assets in the sense that they can be named and counted. The experience of your staff and the domain knowledge embedded in your organizational structure are soft assets: they are not easy to identify or quantify. Nevertheless, they are also important to reducing development time and cost and to effectively applying the hard assets. Steps you can take to understand your domain assets are described below.

Individual and Organizational Expertise

Your principal asset in any development activity is the experience of your personnel and the organization to which they belong.

Attributes

EE-1. There are individuals on staff who are experts in the domain (i.e., they have experience building and using domain assets).

EE-2. Your organization has experience building products for this domain.

Supplemental Information

The first attribute really focuses on the question of how ready your organization is to produce with reusable assets. Personnel need to understand where and how to obtain the assets, how to use or adapt them, and how reusable assets are applied in the context of your development process.

You should evaluate the contribution of your organization. Organizational expertise is usually based on experience in building products for the domain. The management of product development will be much easier if the domain falls within the charter of the organization.

Availability of Existing Software

The less that needs to be generated during each production cycle, the higher the potential return. Thus, you want to consider what is available to support

each phase of the development cycle. Requirements, design, source code, test documents for the software, and system and hardware specifications and any management documents, such as development and sustaining engineering plans, are all assets in domain development. Finally, you need to judge the ability of staff personnel to use existing assets in developing products.

Attributes

AS-1. Existing assets support all work products needed to develop new members of a product family (i.e., requirements, design, code, test data, and documentation).

AS-2. Existing assets are well integrated.

AS-3. The features of all assets are clearly traceable to the requirements.

Supplemental Information

These three attributes address the software development process. You should consider how the identified assets will contribute to the overall production process. For example, having a requirements document to describe the product family is good. But is it organized to be easily adapted? Are the links from requirements to design mapped out so changes can be easily traced?

Quality of Existing Software Assets

Quality can have several meanings, ranging from well documented and reliable software to software that provides customer satisfaction. For the case of reuse, quality is a measure of how well software meets its own specified requirements. The quality of your current software is, in most cases, proportional to its salvage potential, i.e., the money saved by repackaging or reengineering the existing asset rather than building one from scratch.

Attributes

AQ-1. Packaging existing assets for reuse will be much less expensive than developing them from scratch.

AQ-2. Assets can be adapted for expected market needs and packaged for reuse at less cost than developing them from scratch.

Supplemental Information

These two attributes address two circumstances for converting an asset to a packaged ready-to-use form. Packaging refers to preparing documentation, abstracts, catalog entries, and other items needed to permit the asset to be located, evaluated, and reused with minimal effort. The first case only addresses packaging because it is assumed that the functional requirements have not changed. The second case concerns the situation in which projected requirements are different (e.g., more precise or more flexible). In the first case, the asset can be treated as a black box. In the second, you must look inside and evaluate whether it can be readily understood and adapted. In either case, you need to judge whether starting with existing assets is more cost effective than building them from scratch.

A.3.3 Commonalities and Variabilities

The domain's commonality and variability are the primary factors that lead to the feasibility of creating reusable assets that allow a variety of systems to be built. Without commonalities, reusable

components cannot be built. But if components do not accommodate legitimate variations in the needs of end users, they may not have a large enough customer base to justify their development as reusable assets.

Commonalities in Behavior, Structure, and Implementation

Common components of a product are the essence of reuse. You should look for common requirements, structure, and behavior across the anticipated product family. Consider the amount of commonality in designs and documentation as well as code. If a common architecture is feasible, then exploiting commonality will probably be easier.

Attributes

CR-1. A large portion of the products can be built with common components.

CR-2. A common architecture for the domain is feasible to enhance reuse.

Supplemental Information

Reuse will not have a substantial impact on development unless a sizable portion of your products is derived from reusable assets. This is an area where you will almost certainly need some supporting material (see Section 4.5.3).

Variability

Variability is what differentiates individual members of the product family (as well as other products in the domain). Each of the attributes below corresponds to some way to accommodate variable requirements at reasonable cost. You should estimate how useful each is for your domain.

Attributes

VR-1. Variable requirements can be managed (i.e., either by isolation to separate components or by providing an enumerated set of options).

VR-2. Common components can be isolated from variabilities in supporting hardware and software.

VR-3. Nonessential variability in customer requirements can be managed.

Supplemental Information

The most basic method for accommodating variability in requirements is to isolate each class of variability for a separate component, thus localizing the revisions needed when the corresponding requirement changes.

Where variability can be enumerated, a preplanned product line is feasible. Otherwise, new versions must be produced on demand and standard interfaces are needed to accommodate the types of variability expected. It may not be necessary to enumerate all possibilities. For example, most word processors have unique file formats. It may be sufficient for your software to read only the most popular ones.

A flexible interface can open the way to accommodating wide variations in specific functions. This is common practice for printers (for which there is no widely used set of control codes). A small driver module is written to translate your commands and data to a form that a particular printer will understand.

There is a natural tension between the simplicity that minimizes development cost and the endless list of features that users dream up. The last attribute suggests that the producer can take a proactive position: providing

functionality while selling users on the idea that it is sufficient. Even in an era of flexible manufacturing you must decide how much variation is sufficient to meet market demands. Consider how much control you have over product requirements in evaluating this attribute.

A.3.4 Domain Stability and Maturity

Domain stability or lack of it is both a problem and an opportunity. Rapid changes in customer needs or underlying technology for a product would appear to make reuse ineffective. This is closely related to the maturity of the product domain or of the supporting technology (see Section 4.5.4 for a definition of maturity). However, where components can be identified that are not likely to change (i.e., be common across many future versions), reuse can reduce the cost and time required to produce each of the many required versions.

Stability of Customer Needs

As a customer's needs or requirements change, so must the product. You need to understand how extensively and quickly those needs are likely to change. You need to consider how much time you will have to recover the investment in reusable assets.

Attributes

CS–1. There is sufficient time to recover the investment cost before evolution of customer needs makes the assets obsolete.

CS-2. The trends in customer needs are clear enough to permit the design of assets that will accommodate the changes.

Supplemental Information

These two attributes describe different ways of dealing with changes in customer needs. Where an asset will be reused frequently, benefits will be realized before the assets become obsolete. Where certain customer trends are evident, adaptable features corresponding to those trends can be built into the asset, thus extending its useful life. Looking at the past can identify the pace of change or trends that can be accommodated within design.

Technological Evolution

Technological change is the other source of instability. You should assess the rate of change in the underlying technology for the domain and the sensitivity of the assets to those technologies. Where much change is expected, it may be necessary to constrain asset designs to isolate them as much as possible from the anticipated changes. It may also be necessary to concentrate on abstract assets that are unaffected by the changes.

Attributes

TS-1. There is sufficient time to recover the investment cost before evolution of the underlying technology makes the assets obsolete.

TS-2. The technological trends are clear enough to permit the design of assets that will accommodate the changes.

Supplemental Information

These two attributes describe different ways of dealing with changes in technology. These attributes are analogous to those described above for customer requirements. At least technological trends are somewhat easier to predict. However, you must look out for possible breaks with past trends. A

different technology can arise and displace the old. For example, there is the possibility that optical recording will replace magnetic recording along with all the associated software.

A.3.5 Standardization in the Domain

Standards do not generally define a product. Rather, they define certain individual features of a product (e.g., interfaces or recording formats). Software corresponding to a standard feature can be fixed as a reusable component and may be available commercially. Some features of a new domain will usually be covered by existing standards, and new standards will evolve for additional features (see Section 4.5.5 for a definition of various levels of standards).

Standards may also apply at the architectural level. When a product domain is well understood, developers usually adopt standard approaches to design and development. Compilers are a good example of this phenomenon. Compilers are not usually developed from scratch. You either buy one or use a compiler generator, such as Lex and YACC, to create one.

Standards

To assess this factor, consider features of the domain that are covered by existing standards, features that will likely become standards, and features that you can standardize to your advantage.

Attributes

FS-1. Existing or planned standards will stabilize requirements.

FS-2. COTS components matching existing standards are available.

FS-3. Industry standards are coming, and a substantial market will exist for reusable assets that conform to them.

FS-4. The organization has a key position in the domain and can lead in developing needed standards.

FS-5. Organizational standards are an appropriate basis for designing reusable assets.

Supplemental Information

The following paragraphs provide an overview of the various kinds of standards and the effect they can have on a product. See Section 4.5.5 for additional discussion of standards.

Some markets simply do not develop until official standards are established. People recognized some time ago that compact disks would be an attractive medium for computer data. But until a robust coding system was developed and agreed to, no one would take the plunge. After the standards were in place, production of compact disk read-only memory (CDROM) disks began, and sales of CDROMs and drives took off. Of course, this required development of software for authoring and accessing the disks. No one would question that the market for related products is already quite substantial. Some of the early participants in developing CDROM software technology could reap great rewards, as did Microsoft, following its recognition that every microcomputer would need a disk operating system.

Standards are not always that critical, but the market demands a certain amount of consistency. The windows style of graphic user interface that traces its roots to Xerox PARC, for example, has found expression in a number of different operating systems. Some of these have become de facto standards (e.g., X-windows).

Some circumstances do not require a standard, but a developer can still use internal standards to advantage. For example, you could standardize the common elements or architecture of an entire product line, such as medical instruments. While industry-standard windows interface might be too large or complex for that application, customers would probably appreciate a standardized look and feel, which you could define.

This page intentionally left blank.

APPENDIX B. REUSE CAPABILITY MODEL

This appendix describes the RCM. The RCM is a self-assessment and planning aid for improving your organization's reuse capability. Apply the RCM within the reuse adoption process to aid your organization in making an intelligent business decision on how to practice reuse (see Section 3.2 and Section 5). The RCM has two components: an assessment model and an implementation model. Use the assessment model to understand the present state of your organization's reuse practice. It will help you identify your organization's strengths and improvement opportunities. Use the implementation model in planning the implementation of your reuse program. It will help you establish reuse adoption goals (the desired state of practice) and the staging of the goals (as a transition approach) that you will use to evaluate your alternative reuse adoption strategies.

The primary users of the RCM are companies that are developing or maintaining multiple software systems, versions, or products for one or more customers (government or commercial). The model may be used by organizations that have no reuse program and are planning to start one, as well as organizations that have a reuse program in place and plan to improve their reuse capability. Specifically, the model is applied by those individuals within an organization who are responsible for defining and implementing the organization's reuse program.

Use of the model requires an organization to take a comprehensive view of its process for software development with respect to reuse. The model is applied across the levels of an organization from the product (or project) level through the product line level to the business level. It is applied across the organization's functional boundaries, such as management, engineering, legal, marketing, research, etc. It also requires a company to look outside of its own boundaries to its customers and suppliers.

Section B.1 describes the RCM, its major components, underlying concepts, and use. Section B.2 documents the assessment model, and Section B.3 documents the implementation model.

B.1 INTRODUCTION TO THE RCM

To successfully adopt a new technology, the organization must understand its present state of practice, be able to identify the desired state of practice, and develop a strategy that will successfully move the organization toward the desired state (Przybylinski, Fowler, and Maher 1991). Many organizations today do not have a clear understanding of the state of their reuse practice, and it is not clear to them what constitutes the desired state. It thus becomes very difficult to develop a strategy that would lead to an improved reuse practice.

The RCM is designed to support an organization in assessing its present state and identifying a desired state. The RCM has two components: an assessment model and an implementation model. The RCM is divided into two models to emphasize its dual use and to separate concerns to avoid misinterpretation of the results from using the model.

The assessment model consists of a set of critical success factors, defined in terms of goals, that are used by an organization to assess the present state of its reuse practice. From this assessment, the organization will get a list of its strengths and a list of potential improvement opportunities.

After an organization completes the assessment, it selects the desired state and develops a strategy, which can be a very complex task. To aid organizations in this task, the implementation model helps in prioritizing the critical success factor goals by partitioning them into a set of stages. The stages provide guidance—not a rigid approach—for establishing reuse adoption goals (the desired state) that are used to evaluate alternative reuse adoption strategies. There may be multiple implementation models, depending on the scheme used for prioritization and partitioning; this appendix describes an implementation model based on a risk-reduction scheme.

Underlying the assessment model and implementation model is the concept of reuse capability. Section B.1.1 provides a definition of reuse capability. Subsequent sections describe the assessment model and implementation model in more detail.

B.1.1 Reuse Capability

Reuse capability refers to the range of reuse results that an organization is able to achieve through its process. Improving these results and their range is the purpose of the RCM. To improve these results requires that they be precisely defined. Thus, reuse capability is:

> The range of expected results in reuse effectiveness, proficiency, and efficiency that can be achieved by an organization's process

Reuse effectiveness and proficiency are defined in terms of reuse opportunities. A reuse opportunity is an occasion where an asset (existing or to be developed for reuse) may satisfy a need (current or anticipated). An asset is any tangible resource that may apply to the solution of a problem (e.g., requirements, design, code, test cases, etc.). Within the set of reuse opportunities, there are potential reuse opportunities and targeted reuse opportunities. Potential reuse opportunities are the set of reuse opportunities that will result in actual reuse when exploited. Targeted reuse opportunities are the set of reuse opportunities toward which an organization directs its effort (implicitly and explicitly). A targeted reuse opportunity may not always be an actual reuse opportunity (e.g., when an asset is developed for reuse for which there is no real need).

Reuse effectiveness is the ratio of the value of actual reuse opportunities exploited to the value of potential reuse opportunities. It may be represented as:

$$R_A / R_P$$

where:

R_A = Total value of the actual reuse opportunities exploited

R_P = Total value of the potential reuse opportunities

Reuse proficiency is the ratio of the actual reuse opportunities exploited to the organization's targeted reuse opportunities. The target may be implicitly or explicitly defined. Reuse proficiency may be represented as:

$$R_A / R_T$$

where:

R_T = Total value of the targeted opportunities for reuse

Reuse efficiency is the ratio of reuse benefit to reuse costs. It may be represented as:

$$N(C_{NR} - C_R) / C_D$$

where:

N = Number of systems, versions, or products developed with the reusable assets

C_{NR} = Cost of developing new assets without reuse

C_R = Cost of using (finding, evaluating, adapting, etc.) reusable assets

C_D = Cost of domain engineering (acquiring or developing assets for reuse, building a reuse infrastructure)

Various value functions could be used to compute R_A, R_P, and R_T. One possibility would be the cost of meeting the opportunities without reuse. R_P, then, would represent the upper limit on possible cost savings from reuse. Reuse efficiency was derived from the model for return on investment from the reuse economics model (Gaffney and Cruickshank 1992). Reuse efficiency is also analogous to the quality-of-investment measure defined in Barnes and Bollinger (1991).

NOTE: The definition of reuse capability and the mathematical expressions provided above are used in this model to precisely describe the desired reuse results; they are not metrics that an organization must calculate.

Figure B-1 illustrates the relationship between reuse capability and the concepts of reuse proficiency and effectiveness. For the purpose of illustration, assume that the ovals represent the sets of potential and target reuse opportunities and the actual reuse opportunities exploited. Assume also that the value of the assets corresponding to these opportunities is equal to the area of the ovals. Reuse proficiency and effectiveness, then, are equivalent to the area of the actual reuse oval divided by the area of the target and potential ovals, respectively. The set of potential opportunities may correspond to a single domain, subdomain, or multiple domains or subdomains.

Figure B-1 presents two possible cases: one resulting in a low reuse capability, the other in a high reuse capability. The low reuse capability case is characteristic of an ad hoc approach to reuse where the potential opportunities are not identified (illustrated by dashes). Because the potential opportunities are not known, the target opportunities will likely fall outside the potential opportunities. The target opportunities may also not be explicitly defined (or planned) but represent the sum of the opportunities pursued by individuals. When the target opportunities fall outside the potential opportunities, the actual reuse is constrained to the intersection. The end result is a low reuse proficiency and

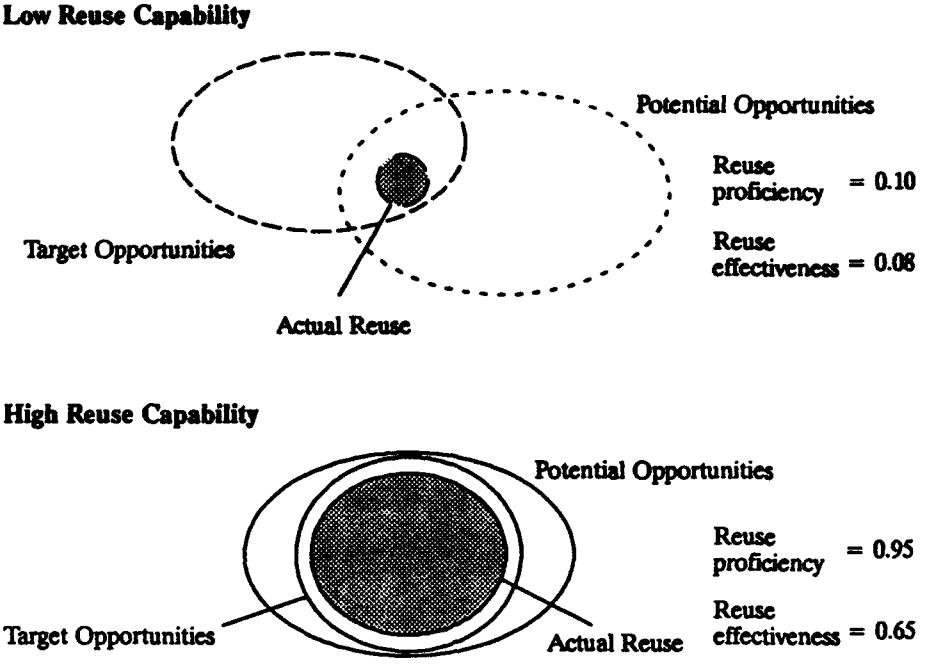


Figure B-1. Reuse Capability

effectiveness. The reuse efficiency is also likely to be low because effort is expended on opportunities that do not result in actual reuse.

The second case is characteristic of systematic reuse (Wartik and Prieto-Diaz 1992) where the organization identifies its potential opportunities, ensures that the target set of opportunities falls within the potential, and has a process that ensures the target is met. The target may not contain the entire set of potential opportunities because the potential benefit from those opportunities outside the target may not be worth the additional reuse investment; i.e., the target is focused on the opportunities with the highest payoff.

In short, the aim of the model is to aid an organization in achieving the type of results illustrated in the high reuse capability case in Figure B-1, i.e., to be able to achieve more reuse (effectiveness), consistently meet reuse targets (proficiency), and maximize the payoff from reuse (efficiency).

B.1.2 Assessment Model

The assessment model is a mechanism to be used by an organization to gain an understanding of its current reuse capability and to identify potential opportunities for improving its capability. However, it is not sufficient to develop a plan based on reuse capability alone. Other factors, such as organizational goals, budget, schedule, domain suitability, etc., vary with the situation and should be taken into consideration. The situation-dependent factors are assessed separately within the reuse adoption process, and the results are combined with the reuse capability assessment results to develop a plan that is appropriate to the given situation.

B.1.2.1 Critical Success Factors

The assessment model consists of a set of critical success factors that correspond to issues most critical to improving reuse capability. For instance, one factor critical to improving reuse effectiveness

is needs identification. For an asset to be reused, it must satisfy a need; thus, an organization should identify these needs and use them as a basis to acquire or develop reusable assets. This will help improve the organization's reuse effectiveness by focusing its efforts where there are needs to be filled. Table B-1 lists the set of critical success factors.

Table B-1. Critical Success Factors

Application Development Factors	Asset Development Factors	Management Factors	Process and Technology Factors
Asset awareness and accessibility	Needs identification	Organizational commitment	Process definition and integration
Asset identification	Asset interface and architecture definition	Planning and direction	Measurement
Asset evaluation and verification	Needs and solutions relationships	Costing and pricing	Continuous process improvement
Application integrability	Commonality and variability definition	Legal and contractual constraints	Training
	Asset value determination	Intergroup coordination	Tool support
	Asset reusability		Technology innovation
	Asset quality		

The factors are organized into four groups: application development, asset development, management, and process and technology. The groups reflect different views or roles of reuse in an organization.

The application development factors correspond to issues critical to the use of reusable assets in the development or maintenance of end products. These factors cut across the activities performed in the organization's software development process. However, the model is defined so that it is not specific to a single development process, such as waterfall or spiral.

The asset development factors correspond to issues critical to the acquisition or development of assets for reuse. These factors cut across the types of reusable assets, such as code, design, requirements, test cases, architectures, etc. Like the application development factors, the model is defined so that it is not specific to a single reuse approach, such as compositional or generative approaches.

The management factors correspond to issues critical to management's role in facilitating reuse. These factors cut across organizational levels from the product level to the product line level and to the business level. The factors also cut across management functions, such as product management, legal, finance, and marketing.

The process and technology factors correspond to general process issues that are applicable across the application development, asset development, and management views. Many of these factors are adaptations of those CMM key process areas that are also critical to reuse.

B.1.2.2 Critical Success Factor Goals

Each of the critical success factors are defined in terms of one or more goals. The goals identify the results to be achieved to satisfy a critical success factor. There are many ways to perform reuse, and

new ways will continue to be developed. Thus, the critical success factor goals focus on "what" is to be accomplished rather than "how" it is to be accomplished to provide the needed flexibility. For example, goals for the **needs identification** factor state that:

1. Current developer needs for solutions are identified.
2. Anticipated developer needs for solutions are identified.
3. Current customer needs for solutions are identified.
4. Anticipated customer needs for solutions are identified.
5. Identified needs are used as a basis for acquiring or developing reusable assets to meet the specified needs.

These goals capture the issue that if an asset is to be reused, then it must satisfy a need. These goals also distinguish the different classes of needs to be addressed. There are customer and developer needs: a customer may need a tool for developing a flight plan, and a developer may need a function to compute the time to travel between two points given the distance and velocity. There are also current and anticipated needs: when the current need specified by the customer is a tool to develop a flight plan, an anticipated future need might be the ability to prefly the flight plan in a simulation. The ability to identify needs in each of these classes can greatly improve an organization's reuse capability.

Section B.2 describes the set of critical success factors and corresponding goals. Section 5 describes how to conduct an assessment of your reuse capability.

B.1.3 Implementation Model

Establishing improvement goals and developing strategies to meet those goals can be a very complex task when taking into account the wide spectrum of information generated from the assessment. The implementation model helps reduce this complexity by prioritizing the goals associated with the critical success factors and partitioning the goals into stages. The stages represent possible operational steady states in an organization's reuse practice, and successive stages build on practices established at earlier stages.

The set of implementation stages will vary depending on the scheme used to prioritize and partition the goals. There are multiple paths to success, and there is no one best path for all organizations. Thus, there can be multiple implementation models.

This version of the RCM provides one implementation model; however, other implementation models may be generated if warranted. This first implementation model is designed to be applicable to a broad audience. It is based on a risk-reduction growth scheme; i.e., the goals are prioritized in a manner that reduces the probability of failure (by deferring goals that might require the use of advanced practices to later stages when appropriate) and minimizes the consequence of failure (by initially isolating the scope of the organization's reuse practice and then growing it within the organization).

An implementation model based on a risk-reduction growth scheme has a number of advantages. It increases an organization's probability of success in implementing reuse. It allows for an earlier return on investment. It encourages organizations to try reuse, and it enables them to evolve their reuse practice.

There are four stages in the risk-reduction growth implementation model: opportunistic, integrated, leveraged, and anticipating. The names of the stages provide a handle for referencing the stage. They

were chosen to reflect a key characteristic of their stage; they do not represent a capability scale. A brief description highlighting key characteristics of the stages is provided below; each characteristic corresponds to a goal in the implementation model. Section B.3 documents the complete implementation model.

- *Opportunistic.* Individual projects develop a reuse strategy (i.e., how reuse will be practiced in their project). The project staff support the reuse strategy, and resources are committed to enact the strategy. Reuse activities are defined in the project's software plan. Specialized reuse tools, automated or nonautomated, are used where advantageous to support the defined reuse activities. The product developers identify potential reusable assets throughout the project life cycle; the assets reused could be requirements, designs, code, architectures, systems, etc. Current developer needs are identified and used as a basis for acquiring or developing reusable assets. Commonalities between needs are identified and used to target assets for multiple use.

- *Integrated.* A standard reuse process is defined and integrated with the organization's standard software development process. The organization's structure, policies, procedures, etc., support the standard process. Tools are tailored to support the standard process. The management and staff are actively involved in defining and implementing the standard process. Anticipated developer needs are identified and used as a basis for acquiring or developing reusable assets. Commonalities between assets and between architectures of assets are identified and used to develop adaptable assets and architectures of assets to meet multiple needs.

- *Leveraged.* A product line reuse strategy is developed to maximize the benefits of reuse over sets of related products, and product pricing and funding strategies take into account the expected costs and benefits over the product line. Performance of the standard reuse process is measured and analyzed to identify weaknesses, and plans are established to address the weaknesses. Reuse tools are integrated with the software development environment. Current customer needs are identified and used as a basis for acquiring or developing reusable assets. Transformation relationships between needs and their corresponding solutions are identified and used to enable broad spectrum reuse (Barnes and Bollinger 1991) where reuse of early life-cycle assets result in the reuse of subsequent life-cycle assets without the need for additional analysis.

- *Anticipating.* Management creates new business opportunities that take advantage of the organization's reuse capability and reusable assets. High payoff assets are identified. Customers' needs are anticipated and used as a basis to acquire or develop reusable assets to meet those needs. New technologies are identified that will meet or drive customer needs and are inserted into the organization's product lines. The effectiveness of reuse technologies is measured and used to determine the most effective technology for a given situation. The organization's process has sufficient flexibility to rapidly adapt to new product environments.

  For example, to reduce risk with respect to the needs identification goals, direct ties of the reuse program to customer needs are deferred to later stages to minimize the consequence of failure (failure to meet a customer's need due to unsuccessful reuse would likely be more costly to the organization than failing to meet a developer's need). Also, the ability to anticipate needs is deferred over identification of the current needs because it requires greater understanding of the customers and technology and how they change.

B.2 REUSE CAPABILITY ASSESSMENT MODEL

The reuse capability assessment model is a mechanism to be used by an organization to gain an understanding of its current reuse capability and to identify potential opportunities for improving its capability. The assessment model consists of a set of critical success factors that correspond to issues most critical to improving reuse capability. The critical success factors are defined in terms of one or more goals. The goals identify the results to be achieved to satisfy a critical success factor.

The set of critical success factors and their corresponding goals are provided below. The factors are organized into four groups: application development, asset development, management, and process and technology factors. Supplemental information is also provided with each factor. This information is intended to help you understand the factors and goals, as well as give you ideas on how to satisfy the goals. This information is not intended to be exhaustive.

B.2.1 Application Development Factors

Asset Awareness and Accessibility

An organization's targeted reuse opportunities can be achieved and assets reused only when the application developers are aware of the assets' existence, can find the assets that meet their needs, and can get access to the assets so they can understand, evaluate, and reuse the assets.

Goals

AA-1. Developers are aware of, can find, and have access to any relevant reusable assets and external sources of assets.

AA-2. Developers are aware of and reuse assets that are specifically acquired or developed for their application.

Supplemental Information

Reuse libraries are usually the first things that come to mind when you consider these goals. Reuse libraries do provide mechanisms to help you find and gain access to available assets. Reuse libraries are most useful when you have a large number of assets to manage and the assets are not targeted to specific applications; i.e., they are provided for general use.

Alternative mechanisms to reuse libraries include catalogs, file directories, databases, configuration management systems, application generators, and reuse advocates. When assessing these goals, you are not making a judgment as to whether you are using the most sophisticated mechanism. Rather, you should judge whether your mechanisms sufficiently support the awareness and accessibility of the assets.

The assets and sources of assets that you should view as relevant are those assets that may be reused to fill your development needs. This not only includes assets developed by your organization but may also include assets developed by other organizations, assets from previously developed systems, external libraries, and commercial products approved by your organization.

When organizations invest in the development of reusable assets with the intent that the assets be used in specific applications or class of applications, the application developers who are developing those targeted applications

need to be aware of these assets. The application developers also have an obligation to make a best effort to use these assets so the investment can be recovered. The application developers should work with the asset developers to ensure that the assets meet their needs, rather than arbitrarily reject the assets.

Asset Identification

A necessary activity in any reuse process supporting application development is the identification of candidate reusable assets. The keys to achieving a higher reuse capability are "when" and "what" assets are identified. Potentially reusable assets should be identified throughout the product life cycle. Preferably, assets are identified that result in the reuse of the corresponding later life-cycle assets. Optimally, the highest payoff assets are identified and reused.

Goals

AI-1. Developers identify potential reusable assets in each major activity in the development life cycle that might satisfy their development needs.

AI-2. Developers identify potential reusable assets before establishing constraints that limit reuse opportunities.

AI-3. Developers identify and reuse early life-cycle assets that result in the reuse of corresponding later life-cycle assets.

AI-4. Developers identify and reuse designated high-payoff, reusable assets.

Supplemental Information

Reuse opportunities exist throughout a product's life cycle. For instance, during a requirements analysis, you may be able to reuse functional specifications; during design, you may be able to reuse a component specification; and during implementation, you may be able to reuse code. Thus, your process should encourage developers to look for reuse opportunities throughout the process. Techniques for ensuring developers address reuse include walkthroughs, inspections, and reviews.

As a development proceeds and you begin to converge on a concrete solution, you begin to lose flexibility in posing alternate solutions, limiting your reuse opportunities. By looking for your reuse opportunities early in the process, you have the ability to pose solutions based on your reusable assets. However, if you wait too long, say until after a design has been agreed to, your reusable assets may not work with the agreed design. A good practice is to look for your opportunities one phase early. For instance, you should consider your opportunities for reusing designs while you are working on the requirements.

Ultimately, your greatest leverage comes when you not only reuse an early life-cycle asset, such as a function, but also any derived assets, e.g., the corresponding design, code, and test cases. In the case of application generators, the identification of the derived assets may be done automatically.

Not all reuse opportunities are created equal; some will have a higher payback than others, and some may have no payback at all (e.g., it may cost more to

adapt an asset than to build a new one). As an application developer, you should consider the costs and benefits of reusing an asset so you can reuse those assets that will have a high payoff. Asset developers should make the application developers aware of those assets that are expected to have a high payoff (e.g., assets that require scarce specialty knowledge).

Asset Evaluation and Verification

Too often, organizations commit to reusing assets before the assets are evaluated, only to find later that the assets cannot be reused. This increases the cost of reuse (wasted effort) and reduces efficiency. Minimally, candidate reusable assets should be evaluated to determine whether the assets meet the specified needs before committing to reuse the assets. In addition, assets that meet a developer's need may not live up to the specification; thus, the quality of an asset should also be verified before its selection.

Goals

AE-1. Developers understand and evaluate reusable assets against their needs before committing to reuse an asset.

AE-2. Developers verify asset quality before selection.

Supplemental Information

To avoid surprises, you should evaluate any potential reusable assets. First, you should check to see if the asset is a close fit for your need; be careful with assets that go well beyond your need and with assets that require a stretch of the imagination to see how it meets your need. Preferably, you should inspect the assets before you make your decision. However, this may not always be possible (e.g., you may have to pay for the asset before you can use it). In these cases, you may have to rely on the asset description or specification.

You should also verify the quality of an asset to ensure that it meets your quality requirements. This may require that you test the asset (e.g., in the case of code). If that is not feasible before selection, then you should at least verify the quality before integration with your application. As you gain confidence in the quality of the supply of reusable assets, your need to test the assets may diminish. Some organizations certify their reusable assets to indicate their degree of confidence in an asset.

In the case of application generators, the generator may automatically select the assets that meet your specification whose quality has been verified in advance. However, you may still need to verify the quality of the generated software in your context.

Application Integrability

A key contributor to the cost of reusing assets is the integration of those assets into the application. In addition to the adaptability of the asset (see the "asset reusability" factor), these costs are affected by the integrability of the application in which the asset is to be reused. For example, the use of unstructured, ad hoc techniques results in less integrable applications versus the use of methods that apply the principles of information hiding and separation of concerns.

Goal

AN-1. Technologies applied in developing applications (i.e., processes, methods, and tools) facilitate the integration of reusable assets.

Supplemental Information

Developing applications that facilitate the integration of reusable assets does not require that you use any specific technology. For instance, it is not required that you use object-oriented techniques or high-level languages to have a successful reuse practice. Rather, consistent use of software engineering principles, such as information hiding, abstraction, separation of concerns, etc., are what is needed to facilitate integration. Object-oriented techniques, for example, embody some of these principles, thus facilitating their use, but not guaranteeing their use.

Whether you use like processes and methods when developing applications and reusable assets will greatly affect ease of integration. For instance, if you use an object-oriented technique to develop your reusable assets and you use structured techniques for developing applications, you can expect to have some difficulty with integration.

To facilitate integration does not require you to throw out your existing techniques in favor of new ones. Rather, you can improve integration by improving on the way you use your existing techniques.

Other concepts or techniques that can facilitate integration include standard interface specifications, layered architectures, program families, parameterization, macros, utilities, and templates.

B.2.2 Asset Development Factors

Needs Identification

If an asset is to be reused, then it must satisfy a need. To ensure reuse, these needs must be identified. There are two distinct classes of needs: customer and developer. There is also a temporal factor to consider; i.e., there are current needs and future needs. Identification of all of these needs is necessary to identify the best reuse opportunities.

Goals

NI-1. Current developer needs for solutions are identified.

NI-2. Anticipated developer needs for solutions are identified.

NI-3. Current customer needs for solutions are identified.

NI-4. Anticipated customer needs for solutions are identified.

NI-5. Identified needs are used as a basis for acquiring or developing reusable assets to meet the specified needs.

Supplemental Information

In assessing these goals, you are trying to ascertain the extent to which you understand your full set of developer and customer needs and to use this understanding to identify high-payoff reuse opportunities. For example, if you focus on your current project needs (i.e., projects that are in progress or planned), you can look for commonality across these needs and provide reusable assets to meet these common needs. This greatly increases the likelihood that the reusable assets will be used on the current projects.

However, if you do not consider what might be needed in future projects (i.e., unplanned), then there is a risk that the reusable assets may not be reused in the future projects. In this case, you may not recover your investment in the reusable asset unless you were able to recover it on your current projects.

To illustrate the distinction between developer versus customer needs and current versus anticipated needs, assume your company develops word processing, spreadsheet, and database products. Examples of current customer needs include: the ability to write letters, reports, and envelopes; the ability to manage mailing lists; the ability to track accounts payable and accounts receivable; etc. Examples of current developer needs include the ability to manipulate text and graphics, manage files, perform mathematical operations, etc. The customer needs systems to aid in performing its mission, whereas the developer needs resources to produce the systems that will meet the customer's needs.

Examples of anticipated customer needs might include the ability for multiple users to simultaneously work on a document from different geographic locations or the ability to store photographs, video, and music. Anticipated developer needs may then include the ability to manage multiple workspaces or the ability to read or write analog signals. Note that what makes these "anticipated" developer needs is that they are not in your current plan. However, your competitor may already be implementing these features; this is one way you can anticipate your needs.

Sources or techniques for identifying needs include marketing reports, trade journals and conferences, customer and developer surveys, strategic plans, and domain analysis.

Asset Interface and Architecture Definition

One step toward reducing the cost of using reusable assets is to reduce the effort required by the application developer in analyzing the internal behavior of the asset. This requires the relationship of an asset to its external environment to be defined (its interface). Then, by defining the relationships between assets in the same major activity of development (e.g., requirements analysis, design, etc.), larger scale reuse of architectures is possible (e.g., requirements architectures, design architectures, etc.).

Goals

AD-1. The relationships of an asset to its external environment are defined.

AD-2. The relationships between assets within the same major activity (e.g., requirements analysis, design, etc.) of the development life cycle of a product are defined.

Supplemental Information

An asset's interface defines its relationships with its external environment (such as other software assets and hardware). This interface not only includes the data passed to or from the asset but also includes the function performed by the asset, its performance characteristics, limitations, and assumptions about its environment. Explicitly defining this information greatly improves the ability of the potential reuser to identify, understand, evaluate, and use the reusable asset.

An architecture defines the relationships among a set of assets; this can include data flow, sequencing, prioritization, timing, etc. Architectures exist at all levels of development: a requirements architecture can capture relationships between functions (e.g., data flow and control flow), a design architecture can capture relationships between design elements (e.g., intertask communication), and an implementation architecture can capture relationships between packages (e.g., procedure calls). By explicitly defining these relationship, you provide the potential reuser with not only the ability to reuse a single asset but possibly a whole collection of integrated assets.

Techniques that can assist you in defining interfaces and architectures include asset and architecture documentation, reuse libraries, databases, and modeling.

Needs and Solutions Relationships

A reuse opportunity is an occasion where a solution (asset, architecture of assets) satisfies a need. The definition and use of these relationships has the potential to reduce costs substantially. These relationships define the transformation from a need (e.g., requirement) to a solution (e.g., design) and enable the acquisition or development of broad-spectrum assets where reuse of early life-cycle assets results in the reuse of the corresponding later life-cycle assets while reducing the need for in-depth analysis of the later life-cycle assets.

Goals

NS-1. Transformation relationships between needs and their corresponding solutions are defined (e.g., from requirements to design, from design to code, etc.).

NS-2. Identified transformation relationships between needs and solutions are used as a basis for acquiring or developing broad-spectrum assets.

Supplemental Information

The Asset Interface and Architecture Definition factor addressed relationships between assets within a given activity of your process (e.g., relationships within requirements, within design, etc.). This factor addresses the relationships between assets across the activities of your process. For example, for a set of requirements, there may be a corresponding set of test cases for verifying the implementation of those requirements. By identifying the mapping between the requirements and the test cases, you provide the potential reuser with the ability of reusing the test cases as well as the requirements.

The collection of related, cross-process assets is referred to as a broad-spectrum asset. An extreme example of a broad-spectrum asset is a complete product. If you can use an existing product to satisfy a customer's need that otherwise would have required new development, then the customer saves the expense of development and you gain additional revenue on work already completed.

Commonality and Variability Definition

To develop and reuse assets for multiple contexts requires that commonalities and variabilities between needs and between assets over multiple contexts be defined.

Goals

CV-1. Commonality and variability relationships between needs are defined and used to acquire or develop assets to meet multiple needs.

CV-2. Commonality and variability relationships between solutions (assets, architectures of assets) are defined and used to acquire or develop adaptable solutions for meeting multiple needs.

Supplemental Information

Commonalities (or similarities) between needs and between solutions is what makes reuse possible. The greater the degree of commonality between needs, the greater the likelihood a common solution can be developed to meet those needs. Recognizing these commonalities as well as the variabilities (i.e., how they differ) is essential to identifying and exploiting your reuse opportunities.

Recognizing the commonalities between needs can help you in several ways. First, if you can define the commonalities between a new need and a prior need with an existing solution, then you can identify the existing solution as a candidate for reuse. Further, if you can define the variabilities between the two needs, then you can more readily determine how to adapt the existing solution to your current need.

Second, if you can define the commonalities between multiple unmet needs, then you can develop a common solution. Further, by defining the variabilities between the needs, you can develop a common solution that is adaptable to the specified variabilities.

Lastly, by determining the extent of commonalities and variabilities between needs, you can make a judgment as to whether there is sufficient reuse potential to justify development of a reusable asset.

Defining the commonalities and variabilities between solutions (existing and proposed) can guide you in developing common, adaptable solutions, thus reducing your need to support multiple solutions. For example, if you find that your applications are using five different storage managers and there is a high degree of commonality with some minor variabilities, you could possibly replace the five storage managers with a single common storage manager that is parameterized to satisfy the variabilities.

Defining and managing commonalities and variabilities is a key activity in most domain analysis or engineering approaches.

Asset Value Determination

An organization is limited in the amount it can invest in reuse; thus, it may not be feasible to exploit all potential reuse opportunities. To improve its reuse efficiency requires the organization to focus its investment where it will get the best payoff. This requires that the net value (benefits less costs) of an asset be determined.

Goals

AV-1. The net value of an asset is estimated based on experience data and identified needs.

AV-2. The estimated net value of an asset is used as a basis for focusing the organization's resources on acquiring or developing high-payoff assets for reuse.

Supplemental Information

This factor does not imply that you must precisely quantify the costs and benefits of reuse. Rather, in assessing your process, you should judge whether there is sufficient guidance supporting make versus adapt versus design for reuse decisions and judge the extent to which this guidance is applied when making decisions on reuse.

A simplistic heuristic used by some organizations to support decisions on reuse in known as "the rule of three." That is, if you can expect at least three uses from an asset, then it is worth the cost of designing the asset for reuse. Other organizations assume that it is always better in the long run to design assets for reuse. Usually these organizations are developing the asset in conjunction with a product development and pass the cost of designing the asset for reuse directly onto the customer. Their rationale is that assets designed for reuse also reduce life-cycle maintenance costs.

Other techniques supporting make versus reuse decisions include reuse economics models and cost estimation models.

Asset Reusability

The adaptability of an asset relates directly to the costs of reusing (particularly integrating) the asset with applications. The costs can be reduced by applying methods and tools that facilitate the adaptation of reusable assets for use in multiple contexts.

Goal

AR-1. Technologies applied in developing reusable assets facilitate their integration into applications.

Supplemental Information

In general, reusable assets are intended to be used in multiple contexts to save the costs of developing new assets. However, the more it costs to adapt a reusable asset to serve in each of these contexts, the less you save if you develop new assets.

Questions to consider when assessing this factor include:

- Are common processes and methods used for developing reusable assets like those used for developing applications?
- Do the asset developers consistently apply principles, such as information hiding, separation of concerns, encapsulation, and abstraction, when developing reusable assets?
- Are methods used to specify interfaces, structure, behavior, and variabilities of the reusable assets?

Other concepts or techniques that can facilitate reusability include standard interface specifications, metaprogramming, parameterization, macros, utilities, and templates.

Asset Quality

The reuse of low-quality assets can result in high life-cycle costs due to defects, resulting in a higher utilization cost and lower efficiency. This problem is magnified when a low-quality asset is used in multiple contexts. To improve reuse efficiency requires that asset quality be controlled and managed.

Goals

AQ-1. Reusable assets are under configuration control.

AQ-2. Feedback on reusable assets is collected and used to maintain and enhance the reusable assets.

AQ-3. Sufficient data is provided for an application developer to understand, evaluate, and apply reusable assets.

AQ-4. Reusable assets are verified against their specifications.

AQ-5. Reusable-asset quality goals or standards are established and tracked.

Supplemental Information

Like any other software you develop, the quality of reusable assets is a key concern. Also, like any other software you develop, the quality mechanisms you employ for reusable assets can be the same mechanisms you use to ensure the quality of newly developed software, such as inspections, reviews, testing, standards, quality assurance groups, engineering change proposals, etc.

The key difference is the broader impact that the quality of a reusable asset has due to multiple uses. For example, if you change a software package on your current project, you must make sure other members of your team whose software interfaces with your package are aware of the change. In the case of a reusable asset, you may also need to make sure that other users of this asset are aware of the change. This is particularly important in environments where only the latest version of a reusable asset is maintained and all product upgrades are built using the latest version (e.g., operating system service libraries).

Questions to consider when assessing this factor include:

- Are the assets baselined? Are changes controlled?
- Is there a mechanism for reporting defects and suggested enhancements?
- Is essential information on an asset provided (its function, limitations, relationships to other assets, history, level of certification, cost, usage, etc.)?
- Are the asset descriptions verified against the asset for accuracy?
- Is the quality of the assets certified?
- Are reusable asset standards established (e.g., design for reuse standards) and is there a mechanism to ensure compliance with the standards?

B.2.3 Management Factors

Organizational Commitment

The success of an organization's reuse process starts with the organization's commitment (both management and staff) to define, implement, and improve

its reuse process. Without this commitment, the organization cannot expect to exploit its potential reuse opportunities. Minimally, management must support the organization's reuse efforts, such as providing resources. However, achieving a higher reuse capability will require a greater commitment and more active role by management.

Goals

OC-1. Management commits to defining, implementing, and improving the organization's approach to reuse and demonstrates its commitment to the staff.

OC-2. Management commits funding, staffing, and other resources to define, implement, and improve the organization's approach to reuse.

OC-3. The staff supports the organization's approach to defining, implementing, and improving the organization's approach to reuse.

OC-4. Management structures its organization, policies, procedures, and standards to facilitate a standard reuse process supporting multiple product development efforts.

Supplemental Information

Management in this context refers to executive management, senior management, functional area management, and line management. To succeed at reuse requires management's support, both in words and deeds. Management can show its support by:

- Committing to investigate its reuse capability and taking action on the results.
- Communicating its motivation and support for improving the organization's reuse practice to the staff and soliciting the staff's support.
- Actively overseeing the reuse improvement activities and reenforcing its commitment throughout the improvement process.
- Committing the staff capable of addressing reuse issues and the time necessary to implement reuse.

In addition to management's support, the staff must support the reuse effort as well. The staff can show their support by regularly attending scheduled reuse meetings, participating in the development and implementation of a reuse strategy (at least providing input and feedback), and executing the strategy when selected.

Achieving higher levels of reuse may necessitate changes to the organization's structure, policies, procedures, and standards. Specific changes will depend on the given strategy but could involve the creation of new organizations, the consolidation of similar projects, changing roles and responsibilities, etc. These types of changes require a greater commitment from management and the staff.

Planning and Direction

To successfully move an organization toward achieving its targeted reuse opportunities with any consistency requires planning at all levels of the organization. Management should have a strategy for achieving its reuse objectives at the product, product line, and business levels. The product manager must ensure that targeted reuse opportunities within his product are met. The product line manager must ensure that similar sets of products are coordinated to maximize the realization of potential opportunities over the set. The business manager must build the reuse capability of the organization and create new opportunities that leverage that capability.

Goals

PD-1. Reuse strategies are defined for product development efforts in support of product objectives, and product development activities are planned and directed in accordance with the product development's reuse strategy.

PD-2. Product line reuse strategies are developed to maximize the benefits of reuse over sets of related products, and the product line activities are planned and directed in accordance with the strategies.

PD-3. Management develops a long-term strategy for improving the organization's reuse capability, and the organization's improvement efforts are planned and directed in accordance with the improvement strategy.

PD-4. New business opportunities are created that take advantage of the organization's reuse capability and reusable assets.

Supplemental Information

The product (or project or program) manager's primary objectives are to meet the customers' needs on time and within budget. In doing so, the product manager must determine how best to apply the organization's resources to develop a quality product. The product manager should take reuse into account when developing this strategy, i.e., how to leverage the organization's resources by exploiting commonalities within the product and with other products.

In this context, a product line is a set of similar products. A product line manager is an individual or group responsible for coordinating the individual product developments. The product line manager must balance the needs of the individual customer with the business objectives of the organization. Primarily, the product line manager must determine how to apply the organization's resources to best meet all of the customers' needs. Again, reuse should be a key element of the product line manager's strategy in exploiting commonalities across a product line.

The product line manager's responsibilities could include overseeing tradeoffs (i.e., determining what is best for the product line versus what is best for an individual product), increasing commonality across the products in the product line, eliminating incidental variabilities, establishing policies, setting priorities, and establishing standards.

In this context, the business manager refers to those individuals responsible for the overall success of the organization, keeping the organization in business. This involves building the organization's capabilities to effectively meet customer needs and exploiting the organization's capabilities to attract new customers. Reuse can have a key role in supporting the organization's business success.

For reuse to have a role in an organization's business planning requires that reuse be addressed in the organization's business strategy. An example of strategies that a business organization might take include introducing the organization's products into new markets where there is sufficient commonality between the needs addressed by the organization's products and the needs of the new market area. Another approach is to introduce innovations on the organization's products to create a new demand. In both of these cases, the objective is to create new opportunities where existing products and capabilities can be exploited.

Costing and Pricing

To achieve a higher reuse capability requires the organization to invest in reuse. This investment should be based on a product pricing and funding strategy that supports the organization's reuse strategy. These strategies need to take into account the expected costs and anticipated benefits of reuse over the organization's product lines. Minimally, procedures must be established to determine who pays for the development of reusable assets, who pays for the use of reusable assets, and how these costs are determined.

Goals

CP-1. Reuse cost accounting procedures are defined and enacted.

CP-2. Management establishes a long-term plan for funding reuse activities and improvement efforts.

CP-3. Product pricing and funding strategies take into account expected costs and anticipated benefits of reuse over the product or product line.

Supplemental Information

Reuse requires an investment to make reuse happen. It requires an investment in improving the capabilities of your organization, in establishing an infrastructure for reuse, in the creation of reusable assets, and in the use of reusable assets. To ensure a fair return on your investment, you need to manage the costs of reuse, and you cannot do this unless you know the costs. The types of costs you should consider collecting include the cost to develop and maintain assets for reuse, the costs to adapt reusable assets, the cost to manage the reuse program, and support costs (e.g., training, tools).

The nature of reuse is such that an investment is required up front and the return on investment does not come until later, sometimes much later. Occasionally, you will encounter the attitude that "reuse is supposed to save us money, so we can cut your budget." To avoid this situation, your management needs to be aware of the costs of reuse and they need to plan the investment. That is, they need to estimate the expected costs, identify the source of funds, and include the funding in their budget planning.

In addition to understanding the costs of reuse, management must also understand the expected benefits and take them into account when determining how to recoup their investment. Some possibilities include directly charging the customer for the initial asset development, distributing the cost across customers by amortizing the costs of reuse over the expected number of products to be developed with reuse, charging a fee for use, using internal funds recovered via additional sales revenue, etc.

Legal and Contractual Constraints

Failure to comply with relevant legal or contractual constraints could have significant financial consequences, greatly increasing the cost of reuse and reducing the organization's reuse efficiency. Minimally, an organization must be aware of any relevant legal or contractual constraints and comply with them. In addition, an organization should remove or reduce these constraints (e.g., through negotiation) where feasible, thereby increasing its potential reuse opportunities.

Goals

LC-1. Reuse legal or contractual constraints are identified and enforced.

LC-2. Legal and contractual constraints on reuse are removed or reduced to increase the potential for reuse when feasible.

Supplemental Information

Legal and contractual obligations can limit your reuse opportunities. They can affect your ability to reuse assets and other organizations' ability to use your assets. For example, in a DoD contract, the government usually will have unlimited rights to the software developed by the contractor. This means that the contractor cannot charge the government again if that software is reused and the government could provide the software to other contractors. Consequently, this could limit the contractor's ability to recoup its investment in reuse.

Government contracts are not the only source of potential constraints. Legal and contractual constraints can also arise from joint ventures, use of off-the-shelf products, subcontracting, etc.

To avoid violations, your organization needs to be aware of any legal or contractual constraints that may affect your reuse strategy and have mechanisms in place to ensure compliance with any constraints.

Your organization can also take actions to remove constraints and thereby increase your reuse opportunities. For instance, in the example described above, a contractor can develop a reusable asset with its own funds and not in conjunction with any government contract. In this case, the government would only have restricted rights to the asset and the contractor could then reuse the asset again and charge the government.

Intergroup Coordination

For an organization to fully exploit its reuse opportunities requires that application developers have the reusable assets to satisfy their needs when they need them and that asset developers understand the application developers' needs and environment. Meeting both of these conditions requires coordination between the asset and application developers.

Goal

IC-1. Application and asset development activities are coordinated between and among application and asset development groups to identify, track, and resolve intergroup issues.

Supplemental Information

Your organization may or may not have a separate group to develop reusable assets. A common reuse practice is for application developers to contribute assets for reuse from their current development efforts. In this case, the individual engineer is both an application and asset developer; thus, coordination becomes moot. However, when the output of one application developer is expected to become the input for other application developers, the need for coordination grows. If your organization has a separate group for asset development, then coordination is essential if the group is to function effectively.

The objective of the coordination is to identify, track, and resolve intergroup issues. For example, schedule is usually a critical issue, particularly when the application developers are relying on the asset developers for an asset that is still in development.

Technical interchange meetings, joint reviews, joint working groups, integrated product teams, and written agreements are a few mechanisms you can use to facilitate intergroup coordination.

B.2.4 Process and Technology Factors

Process Definition and Integration

The lack of a defined process results in unpredictable performance and wide variations in reuse proficiency and efficiency; a defined process provides consistency. Minimally, the process can be defined for a product in the product plan. Preferably, a standard process is defined for a product line to increase the commonalities between the process artifacts and thus increase the potential for reuse. Optimally, the process should have the flexibility to support the organization in creating or exploiting new opportunities.

Goals

PI-1. Reuse activities and resources are identified in the product software development plan.

PI-2. Standard reuse processes are defined and integrated with the organization's standard software development process.

PI-3. Standard reuse processes provide sufficient flexibility for tailoring the standard processes to the unique needs of the product development efforts.

PI-4. The standard reuse processes have sufficient flexibility to adapt to new product environments (markets).

Supplemental Information

A product development plan represents the definition of the process for developing or maintaining a product. It identifies what tasks will be performed, who will perform the tasks, when the tasks will be performed, and

the artifacts to be produced. For reuse to be a part of your process, it needs to be included in your plan. If reuse is not in your plan, then you have no assurance that your reuse opportunities will be addressed.

A standard process for an organization helps ensure that each project conducts its activities in a consistent manner. It is also the basis or starting point for defining the individual product development plans. Thus, integrating reuse into your standard development process will help ensure that each project will address reuse and do it in a consistent manner. This has an added benefit with respect to reuse in that a standard process increases the commonality between the projects. With a standard process, they will be producing the same types of artifacts (e.g., requirements specifications, design specifications, etc.) in a similar form. Increasing commonality increases the potential for reuse.

Unless you want your standard process to be "dead on arrival," you need to provide the individual projects with some flexibility in applying the standard. The product manager needs to determine how best to apply the standard in meeting the customer's needs; this may require deviations from the standard. For example, you may have a customer that requires the product to be developed using an architecture supplied by the customer (e.g., the Army Command and Control System Common Application Software). If your standard process is built around your own common architecture, then the project manager will likely need to deviate from the standard in this case.

To ensure that any tailoring of the standard is done in an appropriate manner; i.e., the project's needs for flexibility are weighed against the organization's needs for a common process, you can provide guidelines on how to tailor the process and establish a mechanism for approving a project's tailoring of the process.

To support your organization when expanding into new markets or creating technology innovations may require even greater flexibility in applying a standard process. You do not want the standard process or reuse to unnecessarily limit creativity and innovation. There are times when you may want to take nonstandard approaches or disregard your reusable assets to advance technology. It takes a mature organization to recognize when this is necessary and to practice this without corrupting what you are trying to accomplish with the standard process.

Measurement

The collection, analysis, and use of data are critical to reuse on several fronts. It is critical to tracking progress toward reuse objectives. It is critical to accurate estimation of reuse costs and benefits. It is an essential part of continuous process improvement. It provides input to the identification of high-payoff assets, and it provides the basis for determining the effectiveness and efficiency of reuse technologies.

Goals

MS-1. The impact of reuse on cost, schedule, and product is estimated during development planning; then actual impacts are recorded as the development proceeds.

MS-2. Reuse experiences from past and current projects are collected and made available.

MS-3. The effectiveness and efficiency of reuse technologies are measured and used as a basis for determining the most suitable technologies for given situations.

Supplemental Information

Understanding the impact of reuse on cost, schedule, and product (e.g., functionality, performance, and quality) is necessary to develop a valid plan. You can gain this understanding by estimating the impacts based on the information at hand. Then compare your estimates to the actual impacts as your project proceeds. Based on this comparison, you can then adjust your plans accordingly and use the experience to improve future estimates, thereby continually improving the predictability of your process.

Measurement is also a basis for learning and continual process improvement. This should include not only quantitative data, such as cost and schedule, but also qualitative data, such as lessons learned. This data needs to be collected and made available to others so they can learn from past experiences.

Measurement can also help you make decisions on what technologies to use, when to use them, and how to improve them. You do not want to use the most expensive technology when a less expensive technology can do the job. For example, one way to explicitly implement variabilities in a software product is to duplicate the product for each variability (e.g., Booch parts). Another way is to develop a single abstract component that embodies the variabilities and allows the concrete component to be extracted via instantiation (i.e., metaprogramming). It may be inefficient to duplicate a component many times when there are many variabilities. Likewise, it may be inefficient to develop an abstract component when there are very few variabilities. Understanding the costs and effectiveness of these technologies can help you choose the right technology for the right situation.

Continuous Process Improvement

A defined process will provide consistency to an organization's reuse performance, but the actual level of performance may still be unsatisfactory. To increase the level of performance requires a program of continuous process improvement through measurement and refinement.

Goals

CI-1. Performance of the standard reuse processes is measured and analyzed to increase understanding and identify strengths and weaknesses.

CI-2. Plans are established to systematically address weaknesses identified in the standard reuse processes.

Supplemental Information

Improving your process begins with understanding the current state of your process, knowing your strengths, and identifying your improvement opportunities. The RCM and assessment approach are mechanisms to support you in gaining this understanding. However, that does not mean that you must use this model and assessment to satisfy these goals.

Measurement also supports continuous process improvement. For example, suppose that, on the average, 50% of your reuse library is applicable to each product and that, on the average, you are using only 25%; this implies that you are 50% efficient (25/50 = 0.5). This result could indicate a problem with your reuse library or your process; additional investigation would be required to determine the actual cause.

You may also assess your reuse capability in the context of a broader program, such as TQM and the SEI Software Process Assessments.

When you do assess your reuse capability and identify needed improvements, you must then take action on those improvements. Failing to do so runs the risk that the organization will become disenchanted and be unwilling to support future improvement activities.

Training

Achieving a higher reuse capability will likely involve the application of advanced reuse processes, methods, and tools. It will also likely constitute a significant change in the way an organization does its business. For these reuse technologies and changes to succeed, the organization must ensure that management and staff have the skills and knowledge necessary to effectively apply its reuse technologies.

Goals

TR-1. The knowledge and skills necessary to effectively apply an organization's reuse technologies are determined, gaps in the staff's knowledge and skills are identified, and a plan is developed and enacted to fill the gaps.

TR-2. The effectiveness of training to fill the gaps in necessary reuse technology knowledge and skills is measured and analyzed to identify weaknesses.

TR-3. Plans are established to systematically address the weaknesses identified in reuse technology training.

Supplemental Information

The key concern with this factor is that your organization has the knowledge and skills to carry out the reuse strategy you have chosen. This includes both process and product knowledge. Although this factor is named "training," there are many ways your organization might use to fill any gaps, such as classroom training, on-the-job training, videos, manuals, mentors, consultants, subcontracting, new hires, etc.

To ensure that your organization has the necessary knowledge and skills requires not only that you provide training but that you provide the training

that is needed when it is needed. Avoid the trap of providing the training you know how to give rather than the training you know is needed. An effective time to provide training is just before it is needed, known as just-in-time training.

In addition to providing training, you need to know whether the training is effective in meeting the organization's needs. A course evaluation questionnaire is an example of a mechanism for measuring this effectiveness. If you identify any shortcomings, you should take action to improve the training.

Tool Support

The use of tools, both automated and nonautomated, when applied appropriately can reduce the costs of creating and reusing assets. Achieving this requires that any reuse tools work with the process and software development environment, not against it. Reuse tools should also only be applied where there is a payoff; i.e., an expensive tool should not be acquired or developed to perform a low-cost task when an inexpensive approach can do the job.

Goals

TS-1. Reuse tools are used to support defined reuse activities and methods for which they are effective.

TS-2. Reuse tools are acquired, developed, or tailored to support the standard reuse processes.

TS-3. Reuse tools are integrated with the organization's software development environment.

TS-4. Reuse tools provide sufficient flexibility to adapt to new process or product environments.

Supplemental Information

Tools in this context refer to the mechanisms you use to assist in performing a task. Tools may be automated (e.g., reuse libraries, CASE) and nonautomated (e.g., checklists, handbooks). You may have your own "home grown" tools or commercial tools. Tools can greatly improve your efficiency and the reliability of your work. Tools are good for performing tasks that are complex, tedious, and time consuming.

However, if you are not careful, the tools you use can lock you into an inappropriate process. Many tools still have one way of doing things that is not necessarily designed for your tasks. To mitigate this risk, you should avoid a big investment in tools (especially tools that are not flexible) until you have had a chance to mature your process. Avoid tools that might dictate your process or lock you into an immature process. You can begin by using tools to support well-defined tasks (as opposed to a complete process) where there is a clear payoff. For example, you can use your existing configuration management tools to control your reusable assets. This minimizes your investment in new tools and supports a critical function of reuse.

When your understanding of reuse and your process matures, you can then develop or acquire tools to support a greater part of your process (i.e., a set

of interrelated tasks or activities). As this occurs, there will be a greater need to integrate the tools to allow data to be shared, to provide a common control mechanism, and to provide a common presentation, ultimately leading to a seamless environment.

As your use of tools grows, you will still need to retain some flexibility to continue to support changes in the product environment (e.g., advances in technology, expansion to new markets) and changes in your process.

Technology Innovation

Technology continues to advance, resulting in new demands for products. An organization needs to be aware of and use technology to its advantage. Advances in technology may affect existing assets. An asset based on an old technology has less value when a new technology comes to market, creating the need to update or liquidate the asset. An organization can also use new technology to drive customer needs by inserting new technologies into its reusable assets.

Goals

TI-1. Management and staff are aware of new and evolving technologies and standards that may affect their products and reusable assets.

TI-2. New technologies are identified that will meet or drive customer needs, have a clear benefit, and take advantage of the organization's reuse capability. Selected technologies are inserted into the organization's process or product.

Supplemental Information

If you are not careful, changes in technology can make your reusable assets obsolete. For example, a number of organizations had inaccurately assessed the impact of personal computers on the software industry. As a result, they did not position their products to exploit these opportunities, only to find later that they had lost an important segment of their market. The same thing can happen with reusable assets.

You can design your reusable assets to lessen the impact of technological change, but you also need to stay abreast of the technological changes so you can respond appropriately. This includes monitoring or participating in standards development efforts.

You can also use technological change to your advantage. By inserting new technology into your reusable assets, you can increase the demand for your assets. For example, suppose you have a line of office automation software available on personal computers. You may be able to reuse this software by adapting it for the newly emerging hand-held technology. If successful, you will get more value out of your existing software, get a jump on a new technology, and increase the demand for your software as the demand for hand-held products grows.

B.3 REUSE CAPABILITY IMPLEMENTATION MODEL

The reuse capability implementation model is a mechanism to be used by an organization to establish near- and long-term improvement goals (reuse adoption goals) based on the understanding gained by the organization of its reuse capability via the assessment model.

The implementation model constitutes a prioritization and partitioning of the critical success factor goals into stages. The stages represent possible operational steady states in an organization's reuse practice, and successive stages build on practices established at earlier stages. This section provides an initial implementation model based on a risk-reduction growth scheme for prioritization and partitioning. Alternative schemes could be used to define alternative implementation models.

The risk-reduction growth implementation model includes four stages: opportunistic, integrated, leveraged, and anticipating. Tables B-2 through B-5 provide the mapping of the critical success factor goals to the four stages, respectively.

Table B-2. Opportunistic Stage

Critical Success Factor Goals			
Application Development	**Asset Development**	**Management**	**Process and Technology**
AA-1. Developers are aware of, can find, and have access to any relevant reusable assets and external sources of assets. AI-1. Developers identify potential reusable assets in each major activity in the development life cycle that might satisfy their development needs. AE-1. Developers understand and evaluate reusable assets against their needs before committing to reuse an asset. AE-2. Developers verify asset quality before selection.	NI-1. Current developer needs for solutions are identified. NI-5. Identified needs are used as a basis for acquiring or developing reusable assets to meet the specified needs. AD-1. The relationships of an asset to its external environment are defined. CV-1. Commonality and variability relationships between needs are defined and used to acquire or develop assets to meet multiple needs. AQ-1. Reusable assets are under configuration control.	OC-1. Management commits to defining, implementing, and improving the organization's approach to reuse and demonstrates its commitment to the staff. OC-2. Management commits funding, staffing, and other resources to define, implement, and improve the organization's approach to reuse. OC-3. The staff supports the organization's approach to defining, implementing, and improving the organization's approach to reuse. PD-1. Reuse strategies are defined for product development efforts in support of product objectives, and product development activities are planned and directed in accordance with the product development's reuse strategy. CP-1. Reuse cost accounting procedures are defined and enacted. LC-1. Reuse legal or contractual constraints are identified and enforced.	PI-1. Reuse activities and resources are identified in the product software development plan. MS-1. The impact of reuse on cost, schedule, and product is estimated during development planning; then actual impacts are recorded as the development proceeds. TR-1. The knowledge and skills necessary to effectively apply an organization's reuse technologies are determined, gaps in the staff's knowledge and skills are identified, and a plan is developed and enacted to fill the gaps. TS-1. Reuse tools are used to support defined reuse activities and methods for which they are effective.

Table B-3. Integrated Stage

Critical Success Factor Goals			
Application Development	**Asset Development**	**Management**	**Process and Technology**
AA-2. Developers are aware of and reuse assets that are specifically acquired or developed for their application.	NI-2. Anticipated developer needs for solutions are identified.	OC-4. Management structures its organization, policies, procedures, and standards to facilitate a standard reuse process supporting multiple product development efforts.	PI-2. Standard reuse processes are defined and integrated with the organization's standard software development process.
AI-2. Developers identify potential reusable assets before establishing constraints that limit reuse opportunities.	AD-2. The relationships between assets within the same major activity (e.g., requirements analysis, design, etc.) of the development life cycle of a product are defined.	PD-3. Management develops a long-term strategy for improving the organization's reuse capability, and the organization's improvement efforts are planned and directed in accordance with the improvement strategy.	PI-3. Standard reuse processes provide sufficient flexibility for tailoring the standard processes to the unique needs of the product development efforts.
AN-1. Technologies applied in developing applications (i.e., processes, methods, and tools) facilitate the integration of reusable assets.	CV-2. Commonality and variability relationships between solutions (assets, architectures of assets) are defined and used to acquire or develop adaptable solutions for meeting multiple needs.	CP-2. Management establishes a long-term plan for funding reuse activities and improvement efforts.	MS-2. Reuse experiences from past and current projects are collected and made available.
	AR-1. Technologies applied in developing reusable assets facilitate their integration into applications.	IC-1. Application and asset development activities are coordinated between and among application and asset development groups to identify, track, and resolve intergroup issues.	TS-2. Reuse tools are acquired, developed, or tailored to support the standard reuse processes.
	AQ-2. Feedback on reusable assets is collected and used to maintain and enhance the reusable assets.		
	AQ-3. Sufficient data is provided for an application developer to understand, evaluate, and apply reusable assets.		
	AQ-4. Reusable assets are verified against their specifications.		

Table B-4. Leveraged Stage

Critical Success Factor Goals			
Application Development	**Asset Development**	**Management**	**Process and Technology**
AI-3. Developers identify and reuse early life-cycle assets that result in the reuse of corresponding later life-cycle assets.	NI-3. Current customer needs for solutions are identified. NS-1. Transformation relationships between needs and their corresponding solutions are defined (e.g., from requirements to design, from design to code, etc.). NS-2. Identified transformation relationships between needs and solutions are used as a basis for acquiring or developing broad-spectrum assets. AQ-5. Reusable-asset quality goals or standards are established and tracked.	PD-2. Product line reuse strategies are developed to maximize the benefits of reuse over sets of related products, and the product line activities are planned and directed in accordance with the strategies. CP-3. Product pricing and funding strategies take into account expected costs and anticipated benefits of reuse over the product or product line.	CI-1. Performance of the standard reuse processes is measured and analyzed to increase understanding and identify strengths and weaknesses. CI-2. Plans are established to systematically address weaknesses identified in the standard reuse processes. TR-2. The effectiveness of training to fill the gaps in necessary reuse technology knowledge and skills is measured and analyzed to identify weaknesses. TR-3. Plans are established to systematically address the weaknesses identified in reuse technology training. TS-3. Reuse tools are integrated with the organization's software development environment. TI-1. Management and staff are aware of new and evolving technologies and standards that may affect their products and reusable assets.

Table B-5. Anticipating Stage

Critical Success Factor Goals			
Application Development	**Asset Development**	**Management**	**Process and Technology**
AI-4. Developers identify and reuse designated high-payoff, reusable assets.	NI-4. Anticipated customer needs for solutions are identified. AV-1. The net value of an asset is estimated based on experience data and identified needs. AV-2. The estimated net value of an asset is used as a basis for focusing the organization's resources on acquiring or developing high-payoff assets for reuse.	PD-4. New business opportunities are created that take advantage of the organization's reuse capability and reusable assets. LC-2. Legal and contractual constraints on reuse are removed or reduced to increase the potential for reuse when feasible.	PI-4. The standard reuse processes have sufficient flexibility to adapt to new product environments (markets). MS-3. The effectiveness and efficiency of reuse technologies are measured and used as a basis for determining the most suitable technologies for given situations. TS-4. Reuse tools provide sufficient flexibility to adapt to new process or product environments. TI-2. New technologies are identified that will meet or drive customer needs, have a clear benefit, and take advantage of the organization's reuse capability. Selected technologies are inserted into the organization's process or product.

This page intentionally left blank.

APPENDIX C. REUSE ADOPTION RISKS

This appendix presents some of the risks associated with reuse adoption. These were derived from: Minutes of the Reuse Acquisition Action Team (RAAT 1992) and State of Reuse Practice Study data (Software Productivity Consortium 1992d). Additional sources are *Taxonomy-Based Risk Identification* (Carr et al. 1993), which addresses general software development risks though not reuse issues, and *Software Reuse Survey Report* (Frakes and Fox 1993), which contains a discussion of obstacles to reuse and their reported frequency of occurrence.

You should use this taxonomy to identify risks. Then you should estimate the probability of occurrence and the magnitude of the consequence. You can then plan actions to avert costly risks. Clues to the aversion strategy can be found in the risk statement itself and in the category under which it is listed. For example, consider a process risk, specifically, the risk that the software development process used by the organization does not address reuse. Clearly, there are at least two possible solutions: introduce a new process that is based on reuse or provide guidance on how and where to append reuse activities to the existing process. Specific details of your aversion strategy depend upon the situational context.

C.1 ORGANIZATIONAL RISKS

C.1.1 Organizational Structure

- Inadequate support for acquiring and managing reusable assets or supporting users can arise when the organization is not set up to manage software reuse.
- Reusable assets can be difficult to find because they are not kept in a specific place or under control of any particular group.
- Reuse activities conflict with the company's standard development process and methodology.
- It is difficult to maintain components because one group is trying to cover too many complex domains.
- Reuse is inefficient because managers do not have sufficient authority.

C.1.2 Organizational Politics

- Existing organizational objectives conflict with the reuse objective being introduced.
- The development staff do not believe reuse will make their jobs more productive, challenging, or rewarding.

- The reuse incentive program may be viewed as a management "stunt" rather than something from which developers can benefit.
- There is poor use of library components because users do not know the library exists, do not believe components are relevant to their needs, do not know how to access them, or do not trust the components.
- Reuse activity is low because it conflicts with the current culture or developers prefer to build their own.
- Reuse activity is low because the sudden imposition of a reuse plan in place of traditional methods is resisted by developers.
- Managers shun reuse because they fear that the organizational change (e.g., assigning asset development to a separate group) will reduce their empire.
- The funding model for asset maintenance may fail because support for reusable assets was reduced before the costs are amortized.

C.1.3 Organizational Capability

- The asset library or development environment is not adequate for development with reuse.
- Policy guidelines fail to promote use of reusable assets.
- Contract terms do not permit the developer to realize the benefits of reuse or protect them if reuse-specific problems arise.
- The opportunity for reuse in current or future projects is lost because inappropriate standards were chosen. An old standard may not have enough potential to make it worthwhile to write reusable software objects but would allow reuse if parts already exist.
- Reuse activity is low because staff is not adequately trained in utilizing reusable assets on hand, recognizing opportunities to apply them, or building reusable assets.
- Poor component access reduces reuse activity and its cost effectiveness. The developer's terminal does not have access to the repository. Terminals that do have access are unfamiliar or inaccessible.
- Developers have difficulty with reuse because support from library staff is poor.
- Inefficient component retrieval results from a lack of abstracts that help a developer quickly determine what the essential features of reusable assets are.
- Difficulty is encountered during component selection because specialized knowledge is needed about the component or its environment in the system.
- Components are difficult to reuse because they were not specifically designed and documented for reuse, perhaps because the component developers were not trained to write reusable components.

- Library content is poorly understood by developers because content is not well publicized.
- Users have difficulty finding components because no search tools are provided.
- Users have difficulty finding components because tools do not support standard library search methods (i.e., hierarchical or keyword search or a faceted classification system).
- Users have difficulty finding components because insufficient support is provided to select components for specialized functions (tutorial or model-based guidance may be needed).
- Excessive effort is devoted to fixing components because no support is available from the library staff or component authors.
- Poor development cost estimates can arise from a lack of cost models or the historical data necessary to use them.
- Reuse activity and efficiency can be inhibited by limited management vision.
- Reuse potential is misjudged because of poor market awareness.
- Future gains possible through reuse are missed because of excessive emphasis on near-term productivity.
- There is low reuse effectiveness because reuse is treated as a side issue rather than as a part of systems or software engineering.
- There are weak responses to requests for contributions to the asset library because there are no incentives.
- Reuse is inefficient because roles for reuse activities are poorly defined.
- Reuse is inefficient because communication between application engineers, library support people, and domain engineers is inadequate.
- Costs or schedules can be affected when reuse across contractors on multicontractor projects increases interdependency.

C.1.4 Individual Personnel Capability

- Reuse activity is low because personnel do not have experience with selecting and using reusable assets.
- Reuse activity is low because personnel are reluctant to change from the current practice.
- There is poor reuse effectiveness because training in reuse was limited.

C.2 PROCESS RISKS

C.2.1 Life-Cycle Processes

- Reuse activity is low because the chosen development cycle provides no indication of where and how reuse is to be evaluated or implemented.

- Management cannot track reuse activity because reuse data collection is inadequate.
- There are disappointing reuse levels because reuse was not considered early in the project development cycle.
- There are disappointing reuse levels because the development processes, methods, or tools are not compatible with your reuse strategy.

C.2.2 Phase-Independent Activities

C.2.2.1 Resource Management

- Products are inadequate because overly ambitious schedules lead to inferior design and implementation choices.
- There is poor design for reuse because requirements for product evolution were not given.
- Reuse costs are high because of inadequate asset documentation.
- Inadequate reuse arises from difficulty in finding, understanding, or applying reusable assets.
- Reuse activity is low because retrieval tools were poorly chosen or training was inadequate.
- Reuse costs are high because missing components result in a loss of reuse opportunities or expensive searches to find them, which may result from poor repository management.
- Excessive effort is devoted to fixing components because no quality checks are made on new entries or the library entries have no associated quality metric so that users will know what to expect.
- Excessive effort is devoted to adapting components because the range of uses was so narrow or so broad that performance suffers.
- Components from previous projects do not integrate well for new applications because there was no common architecture.
- Applications costs are high despite a good reuse library because of failure to capture the knowledge needed to apply the components in building an application (domain knowledge). Hence, considerable support is needed from domain experts.
- The reuse library is underfunded because the cost of cataloging components was not included.
- Inappropriate reuse can result from a lack of knowledge or training concerning data rights.
- Failure to innovate and update assets arises from overemphasis on reuse to cut cost.
- Reusing assets is difficult because a common notation for describing assets is lacking.
- Reusing assets is difficult because architecture or interface standards for reusable assets are not defined or observed.

- Few contributions are made to the asset library because there is no defined process for testing and accepting new assets.
- Difficulty in supporting reuse may arise from failure to involve the library support staff in the reviews during the development of assets.
- Customer rejection of vendor test data can arise from poor reuse planning or vendor selection.
- Delivery problems may arise from failure to allow for custom upgrades from component vendors (especially if customer requirements change during development).

C.2.2.2 Configuration Management

- Component versions proliferate because there is inadequate control of assets.
- Low reuse efficiency results from failure to analyze the domain before creating a component library (i.e., components often cannot be reused or architecture needs extensive modification for new applications).
- Lower than expected productivity improvement results because the organization focused on low-level component reuse and ignored possibilities for reuse of large components and subsystems.
- Difficulties may arise when reusing assets from distributed libraries because the configuration management procedure does not address the needs of distributed libraries.
- Unpredictable problems arise during reuse because configuration management on assets is poor.
- Problems arise during system integration because configuration control was not applied to the design.
- Problems arise during system integration because configuration control was not applied to the support software.
- Problems arise in reusing assets because baselines at departure points in the evolution of an asset were not established.

C.2.2.3 Quality Management

- Poor design choices for building reusable assets can result from inadequate risk management.
- Poor quality assets for future use may arise from inadequate inspections, walkthroughs, and verification.
- High development cost and resistance to reuse on the part of developers can arise when poor component quality leads to frequent failures.
- Developers cannot tell if a component fits their requirements because documentation is poor and does not tell them what they need to know.

- Components perform poorly and often have to be replaced by custom code.
- Components have excessive memory requirements.
- A component is difficult to understand because no documentation is provided.
- Code components are poorly utilized because supporting information is omitted (e.g., specifications, operator's manuals, version descriptions, test procedures, and integration test plans).
- A component is difficult to modify because its code is incomprehensible.
- Applying reusable assets is difficult because interface design documentation for them is weak.
- Poor performance of completed systems can arise because measurable performance goals in asset requirements documents were lacking.
- Assets do not fit the application because established standards were not observed in building the assets.
- Problems arise during system integration because support software was not adequately documented.
- Problems arise during system test because support software was not tested adequately.
- Costly fixes during the application phase are needed because testing for reusable assets was not adequate.
- Poor asset quality may result from failure to involve a third party in testing reusable assets.
- Poor asset quality may result from failure to hold design reviews for reusable assets.

C.2.3 Phase-Dependent Activities

C.2.3.1 Project Definition

- Assets become obsolete before development cost is amortized because the changing or immature nature of the technology was not recognized.
- Components seldom fit new requirements because component requirements could not be anticipated.
- A contract is lost because of failure to recognize cost, schedule, or risk reductions available through reuse.
- Reuse potential is reduced by a poorly written contract (i.e., the contract does not permit full utilization of assets).
- Poor reuse efficiency may arise from an inadequate build versus reuse tradeoff procedure.

C.2.3.2 Concept of Operations

- Customers are unhappy with products because the design does not provide for user training or system maintenance.
- Customers are unhappy with products because the design does not provide fault tolerance.
- Users are unhappy with the system because human factors were not considered in the design of reusable interface components.
- The reused design poorly matches the concept of operations for current application.

C.2.3.3 Requirements

- Reuse potential is low because of excessive customer requirements.
- Higher cost and risk can result from failure to reuse requirements.

C.2.3.4 Design

- Components are difficult and costly to apply because they are overly complex.
- Low performance and high complexity may result from using components that do not fit the current design.
- Reusable components may not meet performance, operability, or supportability requirements unless selection is done carefully.

C.2.3.5 Implementation

- The system performs poorly because components were inappropriate, poorly designed, or too general.
- The system is too large because components were inappropriate, poorly designed, or too general.
- Proliferation of component versions and high costs may result from poor change control.

C.2.3.6 Integration and Test

- High development costs may result from poor quality components containing many errors.
- There are excessive testing costs because the possible reuse of test procedures and test data were ignored.
- High testing costs arise because test cases supplied with third-party components were inadequate.

C.2.3.7 Acceptance

- There is low customer satisfaction because a reused design does not provide the user interface expected.

C.2.3.8 Evolution

- Assets become obsolete because inadequate funds were allocated for updating them.
- Reuse potential is reduced because of a failure to plan for evolution of the system.
- Reuse potential is reduced because of inflexible system design.

C.2.4 Methods

- Reuse is low because methods are incompatible with application of reusable assets.
- Components are difficult to understand because of inadequate viewpoint representation (behavior, information, function).
- Poor reuse results because poor software engineering methods were used.
- Poor productivity improvement results because the benefits of reusing designs or test methods were ignored.
- Poor productivity with reuse results because the methodology used does not support it.

C.2.5 Automation

- Development costs remain high because incompatible development tools are used.
- Tools do not support reuse.
- The development environment does not support reuse.
- Expensive workarounds occur because of inflexible or inappropriate reuse.
- Difficulty meeting the competition is experienced despite high levels of reuse because the opportunity to automate program construction or code generation was ignored.
- Excessive documentation costs accrue because the opportunity to automate document generation was ignored.

C.3 PRODUCT RISKS

C.3.1 Algorithm

- Inefficient algorithms are used in components.
- There is inadequate fault tolerance of components.

C.3.2 Architecture

- Low levels of reuse or longer development times may result from a lack of standard architecture.

- Excessive component revisions may result from poor system modularization leads.
- Failure to use high-level abstractions can result from excessive focus on low-level parts.

C.3.3 Physical Realization

- Processor needs are underestimated because of insufficient modeling or poor component documentation.
- Memory requirements are underestimated because of poor component documentation or insufficient planning.

This page intentionally left blank.

APPENDIX D. SUMMARY OF LEGAL AND CONTRACTUAL REUSE ISSUES

D.1 INTRODUCTION

Since 1986, a policy has existed within the government to encourage reuse among and between government contractors. The policy favors commercializing software developed under government contracts (DeVecchio 1990).

Research in acquisition since 1986 has shown that additional development in the legal and contractual areas is needed to increase the effectiveness of reuse activities (STARS 1991; DeVecchio 1990; NSIA 1990; Samuelson 1986; Samuelson, Deasy, and Martin 1986; Samuelson and Deasy 1987; Defense Science Board [DSB] 1987). The general conclusion is that software reuse is still not addressed effectively in RFPs, contractor proposals, or contracts. Reuse is now "disincentivized" by:

- Complexity of the legal and ownership issues
- Limitations of present Federal Acquisition Regulations (FAR) and Defense Federal Acquisition Regulations Supplement (DFARS) and of the 1990 proposed changes, with regard to computer software
- Many stakeholders whose needs can enable or prevent successful reuse
- Current lack of adequate incentives to address those needs

Appendix D explores these issues in two sections.

Section D.2 identifies key **legal and contractual barriers to reuse.** Written from a business rather than a legal frame of reference, it describes the effects of the FAR, present and proposed, on software ownership, liability, and incentives to reuse software assets. Although the discussion takes place in the framework of existing contracts, RFPs, and Joint Integrated Avionics Working Group (JIAWG) recommendations, it does not constitute legal advice. Section D.2.2 also includes a brief discussion of recoupment, or recovery of government development costs.

Section D.3 identifies **successful approaches or strategies** that are now in practice, along with approaches that have been **proposed by previous** industry or government studies.

D.2 LEGAL AND CONTRACTUAL BARRIERS TO SOFTWARE REUSE

This section describes the effects of the present and proposed FAR on software ownership, liability, and incentives to reuse software assets. Although the discussion takes place in the framework of existing contracts, RFPs, and JIAWG recommendations, it does not constitute legal advice. Its focus is on the business impact of these regulations, as opposed to the legal aspects.

D.2.1 Complex Legal Issues in Reusable Software Assets

To illustrate the importance of these legal issues, a few common questions about legal rights to reuse software are presented. Answers to these questions are excerpted from DeVecchio (1990). For the cases in Table D-1, the original software is assumed to have been developed under a government contract.

Table D-1. Examples of What Software a Contractor Can Reuse

Question	Response
What software can be reused in the commercial market?	A contractor who develops software under a government contract is entitled to copyright it for reuse outside the government market. A subcontractor developing software for a prime contractor under a government contract can assert the copyright in software (or negotiate with the prime to give him that right).
Can it be reused for another contract with a different government agency?	A contractor can reuse software developed under an Air Force contract or a Navy contract. But keep in mind that the government **cannot pay twice for the same software**. For the Navy contract, the government can only pay for the costs of putting together the software, putting together the documentation, and delivering it. The Navy is then entitled to unlimited rights. A contractor cannot enter into a license agreement with respect to the software delivered to the Navy and cannot charge a royalty (the original developing organization of the reused software has those rights).
Are the government's "unlimited rights" the same as "exclusive" rights?	Case law says that if a contractor develops software under a government contract, the government's "unlimited rights" do **not** mean **"exclusive** rights." The contractor still has the right to sell the software and to charge a royalty. However, if the contractor sells it to somebody who is working on a government contract and charges a royalty, the cost cannot be passed on to the government.

Technical people need to know a minimum about the legal issues that affect their choice of reuse strategies. This section scratches the surface of the complexity in the many buyer-seller relationships found in the government software arena. These include relationships among owners of reusable software assets, developers, third-party developers, and customers. The section then briefly describes disincentives to reuse found in the present FAR and DFARS regulations and the proposed computer software changes that may help make software reuse more practical.

Figure D-1 illustrates the number and variety of sources for software assets that are reusable to a greater or lesser degree. The single largest collection of existing software is found in the DoD inventory (lightly shaded in the figure), some of which can be traced back three decades. A second category is public domain software (unshaded). The third category of reusable software assets (more heavily shaded) represents organizations with the mission to build software for sale. Organizations in this category are now seriously affected by the government's regulations and procedures for acquiring reusable software. These organizations' concerns include questions of ownership, liability, and incentives to both create reusable software and to reuse existing software. After defining terms used in this appendix, the section goes on to the organizations' concerns. The discussion focuses on disincentives to reuse contained in federal regulations governing software contracts.

Table D-2 shows operational definitions prepared by the JIAWG Software Task Group. Definitions are adapted from JIAWG (1990, item 1.2). Note that these operational definitions may not correspond precisely with the legal definitions given later in Table D-3 that was published in the *Federal Register* in October 1990.

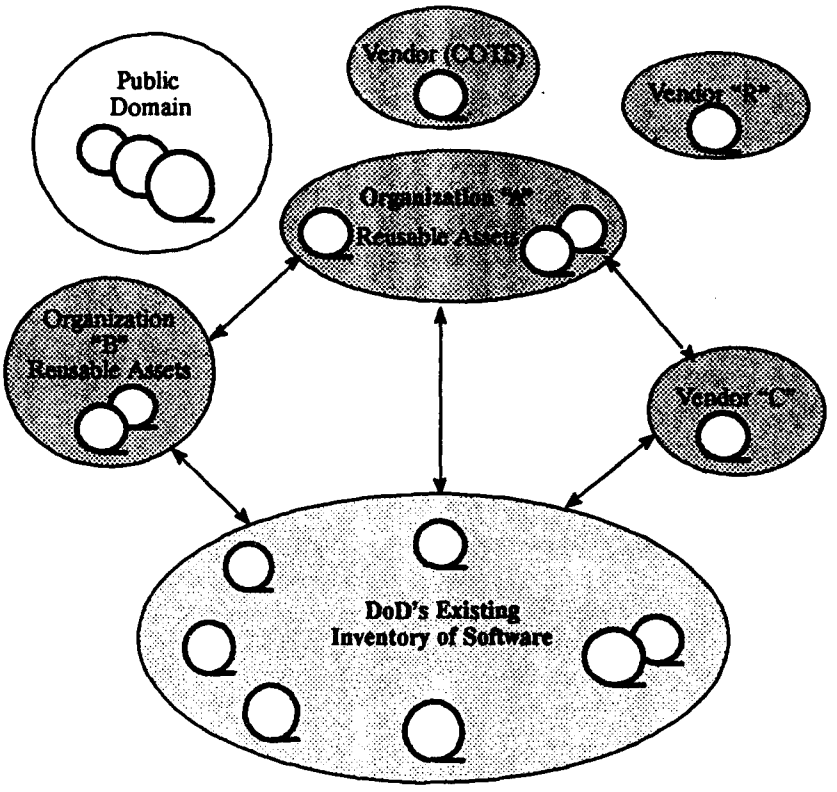


Figure D-1. Complex Relationships in Software Reuse

Table D-2. Operational Definitions of Terms for Software Reuse

Reusable Software Term	JIAWG Definition
Reusable Software Object (RSO)	Life-cycle objects that are created during the software development process, that are needed to operate, maintain, and upgrade the deliverable avionics during its lifetime, and that have the potential for reuse. The objects may include (but are not limited to): requirement specifications, design documents (both top level and detailed), source and object code, test specifications, test code, test support data, user manuals, programmer notes, and algorithms. These objects may be textual, graphical, or both; they are usually stored on electronic media; and, in DOD-STD-2167A and in . . . DFARS Parts 270 and 272, they are defined as computer software and computer software documentation.
Developer	A contractor who creates and makes use of RSOs, which may be incorporated into systems developed for the government. May be a prime or a subcontractor.
Unlimited Rights Software	Software objects that are developed under a government contract and which are delivered with unlimited rights to the government.
Restricted Rights Software	Software objects created at private expense that are not in the public domain and which are to be delivered (to the government) with less than unlimited rights.

Table D-2, continued

Reusable Software Term	JIAWG Definition
Licensed Software	RSOs developed for delivery under government contract, that the developer retains the right to license for third party use on other government contracts. This is applicable only if the alternate proposed license approach proposed herein is used.
Software Reuse Library	An element within a software engineering environment used to store and maintain those RSOs that are potentially reusable within and across aircraft programs.
Reuse Library Catalog	A centralized source of information about the RSOs that populate the Software Reuse Library.
Reuse Library Retrieval	Selecting items from a Reuse Library for systems application.

D.2.1.1 Ownership

The organization's first concern, ownership of reusable software assets, is deceptively simple. Who owns reusable software? This brief summary highlights the overall answer to this question:

The developer owns software assets outright when they are:

- Developed with the developer's **own funds**
- **Not at the same time** as a government contract
- **Not "necessary for performance of** a government contract"

If the developer wishes to reuse his privately developed software assets on another government contract to **retain** his ownership, the developer must specify to the government **in advance** which software assets the developer is delivering with only "restricted rights."

Under existing regulations, the federal government gets "unlimited rights" in software created by a developer if:

- The reused software is developed **with government funds.**
- The reused software is developed **at the same time** as a government contract.
- The software is **"necessary for performance of** a government contract."

Partially offsetting this, the developer under government contract is authorized to **copyright** and license others to reuse the software assets in the commercial market, unless the software is considered a "special work" (DFARS 227.476/252.227-7020).

The federal government gets unlimited rights in software in which the government already had unlimited rights from prior government-funded contracts.

Considering these provisions, it is easy to see why many in the software industry are concerned about selling software to the government. They fear that these provisions allow the government to claim "unlimited rights" in their proprietary software (STARS 1991).

When a developer wishes to reuse software assets owned by another (a "third party"):

- The developer must **negotiate with the owner** for a license to reuse.
- The developer may reuse government-owned software **"for government purposes"** without obtaining any license.
- The developer who wishes to reuse "public domain" software must **look at the documentation or code** for any licensing requirements or restrictions on rights to reuse.

D.2.1.1.1 Effects of Federal Regulations

This section provides important knowledge for software developers. A specialist points out that the FAR:

> . . . give the government the right, (even if you had developed software at private expense!) to take your software and give it to the government's support service contractors (perhaps your worst competitor).
>
> Of course the support service contractor is limited in what he can **do** with that software. But if one of your competitors gets your source code . . . it's in the brains of the competitors' people. From a legal viewpoint, your software no longer has any protection as trade secrets, because the knowledge was given them as a result of the government contract. (DeVecchio 1990)

In the face of such regulations, ownership issues cannot be left entirely to the lawyers. To protect their work, developers need a more detailed (though still very general) understanding of "ownership." Such an understanding begins with the basic provisions of the FAR, DFARS, and interim ruling on regulation changes proposed in October 1990.

The regulations affecting rights to software are covered in the FAR Section 52.227-14 (Rights in Data) for the nondefense agencies and for defense agencies in DFARS Part 227, Subpart 227.4 (STARS 1991; DeVecchio 1990). The present version of the DFARS was published as an Interim Rule in 1988. When published, this interim ruling caused sufficient comment from industry and government that a new draft "advanced notice of proposed rulemaking" was published on October 15, 1990. The intent of this proposal is to replace the current DFARS 227.4 (Interim Rule, 1988) and FAR 27.4 with a single regulation addressing rights in technical data and computer software for all government agencies.

How did this situation develop? The subject of rights to software has been in a state of flux for years. In Samuelson (1986), the SEI traced the origin of the difficulties to DoD's earlier decisions to acquire software products under the then-existing policy for acquisition of technical data. Considering the newness of the software field, this would seem to have been a reasonable approach at the time. In practice, its shortcomings became clear:

> . . . After a period of frustration, it became apparent that it was inappropriate to acquire software as if it were technical data. (The cost of acquiring **government-wide rights—which is what the technical data rights policy provides**—to software that was needed at only one government installation was impeding the acquisition of such software). So software (at least in machine-readable form) eventually became differentiated from technical data in the regulations, although software and technical data policy continue to be somewhat intertwined. Thus, while rights that attach to proprietary software are different from those that attach to technical data, the **same standard data rights clause** is nonetheless used to acquire rights in both (Samuelson 1986, 3) (emphasis added).

SEI went on to say:

> . . . DoD's rules and practices regarding software must make sense not only in terms of the technology but also in terms of the government's needs to use the technology and in terms of the economics of the software industry. The policy also needs to be clear and comprehensible to persons of average intelligence. The current software acquisition practices of the DoD fall short of these goals (Samuelson 1986, 3).

Today, the issues are complicated because both the FAR and DFARS treat software as though it were the same as technical data. The FAR and DFARS have basic clauses that address both subjects: FAR 52.227-14, Rights in Data, and DFARS 252.227-07013, Rights in Technical Data and Computer Software. This is true despite the fact that technical data and computer software are very different (STARS 1991, 3).

This is not a trivial bureaucratic distinction. While the proposed changes apply to treatment of technical data, **from the legal point of view, the changes do not automatically apply to computer software.** When doing business with the government, the specific legends provided in the regulations must be used to achieve any measure of protection for the company's software. As an important example of this seemingly trivial matter, if a developer puts a "restricted rights" legend on computer software documentation (instead of the "limited rights" legend that goes on technical data), he in effect gives the government unlimited rights to the documentation. A developer must make this distinction regularly and systematically. He must put one set of markings on deliverables of "restricted rights" software and databases and a different set of "limited rights" legends on documentation. The proposed changes in regulations governing "technical data" do not automatically apply to "computer software" (DeVecchio 1990).

In 1987, the federal government published regulations governing **non-DoD executive agencies** (such as the Departments of Commerce and the Treasury). The FAR were and currently are different from the DoD regulations. Thus, a contractor encounters problems if it does business with DoD **and** with any other executive agency. The same definitions and rules do not apply to the same software. The DoD regulations differentiate between computer software and documentation, but the regulations for non-DoD agencies do not:

- At the civilian executive agencies, computer software is defined as programs (excluding source code), databases, **and documentation.** All three categories were data subject to restricted rights and different legends than the DoD.

- But at DoD, **software** is different from **documentation.** Different rights and obligations apply to each category. Computer software is defined as machine readable programs (**including** source code) and databases, but **not** documentation.

Table D-3 is simplified to clarify the main issues and is not to be used as legal advice. The table compares the different treatment of several facets of software ownership. These vary among the FAR, DFARS, and proposed regulatory changes of October 1990. In general, the proposed regulations are expected to resolve many disparities that exist between FAR and DFARS and between the regulations on technical data and those on computer software.

Some attorneys, DeVecchio among them, see the possibility that the government will be entitled to **rights** in developers' **source code** if these proposed regulations go into effect and if some other things

happen. That possibility causes a lot of concern in the industry. Developers want to protect their source code.

In Table D-3, notice how definitions for computer software and data (in the first two rows) are different for the FAR and the DFARS. In the third column, the proposed change will continue the combined treatment of data and software. Thus, the issues will continue to be confused.

Table D-3. Comparison of FAR, DFARS, and Proposed Regulatory Change

Terminology	Federal Acquisition Regulations	Defense Federal Acquisition Regulations Supplement	Proposed Regulatory Change (October 1990)*
Software	Computer programs, computer databases, and documentation thereof (FAR 27.401).	Computer software and computer databases (DFARS 227.471). Treats documentation as technical data, not software.	Combined treatment of data and software continues to confuse the issues (STARS 1991).
Data	Recorded information, regardless of form or the media on which it may be recorded. Term includes technical data and computer software (FAR 27.401).	Recorded information, regardless of form or the media on which it may be recorded (DFARS 252.227.7013).	—
Copyright	Contractor authorized to establish copyright claim. Government granted paid-up, nonexclusive, irrevocable worldwide license to reproduce, prepare derivative works, perform, or display publicly by or on behalf of the government. License does not include right to distribute software to public (FAR 27.404(f)(iv)).	Contractor authorized to copyright unless software is considered a "special work," when software becomes sole property of government and is treated as unlimited rights software (DFARS 227.476/252.227-7020). Government granted a nonexclusive, paid-up license to reproduce, to distribute to the public, to perform or display publicly, and to prepare derivative works and have others do so for government purposes (DFARS 227.480/252.227-7013(3)).	More favorable than before. Does not include government right to distribute copies, as now found in DFARS, but does allow full disclosure for government purposes. No resolution of derivative works issues.
Government Purpose License Rights (GPLR)	No definition.	Right to use, duplicate, or disclose data and software, but only in the SBIR program, in whole or in part, in any manner for government purposes. This includes competitive procurement but not commercial purposes. Government can authorize others to use for government purposes.	Newly applies to software; contractor retains exclusive commercial rights for a negotiated period of time. Allows negotiation in cases of mixed funding by developer or government.

Table D-3, continued

Terminology	Federal Acquisition Regulations	Defense Federal Acquisition Regulations Supplement	Proposed Regulatory Change (October 1990)*
Required for Performance of a Government Contract or Subcontract	FAR 52.227-14(b)(i) may be interpreted in a similar manner.	Development was called for in the contract or subcontract, or it was accomplished during and was necessary for performance of a government contract or subcontract (DFARS 252.227-7013(c)(2)(ii)). This language requires unlimited rights for the government.	Replaced by "developed and necessary" for performance. This issue continues to be the most significant potential impediment to reuse (STARS 1991, 23).
Commercial Software	Reference to "existing computer software" as privately developed software normally vended commercially under a license or lease agreement restricting its use, disclosure, or reproduction (FAR 27.405(b)(2)).	(1) Title/ownership remains with contractor, (2) use is limited to facility where computer is located, (3) it cannot be made available to third party without contractor's permission (DFARS 252.227-7013(c)(1)(i)).	The restricted rights definition is the same as the one in the old FAR, i.e., five rights as opposed to four at DoD.
Unlimited Rights	Use, disclose, reproduce, prepare derivative works, distribute to public, perform and display publicly, in any manner, for any purpose and to have and permit others to do so (FAR 27.401). Note that "software" is not mentioned.	Use, duplicate, release, disclose in whole or in part in any manner for any purpose. Can give same rights to other parties (DFARS 227.471).	—
Restricted Rights Software**	(1) Use with computer for which it was acquired; (2) use on backup computer; (3) copy for archive; (4) modify, adapt, or combine with other software, provided that the modified . . . portions of any derivative software . . . are made subject to the same restricted rights; (5) any other rights not inconsistent with stated minimum rights (FAR 227.471).	(1) Use with computer for which it was acquired, (2) use on backup computer, (3) copy for archive, (4) modify, adapt, or combine with other software, provided that the modified . . . portions of any derivative software . . . are made subject to the same restricted rights. Any other rights not inconsistent with stated minimum rights (DFARS 252.227.7013).	—

Table D-3, continued

Terminology	Federal Acquisition Regulations	Defense Federal Acquisition Regulations Supplement	Proposed Regulatory Change (October 1990)*
Form, Fit, and Function Data	For computer software, it means data identifying source, functional characteristics, and performance requirements but **specifically excludes the source code, algorithm, process, formulae, and flow charts of the software** (FAR 27.401).	Technical data that describes the required overall physical, functional, and performance characteristics (along with the qualification requirements, if applicable) of an item, component, or process to the extent necessary to permit identification of physically and functionally interchangeable items (DFARS 252.227.7013).	—
Unpublished Software	No definition.	Not yet released to public or furnished to others without restriction on further use or disclosure. Note that this definition includes developed software not yet or perhaps never intended to be commercialized.	—

* Proposed regulatory change to the FAR: Rights in Technical Data - Advanced Notice of Proposed Rulemaking, Federal Register, Vol. 55, No. 199, October 15, 1990.

** "Restricted rights software" is "Computer software developed at private expense and that is a trade secret; is commercial or financial and confidential or privileged; or is published copyrighted computer software; including minor modifications of such software (FAR 27.401)."

Sources: STARS (1991); DeVecchio (1990); FAR (1988), DFARS (1990)

Other highlights of Table D-3 show that:

- Copyright treatment is more favorable than before. The proposed regulatory change does not include a government right to distribute copies, as now found in DFARS. It does allow full disclosure for government purposes.
- Under copyright, there is no resolution of "derivative works" issues in the proposed changes.
- GPLR now applies to **technical data** (not software), but it may be possible to negotiate GPLR for computer software. The contractor retains exclusive commercial rights for a negotiated period of time. GPLR allows negotiation in cases of mixed funding by the developer or government; FAR recognized only unlimited rights or restricted rights.
- The concept "required for performance of a government contract or subcontract" was replaced by "developed and necessary for performance." DeVecchio (1990) interprets this to mean, "If you develop software at private expense, it is yours. But, if it is **necessary for the performance of the contract, it is ours.**" If the development was funded under a contract, then the government would get unlimited rights.

  However, even if a developer funded it at private expense, if the computer software is developed **during the performance of a contract** and he must develop that software to perform

the contract, then (notwithstanding that the developer used company funds and that the software was not required expressly in the contract) the government will get rights in the software (DeVecchio 1990). "This issue continues to be the most significant potential impediment to reuse" (STARS 1991, 23).

- The government has "unlimited rights" to use the documentation for government purposes. Note in Table D-3 that "software" is not mentioned.

- Regarding "restricted rights" software, if the developer used his own money, including any "indirect accounts," he would be able to restrict the government's rights. For this purpose, IR&D, bid and proposal costs, and costs allocated to overhead accounts in accordance with government standards for generally accepted accounting principles are **not** considered costs of development. Private expense is anything **other than direct** government funding (DeVecchio 1990).

- "Form, fit, and function data" gives contractors to DoD agencies the **right to withhold** software, a right non-DoD agencies have had since 1987. If the contracting officer agrees that the government does not really need all of the software in the performance of the contract, the developer could **withhold the software and only give form, fit, and function data.** Note that the contractor would also need to negotiate out the "deferred ordering" clause, which gives the government the right to demand the data within 3 years.

- Regarding "commercial" software, the **restricted rights** definition is the same as the one in the old FAR, i.e., **five** rights as opposed to four at DoD. Under the proposed regulations, the government will have the right to give a developer's computer software to support services contractors, even at DoD, a right they did not previously have. An apparent anomaly is that DoD would have the right to give a developer's commercial computer software to **third parties.** The government does **not** have this right if the software was developed at private expense and is **not** commercial.

- The "prime-sub negotiation of restricted rights" has never been expressly stated in the regulations. If a prime contractor, under a contract containing the basic data rights clause, acquires existing commercial computer software from a subcontractor at any tier for delivery to or for use on behalf of the government, the contracting officer may either authorize the use of the "commercial computer software" clause **or approve any variations to or limitations on the restricted rights** (the **five** minimum rights) set forth in the basic data rights clause in a collateral agreement incorporated into the contract.

  This clause says that a subcontractor to a government prime contractor may negotiate with the prime **greater restrictions** than those in the main regulations, i.e., That is, a subcontractor can negotiate with the government to give the government less than the minimum rights.

- Regarding "unpublished" software, if a developer does not put the copyright symbol (•) on the software and tells the government that it is **unpublished** software (meaning it is a trade secret), then the government does **not** have the right to disclose it to others, except in accordance with the government's restricted rights.

  This can be important. Under FAR, if the developer put a copyright symbol on the software, the government would treat it as published computer software—**which gives it the right to disclose it to others.**

In the DoD, a copyright symbol on the software automatically gave the government a copyright license coextensive with whatever rights the developer had.

Paradoxically, a developing organization will have greater rights if it does not put the copyright symbol on some software that has not been disseminated or that has not been published outside the company. In dealing with an executive agency other than DoD or with these new regulations, the developer will have greater rights if he does not put the copyright symbol on some software.

Other aspects of this situation are discussed in Section D.2.1.1.2.

Understanding this minimum level of detail is necessary for developers because, under both the DoD and FAR regulations, the federal regulation takes priority if any terms of a developer's standard licensing agreement conflict with the requirements of the federal regulation. This means that developers have very little protection if they enter into contracts with prime contractors or with the federal government. Neither their standard license agreement nor their claim to have copyrighted commercial computer software protects them (DeVecchio 1990).

D.2.1.1.2 Copyright Law

Reused software qualifies as "derivative works" under another legal specialty, copyright law. Copyright law, which comes into effect for "derivative works," is intertwined with the FAR and DFARS and has perhaps even greater impact on ownership of reused software. From the government viewpoint, DoD personnel are concerned about derivative works resulting from reusing software. SEI describes many problems shared by developers and government that can ". . . arise when 'derivative works' are created from an original piece of software" (Samuelson 1986, 9). SEI explains:

> The term software reuse has several meanings. A common factor to each of these meanings . . . is that it may be an instance of the **creation of a derivative work** which may involve the complex regulations of the **copyright law.**
>
> . . . For the government to make, or have made for it, software which is in some way derived from a program in which another party holds a copyright, without having first obtained the permission of the copyright holder, raises the possibility that the government will be found to have infringed the copyright. As a result, the government may be prohibited from making use of the newly developed software.
>
> The potential impact of the derivative works right for software is broader even than its effect on software reuse projects. Virtually any effort which in some way **alters software and causes it to act in a way different from its original function** may be found to be the creation of a derivative work should the copyright holder challenge the government's actions in court. Thus, even basic maintenance and enhancement efforts . . . to the extent that the changes may be said to improve the software, might be found to be derivative works—the creation of which infringes the rights of the copyright holder. Such projects also raise questions as to ownership rights in the newly created product. (Samuelson 1986, 9)

In general, industry prefers to protect its intellectual property rights by using "trade secret" protection rather than patents or copyrights. This strategy helps industry avoid disclosing to the public any details of products or intellectual property.

However, SEI emphasizes that DoD procurement regulations **do not recognize the existence of trade secret protection** for software or technical data. Instead, DoD regulations ". . . specifically incorporate copyright law in some respects, and also seem to contemplate that copyright law may govern as to some things" (Samuelson 1986, 17).

D.2.1.1.3 Implications for Developers

The National Security Industrial Association (NSIA) published a developers' view, *The Business Issues Associated With Software Reuse,* in December 1990. It reports on a study conducted by the Software and C^{3}IC Committee's Joint Subcommittee on Software Reuse. The findings are particularly relevant to this project because the NSIA subcommittee limited its study to considerations related to reusable Ada software in software-intensive DoD weapon system applications. The subcommittee concentrated its work on near-term issues and solutions as being more profitable in stimulating the concept of software reuse (NSIA 1990, 7).

NSIA findings on ownership. The report urges that the government continue to support initiatives in the area of software data rights:

> No data rights issues emerged that were **uniquely** associated with reusable software (emphasis added). . . . Data rights issues associated with software reuse are very important and must be resolved. However, it turns out that **the set of problems associated with rights to reused software are precisely those that have been and are currently being addressed by government and industry.** (NSIA 1990, 21)

> The government should continue to vigorously support on-going initiatives in the area of software data rights. . . . Related to data rights, nothing additional needs to be done by the government in order to promote the concept of reuse, other than to support the software data rights initiatives already in progress. (NSIA 1990, 21)

An earlier report (DSB 1987) made a similar recommendation in the area of software data rights for all military software:

> DoD should follow the concepts of the proposed **FAR 27.4 for data rights for military software,** rather than those of the proposed DoD Supplement 27.4, or it should adopt a new "Rights in Software" Clause as recommended by Samuelson, Deasy, and Martin (1986) in Appendix D6. (DSB 1987, 32)

Actions needed to protect developers' privately funded software. If these regulations are passed, a company will be able to protect its rights in its privately developed software to the maximum extent by doing three things:

1. Have in place a mechanism (system, procedure, discipline) for ensuring that it **can prove** that it develops its software at private expense. Before entering into a contract, a contractor must specify in an agreement which computer software developed with the contractor's funds it intends to supply to a DoD agency or a prime contractor with restricted rights, not unlimited rights. This agreement must be incorporated into the contract. The company then represents that it is providing a **correct** written assertion of restricted rights data and that the company is entitled to limit the government's rights.

   Proving scope and stage of development of privately funded software. A contractor must be able to prove that it develops its software at private expense before any government contract. Thus, the person who signs the certificate must **know or have a reasonable basis to believe** that he can restrict the government's rights in this matter. He must convince the government that the software was (a) developed before any government contract and (b) was developed without any government funding.

   Companies need internal procedures to prove their assertions. They must be able to document the stages of development of their privately funded software. To face a challenge by the

government, the company must be able to prove that the **software was workable** (to a degree that people skilled in the art would say "that software will work") before it signed that contract.

A company also must have a mechanism for a clear understanding of its engineers' **scope** in its privately funded development work. It also needs to watch what the engineers are doing; when they are creative, engineers may go beyond the bounds of a statement of work. If their work should get into an area that is contractually required in performance of the contract, the company could lose its rights.

A contractor must check new RFPs for requirements that might overlap with its privately funded software. Before the company starts its proposal effort, it must be sure that the contract it is proposing **does not overlap** with an IR&D project it already has.

If the RFP requires some related development work and the company has not finished its independent development work, then it negotiates with the government so that the contract omits that development. Otherwise, the privately funded development may **occur during** and **be necessary for the performance of** the government contract. Then, no matter who funded the software, the government will get unlimited rights.

Before they open projects for computer software, contractors must check their existing contracts for requirements that may overlap with the privately funded software. Unless they make sure that their privately funded development work is **not similar to** work already in progress under a government contract, the government will get unlimited rights in that development.

2. When dealing with an executive agency other than DoD under the new regulations, state in the agreement the **characteristics of the privately developed software that will be recognizable** if the government should choose to modify it.

3. Do the best it can to **avoid delivering** privately developed software to the government. If the government requires delivery, the company can try to negotiate constraints on what the government can do with it. A subcontractor, dealing with a prime contractor, will have somewhat more leverage when it attempts to negotiate this.

A company needs these disciplines to demonstrate that its software was developed at private expense. Until it can show that, it cannot restrict the government's rights in any privately developed software. Even though this may seem excessive, with all this care, the regulations make it difficult for a company to prove its case (DeVecchio 1990).

D.2.1.2 Liability

The most credible study regarding the effects on industry of software reuse was NSIA (1990), also discussed in Section D.2.1.1.3.

With regard to liability, the report stated:

> The risks of liability due to reuse of software are important and deserve serious management attention. However, the liabilities associated with software reuse do not appear to offer a significant impediment to the concept. (NSIA 1990)

The NSIA noted that its initial perception of the importance of legal liability changed after the subcommittee had completed its study:

> . . . Once software is modified by someone other than its owner, the **owner's exposure becomes almost nil.** In developing DoD weapon systems, software being reused will almost certainly require some modification. Consequently, the system developer will bear the primary liability for the reused software just as he does for vendor supplied hardware. . . . Integration of software wherein some portion is being reused is not very different from traditional hardware integration and the liability considerations are the same.
>
> . . . During the first several meetings of the subcommittee, the many liabilities of ownership, especially legal, were perceived as deterrents to the voluntary engagement in software reuse. However, after these meetings . . . the subcommittee concluded that this was not a significant issue. (NSIA 1990, 20)

D.2.1.2.1 Implications for Developers of Liability Concerns

The NSIA report noted that its perception of the importance of the legal liability issue had changed after the subcommittee had completed its study:

> The legal liabilities associated with software reuse do not appear to offer a significant impediment to the concept. . . . The risks of liability due to reuse of software are important and deserve serious management attention. However, the liabilities associated with software reuse do not appear to offer a significant impediment to the concept. (NSIA 1990, 20)

D.2.1.2.2 Government-Owned Government- or Contractor-Operated Reuse Libraries

CARDS (1993b) includes discussions of liability issues regarding domain-specific reuse libraries owned by the government and operated by the government or a contractor. It states that liability depends upon the library's degree of involvement with reusable software and identifies factors influencing liability:

- Nature and frequency of the library's activities (e.g., is it merely storing assets or is it certifying assets, are all activities performed each time an asset is submitted?)
- How the activity is performed (e.g., is the activity based on an objective set of criteria or standards?)
- Function or scope of knowledge of the library's staff, suppliers, and subscribers (e.g., is the library staff's role strictly administrative?)
- Technical qualifications for the library staff, supplier, and subscriber (e.g., is the person a system or software engineer with domain experience?)
- Purpose of the library, supplier, and subscriber (e.g., is the library intended to extensively evaluate reusable software components as quality assurance to the subscriber?)

CARDS (1993b) states that "many issues of contention can be avoided through negotiation and careful drafting of supplier and subscriber agreements."

D.2.1.3 Incentives to Create Reusable Software and to Reuse Existing Software

It is widely believed that the current environment for software acquisition is not compatible with reuse of software (except Non-Developmental Items and COTS). DoD can offer few incentives for

contractors to create reusable software assets and to reuse existing assets. In practice, the present FAR environment is seen as a **disincentive** to software developers to either produce reusable code or to incorporate reusability into their design efforts (DeVecchio 1990; NSIA 1990).

This is not a new perception. In 1987, the DSB Task Force on Military Software, with considerable vision, observed this situation:

> . . . today's policies actively discourage the reuse of software modules from one system to another. We recommend a variety of policy changes, each designed to encourage reuse, and indeed, the establishment of a public market in reusable software parts. (DSB 1987, 3 recommendations 29-33)
>
> . . . The SEI should establish a prototype module market, focused originally on Ada modules and tools for Ada, with the objective of spinning it off when commercially viable. (DSB 1987, 3)

The DSB then recommended comprehensive action to overcome the problems:

> The Undersecretary of Defense (Acquisition) and the Assistant Secretary of Defense (Comptroller) should direct Program Managers to identify in their programs those subsystems, components, and perhaps even modules, that may be expected to be acquired rather than built; and to reward such acquisition in the RFPs. (DSB 1987, 3 recommendation 31)

The DSB clearly intended to stress the need for incentives in several areas:

> . . . The Undersecretary of Defense (Acquisition) should develop economic incentives, to be incorporated into standard contracts, to **allow contractors to profit from offering modules for reuse,** even though built with DoD funds. (DSB 1987, 36 recommendation 29)
>
> The Undersecretary of Defense (Acquisition) should develop economic incentives, to be incorporated into all cost-plus standard contracts, to **encourage contractors to buy modules and use them** rather than building new ones.
>
> Acquisition contracts should be structured so that contractors will be motivated to build and sell reusable software, and to buy it. Reusable software will be successful **when contractors decide it is in their competitive self-interest** to reuse software rather than to develop it each time. The proper incentives with **respect to data rights, warranties, licenses, liabilities, and maintenance** must be included in the RFPs and the contracts. (DSB 1987, 36 recommendation 30)
>
> DoD should devise increased productivity incentives for custom-built software contracts, and make such incentivized contracts the standard practice. A new contracting form, part-way between fixed-price and cost-plus-fee, should be devised. (DSB 1987, 30 recommendation 17)
>
> DoD should develop metrics and measuring techniques for software quality and completeness, and incorporate these routinely in contracts. (NSIA 1990, 31 recommendation 19)

The NSIA subcommittee's view, very similar to that of the DSB, was consistent with the marketing dynamics in the industry. NSIA recommends that the government use the **competitive process** as a powerful tool for fostering software reuse, continue to vigorously support ongoing initiatives in the area of software **data rights**, use only **cost-type contracts** wherein software reuse is to be an important factor, and invest its available software reuse resources in software engineering **standards**, particularly in standards specific to the concept of reuse (NSIA 1990, 3).

NSIA sees no net saving in the short term by developing and using reusable software and no incentive for contractors to do so—except **by using their internal libraries** of reusable software assets.

Consistent with this view, NSIA believed that the government should use only cost-type contracts when software reuse is seen as an important factor (NSIA 1990, 15):

> Industry is also not convinced of the potential for cost savings by using what is currently available in the libraries. Companies believe that the software currently in reuse libraries suffers from poor or nonexistent documentation, and from poor software engineering practices. Furthermore, by using software from a library, **a company does not gain a unique advantage over competitors** because the library is available to everyone. In addition, the risks and costs of using library software cannot be determined quickly due to the lack of necessary standards and proper documentation. (NSIA 1990, 12)

> . . . many companies, primarily the large prime contractors, are in the process of establishing internal proprietary software reuse libraries. Further, those companies are drawing from those libraries, whenever practical, to use in the development of proposals and subsequently, the software development project. There is evidence that, in some of these cases, software reuse made the difference in winning the contract. (NSIA 1990, 11)

The subcommittee saw an opportunity to take significant action on its own in a way that would benefit both reuse across government programs and help promote commonality among companies' internal software libraries:

> . . . if the level of abstraction of the software assets was raised from the code level to the requirement and specification level, including common library techniques and standards for reusable requirements and specifications, the government would promote greater software asset reuse within companies and develop a basis for cross-company use in the future. One benefit of this effort would be to help develop a bridge between different companies' internal libraries and to promote commonality in more abstract levels of software asset libraries. (NSIA 1990, 17)

NSIA stressed several actions for the government to initiate. The government should:

> . . . investigate a means for effectively promoting increased utilization of **internal corporate** (emphasis added) software **reuse libraries.** (NSIA 1990, 3)

> . . . make an effort to develop techniques and tools for applying reuse at the requirements and specifications level and encourage contractors to utilize and/or incorporate the results into their internal proprietary libraries. . . . The government can make a significant contribution to the evolution of software reuse by developing system requirements and specifications that are themselves reusable. Reusable requirements foster reusable designs and algorithms, which in turn foster reusable new code. . . . Software reuse at the lower levels of abstraction will be promoted by reuse at the higher levels of software abstraction.

> Standards for reusable requirements and specifications, followed by the development of reusable requirements and specifications that conform to such standards, would foster greater software asset reuse. This is common practice in the specification of hardware requirements for system developments. Altimeters, radars, radios, generators, etc. are commonly used across platform and system developments. (NSIA 1990, 17)

Table D-4 lists the possibilities developed and described in Institute for Defense Analysis (1991).

Table D-4. Mechanisms Applicable to Reuse

Encourage Reuse	Discourage Reuse
Competition for government contracts	Reluctance to give up organizations' expertise and ways of doing business
RFP or source selection criteria that include software reuse	RFP or source selection criteria that penalize bids using existing software assets

Table D-4, continued

Encourage Reuse	Discourage Reuse
Tools to find and retrieve assets	Lack of critical mass of reusable assets
Software engineering techniques to help build reusable assets and encourage reuse	Current state of practice that is oriented to creation from scratch and away from supporting existing assets
Government commitment to take risks to encourage reuse	Pressures on program managers to avoid risk and meet schedule and budget
Contracting mechanisms that encourage reuse across programs	Access to and protection for proprietary software
Emphasis on prototyping to help establish demands for reusable software assets	
Standards to provide basic structure and confidence that assets will have long life	
Education on benefits and costs of reuse	
Source: Adapted from Institute for Defense Analysis (1991)	

The STARS report describes a new possibility, use of value engineering, found in FAR 48.101 (b)(2). Their recommendation is that this avenue be explored and a test program be selected to investigate applicability of the concept to software reuse. It also discusses profitability, sales volume and market share, technology lead, prestige, and recognition (STARS 1991, 15).

D.2.1.3.1 Implications of Incentives for Developers

NSIA findings on incentives to reuse. In its report, NSIA listed three initial questions that had driven its study: "(1) What can be done to encourage companies to create software that is reusable, both by themselves and, eventually, by other firms? (2) What encouragement would result in the utilization of software previously developed for reuse? and (3) What are the legal and fiscal barriers to software reuse?" (NSIA 1990, 2). Its recommendations were traditional approaches and were well grounded in the field of software engineering. NSIA recommends that the government:

> . . . utilize the **competitive process** as a powerful tool for fostering software reuse.
>
> . . . continue to vigorously support ongoing initiatives in the area of software **data rights.**
>
> . . . use only **cost type contracts** wherein software reuse is to be an important factor.
>
> . . . invest their available software reuse resources in software engineering **standards,** particularly in standards specific to the concept of reuse. (NSIA 1990, 3)

The report concludes that, "**Competition** is the dominant factor in motivating industry." It adds that "financial incentives, such as royalties, for the purpose of fostering software reuse do not appear to be implementable" (NSIA 1990, 3):

> The competitive process offers the greatest potential for fostering software reuse. Incentives that help companies win new contracts are considered by the industry to be more critical than incentives which may incrementally increase profits earned on software development efforts. (NSIA 1990, 14)
>
> . . . The area of greatest significance was found to be the competitive contracting process. Winning the contract takes priority over all other business considerations. The industry representatives generally agreed that the inclusion of some software reuse considerations in the government's process of selecting the winning contractor (source selection) would have much greater potential for fostering reuse than

> any of the other business incentives that had been suggested. . . . Including appropriate reuse incentives in the source selection process is the most effective business incentive for software reuse that was uncovered by the subcommittee.
>
> . . . The subcommittee actively searched for business incentives to foster software reuse. . . . (industry) representatives generally felt that an increased profit potential might provide some useful incentive, but, since software development is only a portion of the total effort in many contracts, this incentive may not be significant. (NSIA 1990, 14)

D.2.1.3.2 Software Maintenance and Enhancements

While it is essential for the government to have rights in documentation for software it has contracted to have built, SEI stated that:

> . . . Obtaining rights in the government to modify software is not a current software licensing problem of DoD. The DoD procurement regulations require that in all software acquisition contracts for proprietary software the government must at minimum get the right to modify the software. This regulatory authority is important since copyright law might otherwise prohibit the modification of software without the permission of the copyright owner to make a "derivative work." DoD regulations appear to be sufficient to secure for DoD the right to modify software it acquires. (Samuelson 1986, 7)

D.2.1.3.3 Need for Better Training About Software, Data Rights, and Intellectual Property Law

In the case of training for procurement personnel, SEI reported a serious need:

> . . . Although DoD is fortunate to have many dedicated, competent individuals among its procurement personnel, these individuals reported that they feel inadequately trained for the role they have to perform in complex software acquisition contracts. Much of the software that the contracting personnel must acquire is "state of the art" technology. Communication between procurement personnel and users seems to be infrequent, which makes maintenance and supportability planning more difficult. Often procurement personnel have no training in software technology, software life cycles, or software support systems. Further, the procurement regulatory structure within which the negotiation process must proceed—especially as to data rights—is quite complex. Finally, the turnover rate among procurement personnel is high, which only aggravates the situation. (Samuelson 1986, 8)

D.2.2 Recoupment of Government Development Costs

An Interim DFARS Rule, 52.235-7002, Recovery of Nonrecurring Costs on Commercial Sales, requires that contractors pay back government funds that were used to provide contract support for developing an item. Payback is required when a contractor "intends" to enter commercial sales of the item or essentially similar items. Because this may have a serious impact on software reuse, JIAWG identifies this as a major barrier to reuse. JIAWG comments that:

> The application of recoupment to software seems more accident than conscious policy. To our knowledge, no one responsible for recoupment policy has studied the implications of applying recoupment to software, especially as the ranges of possible reuse expand and DoD depends more on COTS (Commercial Off the Shelf Software) and multi-use software components to develop military software. (JIAWG 1991)

Its proposed solutions include promulgating a policy mandating negotiation of recoupment in software acquisitions. Without such negotiations, the contract should not contain a recoupment

clause. JIAWG also recommends beginning a dialog with NSIA to support the NSIA project to establish an equitable recoupment policy and to broaden the investigation to include recoupment of industry development investments (JIAWG 1991, 17-18).

D.3 SUCCESSFUL APPROACHES OR STRATEGIES FOR REUSE

This section identifies successful approaches or strategies that are now in practice, along with approaches that have been proposed by previous industry and government studies. Table D-5, which is adapted from STARS (1991), shows several such applications, categorized by the nature of the reused software assets.

Table D-6 summarizes important differences between the government's "unlimited rights" and "restricted rights" in software. Several basic strategies are identified in Table D-7 along with the government agency's rights that are appropriate for protecting an organization's proprietary investments.

Table D-5. Examples of Recent Reuse

Reuse Assets	Nature of Reuse		Description of Reuse
	Internal	External	
Code (Similar Domain)	FCDSSA	—	Navy facility produces ship-unique software for Fleet Combat Direction System installations.
	Foxboro	Foxboro	Foxboro claims 80% internal reuse of code without modification, also licenses code packages for reuse.
	EVB	EVB	EVB licenses code packages for reuse.
	ATCCS	—	Army requires use by ATCCS contractors.
	AFATDS	AFATDS	Magnavox requires in-house assessment before initiating new development. Government award fee criteria assess the degree of reuse. Modified software creates new code, which is considered "derived" software and may raise copyright issues.
	Nobel Tech	—	Swedish company obtained competitive advantage in telephone system domain.
Code (Dissimilar Domain)	—	UNAS	TRW developed UNAS, deriving it from government-owned software and sold commercially.
	EVB	EVB	EVB licenses code packages for reuse.
	AFATDS	AFATDS	Modified software creates new code, which is considered "derived" software and may raise copyright issues.
Specification "A," "B," and "C-5" specifications	—	ATCCS	Associate contractors required to use specifications generated by TRW. Award fee used to provide incentives to contractors for reuse. Reusing specifications appears to raise level of interface and avoid changing software.
Domain Knowledge Base	ATCCS	—	SEI support for ATCCS.
	SDIO NTB	—	National Test Bed.
	Navy Research Lab C^3I	—	STARS Task UR40.
Top Level Architecture	Granite Sentry	—	Air Force acted as system integrator.
Templates	ASD(WPAFB) Flight Simulator Program	—	No CSCIs. Templates used to regenerate software. LORAL is prime contractor. SEI helped develop templates.
Source: Adapted from STARS (1991)			

Table D-6. Government Customer Now Gets These "Rights" in Software

For Government Purposes, With "Unlimited Rights" to Use Software	"Restricted Rights" to Use Software
Use on any government computer.	Only for the computer for which the software was acquired.
Use with any government computer.	On a backup computer.
Duplicate for any purpose.	Copy only for archive purposes.
Disclose to other parties for government purposes.	Under DoD contracts, modify or combine with other software (ensuring that the derivative software based on restricted rights software contains same restrictions) **even though** it is copyrighted or subject to license agreements.
Give same rights to other parties (e.g., competitors with related government contracts).	Any other rights not inconsistent with the stated minimum rights (FAR 227.471).

Table D-7. Summary of Legal Rights Appropriate for Candidate Development Strategies

Organizational Strategy for Reuse	Developer's Strategy	Government Agency's Rights
Reuse at the project level.	Reuse owned assets (developed with company funds) on any government projects.	Restricted rights only.
Reuse across projects.	Reuse owned assets (developed with company funds) on projects in related fields; i.e., generate many varieties of end products similar to those that we now produce.	Restricted rights only.
Develop similar products in the same domain.	Reuse owned assets (developed with company funds) on contracts in same domains.	Restricted rights only.
License to third-party developers.	Negotiate license with owner (e.g., Booch parts).	Restricted rights only.
Use public domain software (e.g., Booch).	Examine documentation and source code for notification of rights given by owner; may be restricted to not-for-profit use.	Examine documentation and source code for notification of rights given by owner; may be restricted to some uses.
Reuse business.	Develop and provide libraries of **government-owned** reusable assets to government customers; use such assets in developing new software systems.	Unlimited rights.

This page intentionally left blank.

APPENDIX E. REUSE ASSESSMENT REPORT ANNOTATED OUTLINE

EXECUTIVE SUMMARY

The executive summary provides a synopsis of the entire report's content to a management-level audience. A reader should be able to get a general idea of the motivation and approach for the assessment effort. Everything discussed in the executive summary should be discussed in greater detail in the document. You should write the executive summary last. If you do a good job on the document, you can get a good first draft of the executive summary from the rest of the report. Try to keep this summary to a maximum of five pages. If necessary, you can use figures and tables.

Cover at least the following topics:

- Identify the organization and products covered by the assessment.
- List the objectives of the assessment.
- Summarize the important findings of the assessment.
- Summarize the conclusions and recommendations for further action.

1. INTRODUCTION

The introduction provides background information on the assessment and the approach to performing the assessment.

1.1 SCOPE

Identify the organization and products covered by the assessment. Explain where and when the assessment was conducted.

1.2 ASSESSMENT OBJECTIVES

Describe the objectives of the assessment.

1.3 PARTICIPANTS

Identify those who participated in the assessment. Include their roles where appropriate.

1.4 CONDUCT OF THE ASSESSMENT

Describe how the assessment was conducted.

1.5 ORGANIZATION OF THE REPORT

Describe the organization for the remainder of the report.

2. DOMAIN ASSESSMENT RESULTS

This section is intended to present the results of the domain assessment. You should summarize the objectives of the assessment team and how the assessment was conducted. It is important that you define here the domains assessed.

The following subsections are used to report the factors identified through consensus of the team participants on the potential benefits of reuse in each domain. Describe here the format to be used in presenting each factor. A suggested format is:

Factor: Identification of a domain characteristic or property of the organization's assets and processes that will support successful reuse in the domain considered.

2.1 NAME OF FIRST DOMAIN

The section may be organized to correspond to the factors in the Domain Assessment Model or key findings of the team.

2.1.1 Name of First Topic

List each factor found.

Additional explanation should be provided to help the reader understand each conclusion.

2.1.2 Name of Second Topic

Repeat for each topic.

2.2 NAME OF SECOND DOMAIN

Repeat for each domain.

2.3 SUMMARY OF DOMAIN ASSESSMENT RESULTS

Report here the major conclusions of the domain assessment. You should include here overall conclusions of the assessment team concerning the reuse opportunities and constraints identified. In particular, you should indicate any additional steps needed before a decision is made to invest in reusable assets.

3. REUSE CAPABILITY ASSESSMENT RESULTS

This section is intended to present the results of the reuse capability assessment. You should summarize the objectives of the assessment team and how the assessment was conducted. You may

also include here some overall conclusions of the assessment team concerning the strengths and opportunities for improvement for the organization.

The following subsections are used to report the findings identified through consensus of the team participants on the effectiveness of the organization, its software engineering process, and its software engineering environment with respect to reuse. Describe here the format to be used in presenting each factor. A suggested format is to describe each finding using one or more of the following:

Strength: Identification of an activity performed or capability possessed by the organization supporting an effective reuse practice

Opportunity: Identification of a good prospect for improvement, building on the organization's strengths, and leading to a more effective reuse practice

Benefit: Identification of the expected advantages gained by carrying out an identified opportunity

Additional explanation should be provided that describes the relevance of the finding to effective reuse and the context in which the participants identified the finding.

3.1 NAME OF FIRST TOPIC

List each finding found under this topic using the format described above.

Additional explanation should be provided to help the reader understand these findings.

3.2 NAME OF SECOND TOPIC

Repeat for each topic.

4. NEXT STEPS

The purpose of this section is to outline the next steps in defining the reuse adoption program.

APPENDIX A. ORGANIZATIONAL PROFILE

This appendix is used to document the organizational profile of the organization. The organizational profile is a rough sketch of the current organization and is used to establish the scope and context for the reuse capability and domain assessments. You may wish to use figures and tables to describe the profile. The profile covers the following areas:

- Organization
- Products
- Business model
- Process
- Environment

The following subsections document the profiles for the areas listed above.

A.1 ORGANIZATION PROFILE

Describe the structure of the organization and how it supports reuse.

A.2 PRODUCT PROFILE

Describe the products produced currently and to what extent reuse is applied.

A.3 BUSINESS MODEL

Describe how business is conducted. You should cover the following:

- Whether development is funded internally or through contracts
- Whether reusable assets are acquired or developed internally
- How application of reusable assets is funded
- How maintenance of reusable assets is funded

A.4 PROCESS PROFILE

Describe the processes and methods applied to software development and how reuse of assets is accomplished with them.

A.5 ENVIRONMENT PROFILE

Describe the environment and tools used for software development and how they support reuse of software.

APPENDIX B. REUSE PRACTICE HISTORY

This section is optional and is intended to summarize the history of reuse practice within the organization.

APPENDIX F. REUSE ACTION PLAN ANNOTATED OUTLINE

EXECUTIVE SUMMARY

The executive summary provides a synopsis of the entire plan's content to a management-level audience. A reader should be able to get a general idea of the motivation, approach, schedule, resources, and risks associated with the reuse adoption effort. Everything discussed in the executive summary should be discussed in greater detail in the document. You should write the executive summary last. If you do a good job on the document, you can get a good first draft of the executive summary from the rest of the action plan. Try to keep this summary to a maximum of five pages. If necessary, you can use figures and tables.

Briefly summarize the following topics in the executive summary:

- The events that led to the initiation of the reuse program and to the development of the action plan
- The objectives and goals of the reuse program
- Highlights of the selected reuse adoption strategy
- Major milestones of the schedule
- Estimated resources required to implement the plan
- Areas of high risk and actions taken or planned to avert these risks

1. INTRODUCTION

The introduction provides background information on the reuse program, defines the scope and direction of the reuse program, describes the expected results of implementing the plan, describes the approach taken to implement reuse, and identifies areas of risk.

1.1 REUSE PROGRAM

Summarize the events that led to the initiation of your reuse program. Identify the sponsors and their motivations for the reuse program. Describe any specific reuse opportunities you intend to exploit by instituting a reuse practice.

Describe the reuse adoption objectives of the reuse program, indicating what is to be accomplished by the program, the time frame, and the extent to which the organization's operations are to be affected.

Also, summarize the events that led to the development of the action plan, such as the domain and reuse capability assessments, strategy development, and strategy selection. Include references to any applicable reports.

1.2 REUSE ADOPTION GOALS AND CONSTRAINTS

Describe the reuse adoption goals, i.e., the specific results toward which the reuse program is directed. Distinguish between those goals you expect to accomplish in the current adoption cycle (near-term goals) and those you expect to accomplish in later cycles (long-term goals). Identify any constraints that limit goal accomplishment.

1.3 REUSE ADOPTION STRATEGY

Describe your reuse adoption strategy for achieving the objectives and goals. Cover the key points of each strategy component: product, business model, process, organization, environment, and transition. Discuss your rationale for choosing the selected strategy, referencing any documented analyses. Briefly describe any new technologies or procedures that will be introduced as part of your strategy, and reference any detailed description in the appendixes or other documents.

1.4 RISK MANAGEMENT

List the areas of high risk and their possible consequences. Identify any actions you have taken or plan to take to avert these risks.

2. ORGANIZATION AND RESOURCES

Organization and resources describe the organizational structure, roles, responsibilities, and resources needed to implement the selected reuse adoption strategy and perform the actions outlined in the action plan.

2.1 ORGANIZATION STRUCTURE

Describe the organizational structure you will use to implement the action plan. Include an organization chart if available. This includes the organizational structure for managing the reuse action plan implementation, as well as the resultant organizational structure for implementing and sustaining your approach to reuse. Identify organizational changes, such as the addition of new working groups or consolidation of existing functions. Identify key management and engineering positions, and describe their responsibilities.

2.2 PERSONNEL

Identify the total personnel resources and skills required to manage and implement the action plan. Identify the individuals or sources of personnel resources.

2.3 FACILITIES

Identify any changes to your facilities or purchases of new equipment or software.

2.4 FUNDING

Describe the source and amount of funding that will be used to manage and implement the action plan.

3. ACTIONS

Actions describe in detail the work to be performed to implement your reuse adoption strategy, the deliverables, and the schedule for performing the actions. You can include a work breakdown structure here to show the action decomposition.

3.1 ACTION DESCRIPTIONS

In this section, you will describe each action in detail. You can use the work breakdown structure as a mechanism for organizing this section, describing the actions in the order they appear in the work breakdown structure. You could also use the reuse adoption goals as an organizing framework, describing the actions associated with each goal.

3.1.1 Name of First Action

Describe the action to the level of detail sufficient for the individual or group responsible for implementing the action to understand what is to be accomplished. Items to include are:

- Definition of the action
- Required inputs
- Deliverables
- Estimated effort required to perform the action
- Planned start and completion dates
- Identification of the organization or personnel responsible for implementing the action
- Funding source or charge code

3.1.2 Name of Second Action

Repeat this subsection for each action.

3.2 ACTION NETWORK

The action network describes the sequential relationships among the actions, e.g., on which actions does Action A rely on for its input. You can use a diagram to illustrate these relationships, such as a PERT chart.

3.3 SCHEDULE AND MILESTONES

In this section, you will provide the overall schedule of the actions. You can illustrate the schedule graphically, e.g., a Gantt chart. You should also identify key events or milestones, such as reuse program reviews, completion of a key deliverable, first significant reuse, etc.

4. DATA COLLECTION

Data collection describes the metrics you will use to monitor the progress of your implementation against your reuse adoption goals.

4.1 METRICS

Define the metrics to be collected. The metrics should be tied to your reuse adoption goals. Indicate the planned values for the metrics if applicable.

4.2 FREQUENCY OF DATA COLLECTION

Describe when the identified metrics data will be collected.

4.3 MECHANISMS FOR DATA COLLECTION

Identify how the metrics data will be collected, who will collect the data, and any special tools or techniques used to collect the data.

APPENDIX A. NEW TECHNOLOGIES AND PROCEDURES

You can use this appendix to describe in greater detail any new technologies or procedures to be adopted as a result of your reuse adoption strategy. Your description should include:

- A description of the new technology or procedure with a list of reference material and sources of expertise
- The reason for adoption
- Enabling technologies or procedures that must be in place for this technology to work
- Skills and training required
- The approach to tailoring the technology to suit your specific needs

You can repeat this appendix for each technology or create a subsection of this appendix for each technology.

APPENDIX G. ASSESSMENT WORKSHEETS

This appendix provides the worksheets used by the assessment team in conducting a domain assessment and reuse capability assessment. Section 4 describes the use of the domain assessment worksheets, and Section 5 describes the use of the reuse capability assessment worksheets. The worksheets are listed below with their corresponding section numbers.

Domain assessment worksheets include:

- Domain assessment attributes (Section G.1)
- Domain assessment: individual response (Section G.2)
- Domain assessment: response summary (Section G.3)
- Domain assessment profile (Section G.4)

Reuse capability assessment worksheets include:

- Reuse capability assessment: individual response (Section G.5)
- Reuse capability assessment: response summary (Section G.6)
- Reuse capability profile (Section G.7)

G.1 DOMAIN ASSESSMENT ATTRIBUTES

Place your estimates in the Individual Response worksheet provided in Section G.2. Use the scale 1 through 5 to indicate the extent to which your domain exhibits the specified attribute.

1 Not exhibited
2 Slightly exhibited
3 Moderately exhibited
4 Mostly exhibited
5 Fully exhibited

Please use the space provided after each attribute to enter any comments regarding your estimates.

Market Potential

Product Situation

PS-1. New planned products could benefit.

Comments: _____

PS-2. Products currently in development could benefit.

Comments: _____

PS-3. Products ready for enhancement could benefit.

Comments: _____

PS-4. The value of using reusable assets is high for the given product situation.

Comments: _____

Market Situation

MS-1. There is substantial demand in the given market.

Comments: _____

MS-2. The current customer base is representative of the market.

Comments: _____

MS-3. Increased market share will result from investment in reuse.

Comments: _____

MS-4. This market is expected to grow.

Comments: _____

Existing Domain Assets

Individual and Organizational Expertise

EE-1. There are individuals on staff who are experts in the domain (i.e., they have experience building and using domain assets).

Comments: _____

EE-2. Your organization has experience building products for this domain.

Comments: _____

Availability of Existing Software

AS-1. Existing assets support all work products needed to develop new members of a product family (i.e., requirements, design, code, test data, and documentation).

Comments: _____

AS-2. Existing assets are well integrated.

Comments: _____

AS-3. The features of all assets are clearly traceable to the requirements.

Comments: _____

Quality of Existing Software Assets

AQ-1. Packaging existing assets for reuse will be much less expensive than developing them from scratch.

Comments: _____

AQ-2. Assets can be adapted for expected market needs and packaged for reuse at less cost than developing them from scratch.

Comments: _____

Commonalities and Variabilities

Commonalities in Behavior, Structure, and Implementation

CR-1. A large portion of the products can be built with common components.

Comments: _____

CR-2. A common architecture for the domain is feasible to enhance reuse.

Comments: _____

Variability

VR-1. Variable requirements can be managed (i.e., either by isolation to separate components or by providing an enumerated set of options).

Comments: _____

VR-2. Common components can be isolated from variabilities in supporting hardware and software.

Comments: _____

VR-3. Nonessential variabilities in customer requirements can be managed.

Comments: _____

Domain Stability and Maturity

Stability of Customer Needs

CS-1. There is sufficient time to recover the investment cost before evolution of customer needs makes the assets obsolete.

Comments: _____

CS-2. The trends in customer needs are clear enough to permit the design of assets that will accommodate the changes.

Comments: _____

Technological Evolution

TS-1. There is sufficient time to recover the investment cost before evolution of the underlying technology makes the assets obsolete.

Comments: _____

TS-2. The technological trends are clear enough to permit the design of assets that will accommodate the changes.

Comments: _____

_____

_____

_____

Standardization in the Domain

Standards

FS-1 Existing or planned standards will stabilize requirements.

Comments: _____

_____

_____

_____

FS-2. COTS components matching existing standards are available.

Comments: _____

_____

_____

_____

FS-3. Industry standards are coming, and a substantial market will exist for reusable assets that conform to them.

Comments: _____

_____

_____

_____

FS-4. The organization has a key position in the domain and can lead in developing needed standards.

Comments: _____

_____

_____

_____

FS-5. Organizational standards are an appropriate basis for designing reusable assets.

Comments: _____

_____

_____

_____

G.2 DOMAIN ASSESSMENT: INDIVIDUAL RESPONSE

Attribute Identifier	Domain			
	A	B	C	D
	Exhibited: 1–not, 2–slightly, 3–moderately, 4–mostly, 5–fully			
Market Potential				
PS-1				
PS-2				
PS-3				
PS-4				
MS-1				
MS-2				
MS-3				
MS-4				
Existing Domain Assets				
EE-1				
EE-2				
AS-1				
AS-2				
AS-3				
AQ-1				
AQ-2				
Commonalities and Variabilities				
CR-1				
CR–2				
VR-1				
VR-2				
VR-3				
Domain Stability and Maturity				
CS-1				
CS–2				
TS-1				
TS-2				
Standardization in the Domain				
FS-1				
FS-2				
FS-3				
FS-4				
FS-5				

G.3 DOMAIN ASSESSMENT: RESPONSE SUMMARY

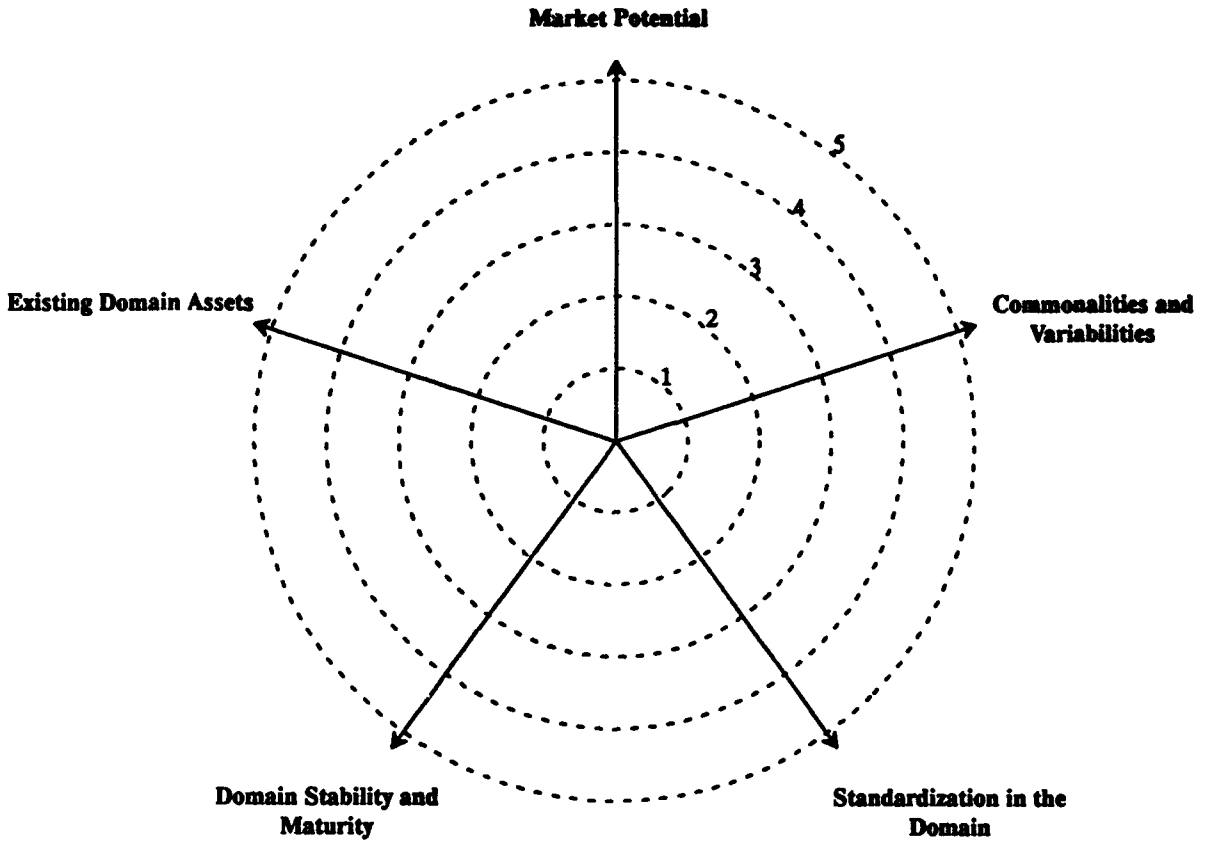
Attribute Identifier	Domain	Assessor Scores								Total	Mean
		E	E	E	E	E	E	E	E		
Market Potential Attributes											
PS-1	A										
	B										
	C										
	D										
PS-2	A										
	B										
	C										
	D										
PS-3	A										
	B										
	C										
	D										
PS-4	A										
	B										
	C										
	D										
MS-1	A										
	B										
	C										
	D										
MS-2	A										
	B										
	C										
	D										
MS-3	A										
	B										
	C										
	D										
MS-4	A										
	B										
	C										
	D										
E (Exhibited): 1–not, 2–slightly, 3–moderately, 4–mostly, 5–fully											

Attribute Identifier	Domain	Assessor Scores								Total	Mean
		E	E	E	E	E	E	E	E		
Existing Domain Assets											
EE-1	A										
	B										
	C										
	D										
EE-2	A										
	B										
	C										
	D										
AS-1	A										
	B										
	C										
	D										
AS-2	A										
	B										
	C										
	D										
AS-3	A										
	B										
	C										
	D										
AQ-1	A										
	B										
	C										
	D										
AQ-2	A										
	B										
	C										
	D										
E (Exhibited): 1-not, 2-slightly, 3-moderately, 4-mostly, 5-fully											

Attribute Identifier	Domain	Assessor Scores								Total	Mean
		E	E	E	E	E	E	E	E		
Commonalities and Variabilities											
CR-1	A										
	B										
	C										
	D										
CR-2	A										
	B										
	C										
	D										
VR-1	A										
	B										
	C										
	D										
VR-2	A										
	B										
	C										
	D										
VR-3	A										
	B										
	C										
	D										
Domain Stability and Maturity											
CS-1	A										
	B										
	C										
	D										
CS-2	A										
	B										
	C										
	D										
E (Exhibited): 1-not, 2-slightly, 3-moderately, 4-mostly, 5-fully											

Attribute Identifier	Domain	Assessor Scores								Total	Mean
		E	E	E	E	E	E	E	E		
Domain Stability and Maturity (continued)											
TS-1	A										
	B										
	C										
	D										
TS-2	A										
	B										
	C										
	D										
Standardization in the Domain											
FS-1	A										
	B										
	C										
	D										
FS-2	A										
	B										
	C										
	D										
FS-3	A										
	B										
	C										
	D										
FS-4	A										
	B										
	C										
	D										
FS-5	A										
	B										
	C										
	D										
E (Exhibited): 1–not, 2–slightly, 3–moderately, 4–mostly, 5–fully											

G.4 DOMAIN ASSESSMENT PROFILE



1 Factor not exhibited
2 Factor slightly exhibited
3 Factor moderately exhibited
4 Factor mostly exhibited
5 Factor fully exhibited

G.5 REUSE CAPABILITY ASSESSMENT: INDIVIDUAL RESPONSE

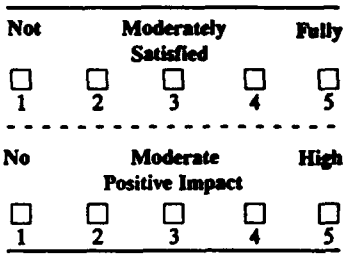
Place an x on the provided scales to indicate your judgment of the extent to which your organization meets the specified goal (1 through 5) and the expected impact on an organization's reuse capability from fully satisfying the stated goal (1 through 5). You should indicate a positive impact when achieving the goal would result in an expected positive reuse performance or prevent an expected negative reuse performance.

Application Development Factors

Asset Awareness and Accessibility

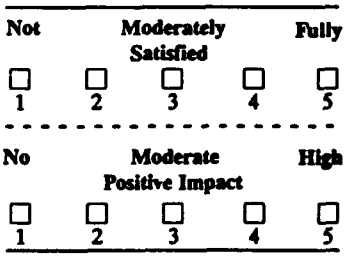
AA-1. Developers are aware of, can find, and have access to any relevant reusable assets and external sources of assets.

Comments: _____



AA-2. Developers are aware of and reuse assets that are specifically acquired or developed for their application.

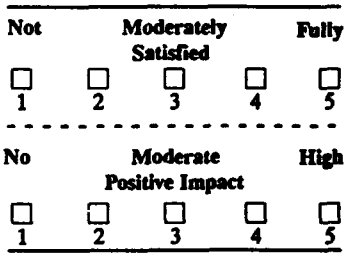
Comments: _____



Asset Identification

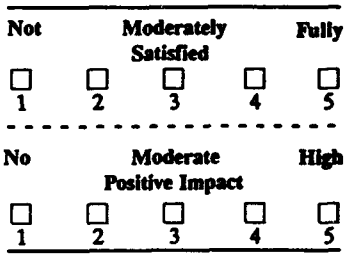
AI-1. Developers identify potential reusable assets in each major activity in the development life cycle that might satisfy their development needs.

Comments: _____



AI-2. Developers identify potential reusable assets before establishing constraints that limit reuse opportunities.

Comments: _____



AI-3. Developers identify and reuse early life-cycle assets that result in the reuse of corresponding later life-cycle assets.

Comments: _____

Not Moderately Fully Satisfied
☐1 ☐2 ☐3 ☐4 ☐5

No Moderate High Positive Impact
☐1 ☐2 ☐3 ☐4 ☐5

AI-4. Developers identify and reuse designated high-payoff, reusable assets.

Comments: _____

Not Moderately Fully Satisfied
☐1 ☐2 ☐3 ☐4 ☐5

No Moderate High Positive Impact
☐1 ☐2 ☐3 ☐4 ☐5

Asset Evaluation and Verification

AE-1. Developers understand and evaluate reusable assets against their needs before committing to reuse an asset.

Comments: _____

Not Moderately Fully Satisfied
☐1 ☐2 ☐3 ☐4 ☐5

No Moderate High Positive Impact
☐1 ☐2 ☐3 ☐4 ☐5

AE-2. Developers verify asset quality before selection.

Comments: _____

Not Moderately Fully Satisfied
☐1 ☐2 ☐3 ☐4 ☐5

No Moderate High Positive Impact
☐1 ☐2 ☐3 ☐4 ☐5

Application Integrability

AN-1. Technologies applied in developing applications (i.e., processes, methods, and tools) facilitate the integration of reusable assets.

Comments: _____

Not Moderately Fully Satisfied
☐1 ☐2 ☐3 ☐4 ☐5

No Moderate High Positive Impact
☐1 ☐2 ☐3 ☐4 ☐5

Place an x on the provided scales to indicate your judgment of the extent to which your organization meets the specified goal (1 through 5) and the expected impact on an organization's reuse capability from fully satisfying the stated goal (1 through 5). You should indicate a positive impact when achieving the goal would result in an expected positive reuse performance or prevent an expected negative reuse performance.

Asset Development Factors

Needs Identification

NI-1. Current developer needs for solutions are identified.

Comments: _____

Not Moderately Satisfied Fully

☐1 ☐2 ☐3 ☐4 ☐5

No Moderate Positive Impact High

☐1 ☐2 ☐3 ☐4 ☐5

NI-2. Anticipated developer needs for solutions are identified.

Comments: _____

Not Moderately Satisfied Fully

☐1 ☐2 ☐3 ☐4 ☐5

No Moderate Positive Impact High

☐1 ☐2 ☐3 ☐4 ☐5

NI-3. Current customer needs for solutions are identified.

Comments: _____

Not Moderately Satisfied Fully

☐1 ☐2 ☐3 ☐4 ☐5

No Moderate Positive Impact High

☐1 ☐2 ☐3 ☐4 ☐5

NI-4. Anticipated customer needs for solutions are identified.

Comments: _____

Not Moderately Satisfied Fully

☐1 ☐2 ☐3 ☐4 ☐5

No Moderate Positive Impact High

☐1 ☐2 ☐3 ☐4 ☐5

NI-5. Identified needs are used as a basis for acquiring or developing reusable assets to meet the specified needs.

Comments: _____

Not Moderately Satisfied Fully

☐1 ☐2 ☐3 ☐4 ☐5

No Moderate Positive Impact High

☐1 ☐2 ☐3 ☐4 ☐5

Asset Interface and Architecture Definition

AD-1. The relationships of an asset to its external environment are defined.

Comments: _____

Not — Moderately Satisfied — Fully
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

No — Moderate Positive Impact — High
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

AD-2. The relationships between assets within the same major activity (e.g., requirements analysis, design, etc.) of the development life cycle of a product are defined.

Comments: _____

Not — Moderately Satisfied — Fully
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

No — Moderate Positive Impact — High
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Needs and Solutions Relationships

NS-1. Transformation relationships between needs and their corresponding solutions are defined (e.g., from requirements to design, from design to code, etc.).

Comments: _____

Not — Moderately Satisfied — Fully
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

No — Moderate Positive Impact — High
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

NS-2. Identified transformation relationships between needs and solutions are used as a basis for acquiring or developing broad-spectrum assets.

Comments: _____

Not — Moderately Satisfied — Fully
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

No — Moderate Positive Impact — High
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Commonality and Variability Definition

CV-1. Commonality and variability relationships between needs are defined and used to acquire or develop assets to meet multiple needs.

Comments: _____

Not — Moderately Satisfied — Fully
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

No — Moderate Positive Impact — High
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

CV-2. Commonality and variability relationships between solutions (assets, architectures of assets) are defined and used to acquire or develop adaptable solutions for meeting multiple needs.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

Asset Value Determination

AV-1. The net value of an asset is estimated based on experience data and identified needs.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

AV-2. The estimated net value of an asset is used as a basis for focusing the organization's resources on acquiring or developing high-payoff assets for reuse.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

Asset Reusability

AR-1. Technologies applied in developing reusable assets facilitate their integration into applications.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

Asset Quality

AQ-1. Reusable assets are under configuration control.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

AQ-2. Feedback on reusable assets is collected and used to maintain and enhance the reusable assets.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

AQ-3. Sufficient data is provided for an application developer to understand, evaluate, and apply reusable assets.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

AQ-4. Reusable assets are verified against their specifications.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

AQ-5. Reusable-asset quality goals or standards are established and tracked.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

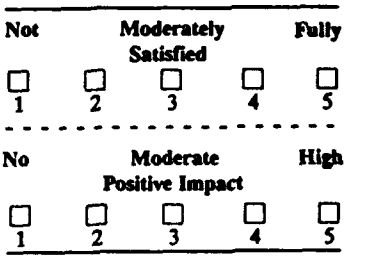
Place an x on the provided scales to indicate your judgment of the extent to which your organization meets the specified goal (1 through 5) and the expected impact on an organization's reuse capability from fully satisfying the stated goal (1 through 5). You should indicate a positive impact when achieving the goal would result in an expected positive reuse performance or prevent an expected negative reuse performance.

Management Factors

Organizational Commitment

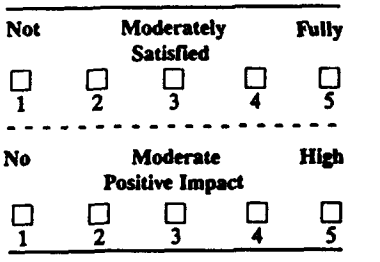
OC-1. Management commits to defining, implementing, and improving the organization's approach to reuse and demonstrates its commitment to the staff.

Comments: _____



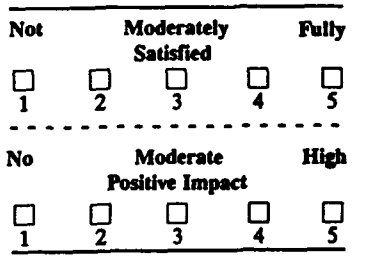
OC-2. Management commits funding, staffing, and other resources to define, implement, and improve the organization's approach to reuse.

Comments: _____



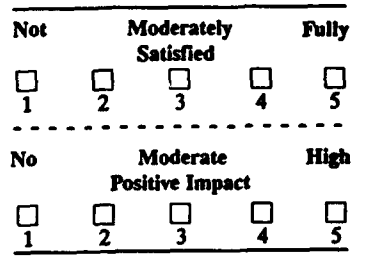
OC-3. The staff supports the organization's approach to defining, implementing, and improving the organization's approach to reuse.

Comments: _____



OC-4. Management structures its organization, policies, procedures, and standards to facilitate a standard reuse process supporting multiple product development efforts.

Comments: _____



Planning and Direction

PD-1. Reuse strategies are defined for product development efforts in support of product objectives, and product development activities are planned and directed in accordance with the product development's reuse strategy.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

PD-2. Product line reuse strategies are developed to maximize the benefits of reuse over sets of related products, and the product line activities are planned and directed in accordance with the strategies.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

PD-3. Management develops a long-term strategy for improving the organization's reuse capability, and the organization's improvement efforts are planned and directed in accordance with the improvement strategy.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

PD-4. New business opportunities are created that take advantage of the organization's reuse capability and reusable assets.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

Costing and Pricing

CP-1. Reuse cost accounting procedures are defined and enacted.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

CP-2. Management establishes a long-term plan for funding reuse activities and improvement efforts.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

CP-3. Product pricing and funding strategies take into account expected costs and anticipated benefits of reuse over the product or product line.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

Legal and Contractual Constraints

LC-1. Reuse legal or contractual constraints are identified and enforced.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

LC-2. Legal and contractual constraints on reuse are removed or reduced to increase the potential for reuse when feasible.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

Intergroup Coordination

IC-1. Application and asset development activities are coordinated between and among application and asset development groups to identify, track, and resolve intergroup issues.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

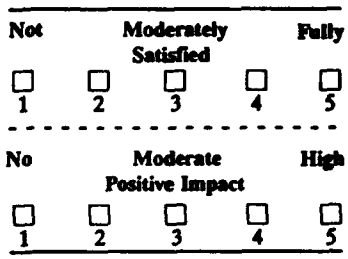
Place an x on the provided scales to indicate your judgment of the extent to which your organization meets the specified goal (1 through 5) and the expected impact on an organization's reuse capability from fully satisfying the stated goal (1 through 5). You should indicate a positive impact when achieving the goal would result in an expected positive reuse performance or prevent an expected negative reuse performance.

Process and Technology Factors

Process Definition and Integration

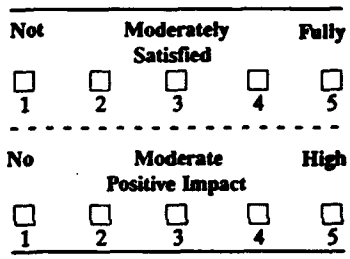
PI-1. Reuse activities and resources are identified in the product software development plan.

Comments: _____



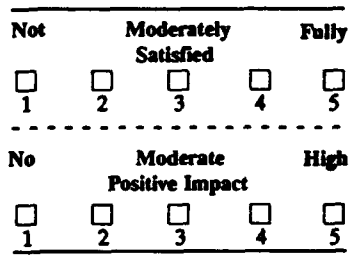
PI-2. Standard reuse processes are defined and integrated with the organization's standard software development process.

Comments: _____



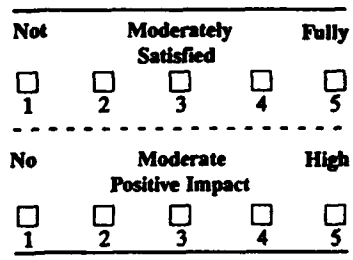
PI-3. Standard reuse processes provide sufficient flexibility for tailoring the standard processes to the unique needs of the product development efforts.

Comments: _____



PI-4. The standard reuse processes have sufficient flexibility to adapt to new product environments (markets).

Comments: _____



Measurement

MS-1. The impact of reuse on cost, schedule, and product is estimated during development planning; then actual impacts are recorded as the development proceeds.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

MS-2. Reuse experiences from past and current projects are collected and made available.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

MS-3. The effectiveness and efficiency of reuse technologies are measured and used as a basis for determining the most suitable technologies for given situations.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

Continuous Process Improvement

CI-1. Performance of the standard reuse processes is measured and analyzed to increase understanding and identify strengths and weaknesses.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

CI-2. Plans are established to systematically address weaknesses identified in the standard reuse processes.

Comments: _____

Not / Moderately Satisfied / Fully
☐1 ☐2 ☐3 ☐4 ☐5

No / Moderate Positive Impact / High
☐1 ☐2 ☐3 ☐4 ☐5

Training

TR-1. The knowledge and skills necessary to effectively apply an organization's reuse technologies are determined, gaps in the staff's knowledge and skills are identified, and a plan is developed and enacted to fill the gaps.

Comments: _____

Not Moderately Satisfied Fully
☐1 ☐2 ☐3 ☐4 ☐5

No Moderate Positive Impact High
☐1 ☐2 ☐3 ☐4 ☐5

TR-2. The effectiveness of training to fill the gaps in necessary reuse technology knowledge and skills is measured and analyzed to identify weaknesses.

Comments: _____

Not Moderately Satisfied Fully
☐1 ☐2 ☐3 ☐4 ☐5

No Moderate Positive Impact High
☐1 ☐2 ☐3 ☐4 ☐5

TR-3. Plans are established to systematically address the weaknesses identified in reuse technology training.

Comments: _____

Not Moderately Satisfied Fully
☐1 ☐2 ☐3 ☐4 ☐5

No Moderate Positive Impact High
☐1 ☐2 ☐3 ☐4 ☐5

Tool Support

TS-1. Reuse tools are used to support defined reuse activities and methods for which they are effective.

Comments: _____

Not Moderately Satisfied Fully
☐1 ☐2 ☐3 ☐4 ☐5

No Moderate Positive Impact High
☐1 ☐2 ☐3 ☐4 ☐5

TS-2. Reuse tools are acquired, developed, or tailored to support the standard reuse processes.

Comments: _____

Not Moderately Satisfied Fully
☐1 ☐2 ☐3 ☐4 ☐5

No Moderate Positive Impact High
☐1 ☐2 ☐3 ☐4 ☐5

TS-3. Reuse tools are integrated with the organization's software development environment.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

TS-4. Reuse tools provide sufficient flexibility to adapt to new process or product environments.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

Technology Innovation

TI-1. Management and staff are aware of new and evolving technologies and standards that may affect their products and reusable assets.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

TI-2. New technologies are identified that will meet or drive customer needs, have a clear benefit, and take advantage of the organization's reuse capability. Selected technologies are inserted into the organization's process or product.

Comments: _____

Not		Moderately Satisfied		Fully
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

No		Moderate Positive Impact		High
☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

G.6 REUSE CAPABILITY ASSESSMENT: RESPONSE SUMMARY

Goal Identifier	Assessor Scores								Total	Mean
	S/I	S/I	S/I	S/I	S/I	S/I	S/I	S/I		
Application Development Factor Goals										
AA-1										
AA-2										
AI-1										
AI-2										
AI-3										
AI-4										
AE-1										
AE-2										
AN-1										
Asset Development Factor Goals										
NI-1										
NI-2										
NI-3										
NI-4										
NI-5										
AD-1										
AD-2										
NS-1										
NS-2										
CV-1										
CV-2										
AV-1										
AV-2										
AR-1										
AQ-1										
AQ-2										
AQ-3										
AQ-4										
AQ-5										

S (Satisfied): 1–not, 2–slightly, 3–moderately, 4–mostly, 5–fully
I (Impact): 1–no, 2–low, 3–moderate, 4–moderately high, 5–high

Goal Identifier	Assessor Scores								Total	Mean
	S/I	S/I	S/I	S/I	S/I	S/I	S/I	S/I		
Management Factor Goals										
OC-1										
OC-2										
OC-3										
OC-4										
PD-1										
PD-2										
PD-3										
PD-4										
CP-1										
CP-2										
CP-3										
LC-1										
LC-2										
IC-1										
Process and Technology Factor Goals										
PI-1										
PI-2										
PI-3										
PI-4										
MS-1										
MS-2										
MS-3										
CI-1										
CI-2										
TR-1										
TR-2										
TR-3										
TS-1										
TS-2										
TS-3										
TS-4										
TI-1										
TI-2										

S (Satisfied): 1–not, 2–slightly, 3–moderately, 4–mostly, 5–fully
I (Impact): 1–no, 2–low, 3–moderate, 4–moderately high, 5–high

G.7 REUSE CAPABILITY PROFILE

Critical Success Factor		Opportunistic	Integrated	Leveraged	Anticipating
Application Development	AA	1☐	2☐		
	AI	1☐	2☐	3☐	4☐
	AE	1☐ 2☐			
	AN		1☐		
Asset Development	NI	1☐ 5☐	2☐	3☐	4☐
	AD	1☐	2☐		
	NS			1☐ 2☐	
	CV	1☐	2☐		
	AV				1☐ 2☐
	AR		1☐		
	AQ	1☐	2☐ 3☐ 4☐	5☐	
Management	OC	1☐ 2☐ 3☐	4☐		
	PD	1☐	3☐	2☐	4☐
	CP	1☐	2☐	3☐	
	LC	1☐			2☐
	IC		1☐		
Process and Technology	PI	1☐	2☐ 3☐		4☐
	MS	1☐	2☐		3☐
	CI			1☐ 2☐	
	TR	1☐		2☐ 3☐	
	TS	1☐	2☐	3☐	4☐
	TI			1☐	2☐

Key:
- ☐ Goal not satisfied
- ◪ Goal slightly satisfied
- ◪ Goal moderately satisfied
- ■ Goal mostly satisfied
- ■ Goal fully satisfied

⊟ Impact less than moderate

Numbers represent goal identifiers

LIST OF ABBREVIATIONS AND ACRONYMS

ARPA	Advanced Research Projects Agency
CARDS	Central Archive for Reusable Defense Software
CASE	computer-aided software engineering
CDROM	compact disk read-only memory
CFRP	Conceptual Framework for Reuse Processes
CMM	Capability Maturity Model
COTS	commercial off-the-shelf
DBMS	Data Base Management System
DFARS	Defense Federal Acquisition Regulations Supplement
DoD	Department of Defense
DSB	Defense Science Board
FAR	Federal Acquisition Regulations
GPLR	government purpose license rights
GUI	graphical user interface
IE	Improvement Efforts
IR&D	independant research and development
ISO	International Standards Organization
JIAWG	Joint Integrated Avionics Working Group
Lex	lexical analysis program generator
NSIA	National Security Industrial Association
OPD	Organizational Process Development

PAD	Project Application Development
PARC	Palo Alto Research Center
PERT	Program Evaluation and Review Technique
PLD	Product-Line-Based Product and Process Development
RCM	Reuse Capability Model
RFP	request for proposal
ROI	return on investment
RSO	reusable software object
SADT	Structured Analysis and Design Technique
SEI	Software Engineering Institute
SEPG	Software Engineering Process Group
STARS	Software Technology for Adaptable, Reliable Systems
TQM	total quality management
VCOE	Virginia Center of Excellence for Software Reuse and Technology Transfer
YACC	yet another compiler-compiler

GLOSSARY

Abstraction	A description of a collection of things that applies equally well to any one of them.
Activity	A step of a process for producing or evaluating data elements to satisfy objectives supporting that process. An activity comprises other steps.
Asset	Any tangible resource that may apply to the solution of a problem.
Business area	A coherent market characterized by potential customers possessing similar needs.
Business area potential	The level of reuse opportunities possible in an organization's business area.
Business model	The component of a reuse adoption strategy that identifies the business entities involved in an acquisition process and the contractual and financial relationships among them.
Configuration management tools	Tools to track changes and identify baselines in software products, including reusable assets.
Constraint	A limitation on decisions.
Control	Entities that constrain or enable an activity or information used to direct performance of an activity. Relates to SADT.
Critical success factor	Element of the reuse capability assessment model corresponding to an issue most critical to improving reuse capability.
Critical success factor goal	A result to be achieved to satisfy a critical success factor.
Current reuse situation	An organization's present reuse situation. See reuse situation.

Customer	The person or organization that specifies the requirements for, accepts, and authorizes payment for a product.
Data element	(1) A piece or collection of information that is used as an input or output of an activity. (2) A collection of information fundamental to a system.
Decision	A choice among allowable alternatives.
Design	The process of defining the software architecture, components, modules, interfaces, test approach, and data for a software system to satisfy specified requirements.
Developer	The person or organization that produces or maintains a product.
Domain	(1) A coherent business area. (2) A product family and an associated production process supporting a product line. (3) A functional area covered by a family of systems or across systems in which similar software requirements exist.
Domain analysis	The study and formalization of domain knowledge. The expertise in a business area is formalized to create standards for problem descriptions and corresponding solutions.
Domain assessment	A method used by an organization to qualitatively estimate its business area potential.
Domain Assessment Model	The set of business environment and product factors that influence an organization's reuse opportunities.
Entrance criteria	Conditions that must be met before an activity can be started.
Environment approach	The component of a reuse adoption strategy that defines the tools to be used to support the software process.
Exit criteria	Conditions that must be met before an activity can be considered successfully completed.
Family	A set of things with enough in common that it pays to consider their common characteristics before noting specific properties of instances.

Generators	(1) Tools that provide automated assistance for generating specific environment components, parts, or harnesses. (2) Tools that transform application-oriented specification languages into executable code.
Goal	A specific, time-related, measurable target.
Implementation stage	Element of the reuse capability implementation model representing a possible operational steady state in an organization's reuse practice. It is defined in terms of critical success factor goals.
Improved reuse situation	An organization's reuse situation resulting from a reuse program. See reuse situation.
Input	Information used and transformed in an activity. Relates to SADT.
Instantiation	Creating a thing from a representation of an abstraction denoting a set of such things.
Life cycle	A sequence of distinct states of an entity beginning with its initial conception and ending when it is no longer available for use.
Mechanism	Roles responsible for performing an activity and methods that can be used to perform an activity. Relates to SADT.
Metaprogramming	A method for creating abstract components and extracting concrete components from them.
Method	Guidance and criteria that prescribe a systematic, repeatable technique for performing an activity.
Methodology	An integrated body of principles, practices, and methods that prescribes the proper performance of a process.
Model	A representation of a thing from which analysis provides approximate answers to designated questions about the thing itself.
Objective	The intended or desired result of a course of action.

Organizational approach	The component of a reuse adoption strategy that defines which organizations will be affected by the reuse program.
Organizational objective	What an organization intends to achieve through the conduct of its business.
Output	Information produced in an activity. Relates to SADT.
Plan	A designation of tasks and resource allocations for accomplishing a specified objective.
Potential reuse opportunity	The set of reuse opportunities that will result in actual reuse when exploited.
Process	A partially ordered set of steps intended to accomplish specified objectives.
Process approach	The component of a reuse adoption strategy that defines the processes, methods, and standards that will be used.
Product	The aggregation of all work products resulting from a process or activity.
Product approach	The component of a reuse adoption strategy that defines the end products that will be affected by the reuse effort.
Product line	A collection of existing and potential products that address a designated business area.
Program	(1) An aggregation of software components that, when integrated with hardware, operates as a unit. (2) A directed, funded effort to acquire, develop, or maintain a product.
Project	An undertaking requiring concerted effort, which is focused on developing or maintaining a specific product. Typically, a project has its own funding, cost accounting, and delivery schedule.
Reengineering	The modification of software to improve its maintainability, reusability, or evolvability.

Repository	An organized, shared collection of information about business and software. A base of knowledge that can support software development and maintenance.
Requirements analysis	The process of studying user needs to arrive at a definition of system or software requirements.
Reusable asset	A tangible resource that is acquired or developed for the solution of multiple problems.
Reuse	The use of an asset in the solution of different problems or different versions of a problem.
Reuse action plan	A plan for implementing a reuse practice based on a reuse adoption strategy.
Reuse adoption goal	A measurable result toward which the reuse program is directed to achieve the reuse adoption objectives.
Reuse adoption objective	What an organization intends to achieve through its reuse program.
Reuse Adoption process	A defined set of activities to incorporate the practice of software reuse as a permanent part of an organization's culture and way of doing business—a process to institutionalize reuse.
Reuse adoption strategy	A plan of action to achieve the reuse adoption goals and objectives.
Reuse agent	(1) A defined role of the Reuse Adoption process. (2) An individual or group that is empowered to plan and implement a reuse program.
Reuse capability	The range of expected results in reuse effectiveness, proficiency, and efficiency that can be achieved by an organization's process.
Reuse capability assessment	A method used by an organization to gain an understanding of its current reuse capability and to identify potential opportunities for improving its capability.
Reuse capability assessment model	Component model of the RCM used in a reuse capability assessment.

Reuse capability implementation model	Component model of the RCM used by an organization to plan the implementation of its reuse program. Specifically, it is used to establish reuse adoption goals.
Reuse Capability Model (RCM)	The set of organizational and process factors that influence an organization's reuse capability.
Reuse champion	(1) A defined role of the Reuse Adoption process. (2) An individual or group that advocates and supports a reuse program.
Reuse-driven software process	A process that describes activities that create or utilize reusable assets.
Reuse economics model	A general framework for making estimates and decisions about the economic desirability of reusing software. It covers costs, productivity, ROI, incremental funding, and quality.
Reuse effectiveness	The ratio of the value of actual reuse opportunities exploited to the value of potential reuse opportunities.
Reuse efficiency	The ratio of reuse benefit to reuse costs.
Reuse opportunity	An occasion when an existing asset or one to be developed for reuse may satisfy a current or anticipated need.
Reuse proficiency	The ratio of the actual reuse opportunities exploited to the organization's targeted reuse opportunities.
Reuse program	An organizational effort to institutionalize and improve the practice of reuse in the organization.
Reuse program plan	A plan for implementing a reuse program. It identifies what is to be accomplished with the reuse program, how the program will be conducted, and who will participate.
Reuse situation	The state of affairs in an organization and its environment with respect to the practice of reuse. It includes characteristics of the organization's process, methods, and tools; its culture; relationships with its customers, suppliers, and competitors; its legal and contractual environment; and so on.

Risk	A potential for incurring undesirable results.
Risk aversion strategy	An approach to change a risk by reducing the loss, reducing the uncertainty, or increasing the choices available that present less risk.
Specialization	Constraining an abstraction to denote a subset.
Specification	A document that prescribes, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system or system component.
Sponsor	(1) A defined role of the Reuse Adoption process. (2) An individual or group that authorizes and reinforces a reuse program.
Step	An activity or an unelaborated action.
Tailor	Modify for a purpose; may set parameter values or instantiate for specific reuse, or, conversely, parameterize or abstract for improved reusability. It includes specializations, generalizations, or conversions, such as tuning to improve performance or quality, customization to improve applicability to specific contexts, or extension to improve conformance to requirements.
Targeted reuse opportunity	The set of reuse opportunities toward which an organization directs its effort implicitly and explicitly. These opportunities may or may not result in actual reuse when exploited.
Task	A work assignment subject to management accountability to accomplish a specified objective.
Transition approach	The component of a reuse adoption strategy that defines how and when the changes to the organization and its policies, procedures, processes, methods, standards, and tools will be accomplished.
User	(1) A defined role of the Reuse Adoption process. (2) An individual or group that uses the adopted reuse technologies.
Validation	The evaluation of a work product to determine whether it satisfies customer needs.

Verification	The evaluation of a work product to determine whether it meets its specification.
Work product	Any configuration-managed artifact that is the embodiment of some data element.

REFERENCES

Barnes, B., and T. Bollinger 1991 — Making Reuse Cost-Effective. *IEEE Software* 8, 1:13-24.

Basili, V., and D. Weiss 1984 — A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering* SE-10:6.

Benson, D. 1991 — "ADGE Systems Architecture Analysis." In *Domain Analysis Workshop*, SPC-91186-MC. Herndon, Virginia: Software Productivity Consortium, September 26-27, 1991.

Boehm, B.W. 1988 — A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 21:61-72.

1989 — *Tutorial: Software Risk Management*. Washington, D.C.: IEEE Computer Society Press.

Carr, M., S. Konda, I. Monarch, F.C. Ulrich, and C. Walker 1993 — *Taxonomy-Based Risk Identification*, CMU/SEI-93-TR-6. Pittsburgh, Pennsylvania: Software Engineering Institute.

Central Archive for Reusable Defense Software (CARDS) 1992 — *Acquisition Handbook, Central Archive for Reusable Defense Software*. Reston, Virginia: Paramax Systems Corporation.

1993a — *Franchise Plan, Central Archive for Reusable Defense Software*. Reston, Virginia: Paramax Systems Corporation.

1993b — *Proceedings: Software Reuse Legal Issues Workshop*. Reston, Virginia: Paramax Systems Corporation.

Charette, R. 1989 — *Software Engineering Risk Analysis and Management*. New York, New York: Intertext Publications, McGraw-Hill.

Cusumano, M. 1991 — *Japan's Software Factories: A Challenge to U.S. Management*. New York, New York: Oxford University Press, Inc.

Defense Federal Acquisition Regulation Supplement (DFARS) 1990

Defense Federal Acquisition Regulations Supplement. 252.227-7013, Rights in Technical Data and Computer Software; 252.227-7022,7023, Government Rights (Unlimited); 252.227-7026, Deferred Delivery of Technical Data or Computer Software; 252.227-7032, Rights in Technical Data or Computer Software (Foreign), April 1990.

Defense Science Board (DSB) Task Force on Military Software 1987

Report of the Defense Science Board Task Force on Military Software. Washington, D.C.: Office of the Under Secretary of Defense for Acquisition, September 1987.

DeVecchio, W. Jay 1990

Legal Barriers to Reuse and Liability, SPC-90093-N. Video presentation to the Software Productivity Consortium. Herndon, Virginia: Software Productivity Consortium.

Elkin, R.M., D.A. Gardner, and R.B. Ireland 1991

"Pathfinding, A Practical Approach to Risk Management." In *Proceedings, Software Risk Management Conference, October 7-9, 1991.* Pittsburgh, Pennsylvania: Software Engineering Institute.

Federal Acquisition Regulations (FAR) 1988

Federal Acquisition Regulations. Code of Federal Regulations, Title 48. Washington, D.C.: Office of the Federal Register, National Archives and Records Administration.

Forman, E.H., T.L. Saaty, M.A. Selly, and R. Whitaker 1990

Expert Choice. Pittsburgh, Pennsylvania: Expert Choice, Inc.

Frakes, W., and C. Fox 1993

Software Reuse Survey Report. Sterling, Virginia: Software Engineering Guild.

Freemon, B.W., and R.G. Crispen 1992

"Testing a Technology for Reuse." *14th Interservice/Industry Training Systems and Education Conference.* San Antonio, Texas: National Security Industrial Association, November 2-4, 1992.

Gaffney, J., Jr., and R. Cruickshank 1992

"A General Economics Model of Software Reuse." *14th International Conference on Software Engineering.* Melbourne, Australia: Institute of Electrical and Electronics Engineers.

Global Protection Against Limited Strikes (GPALS) 1992a

GPALS Software Reuse Action Plan. Washington, D.C.: Department of Defense, Strategic Defense Initiative Organization, February 1992.

1992b

GPALS Software Reuse Strategy. Washington, D.C.: Department of Defense, Strategic Defense Initiative Organization, February 1992.

1992c — *GPALS Software Reuse: Executive Overview*. Washington, D.C.: Department of Defense, Strategic Defense Initiative Organization, July 1992.

Hooper, J., and R. Chester 1990 — *Software Reuse Guidelines*, ASQB-GI-90-015. Atlanta, Georgia: Georgia Institute of Technology.

Institute for Defense Analysis 1991 — *Proceedings of the Workshop on Legal Issues in Software Reuse*, D-1004. Alexandria, Virginia: Institute for Defense Analyses.

Jaworski, A., F. Hills, T. Durek, S. Faulk, and J. Gaffney 1990 — *A Domain Analysis Process; Interim Report—Domain Analysis Project*, DOMAIN_ANALYSIS-90001-N. Herndon, Virginia: Software Productivity Consortium.

Jensen, J., H. Stewart, and P. Whittington 1992 — "Successful Experience With the AdaSAGE Reusable Component Library." In *Proceedings of Tri-Ada '92*. Orlando, Florida: ACM.

Joint Integrated Avionics Working Group (JIAWG) 1990 — *Contract Elements for Software Reuse*. Prepared by JIAWG Software Task Group, June 13, 1990.

Kang, K., S. Cohen, J. Hess, W. Novak, and S. Peterson 1990 — *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, CMU/SEI-90-TR-21. Pittsburgh, Pennsylvania: Software Engineering Institute.

Marca, D., and C. McGowan 1987 — *Structured Analysis and Design Technique*. New York, New York: McGraw-Hill.

Martin, R., G. Jackoway, and C. Ranganathan 1991 — "Software Reuse Across Continents." In *Proceedings of the 4th Annual Workshop on Software Reuse*. Herndon, Virginia: Center for Innovative Technology.

Mettala, E., and M.H. Graham 1992 — "The Domain-Specific Software Architecture Program." In *Software Technology Conference, 1992 Proceedings*. Los Angeles, April 28–30, 1992.

National Security Industrial Association (NSIA) 1990 — *The Business Issues Associated With Software Reuse.* Report of a study conducted by the Software and C^3IC Committee's Joint Sub-Committee on Software Reuse. Washington, D.C.: National Security Industrial Association, December 15, 1990.

Over, J. 1992 — "Software Process Asset Library." In *Software Technology Conference, 1992 Proceedings*. Los Angeles, April 28–30, 1992.

Paulk, M., B. Curtis, M.B. Chrissis, and C. Weber 1993 — *Capability Maturity Model for Software*, CMU/SEI-93-TR-24. Pittsburgh, Pennsylvania: Software Engineering Institute.

Prieto-Diaz, R. 1991 — Making Software Reuse Work: An Implementation Model. *ACM SIGSoft Software Engineering Notes.* July 1991.

Prieto-Diaz, R., and G. Arango 1991 — *Domain Analysis and Software Systems Modeling.* Los Alamitos, California: IEEE Computer Society Press.

Przybylinski, S., P. Fowler, and J. Maher 1991 — *Software Technology Transition.* Tutorial presented at the 13th International Conference on Software Engineering. Pittsburgh, Pennsylvania: Software Engineering Institute.

Redwine, S., Jr., C. DelFosse, and W. Spencer 1992 — 1991 Technology Transfer at the Software Productivity Consortium. *American Programmer,* 11-19.

Reuse Acquisition Action Team (RAAT) 1992 — Minutes of the Reuse Acquisition Action Team. Under the ACM SIG Ada Reuse Working Group, January 1992.

Reuse Library Interoperability Group 1992 — *Reuse Library Interoperability Group (RIG) Mission and Organization.* Arlington, Virginia: Applied Expertise Incorporated.

Royce, Walker 1990 — "Reusable Ada Components for Large, Distributed Multi-Task Networks: NAS." In *Aerospace Computing.* TRW, January 1990.

Saaty, T. 1980 — *The Analytic Hierarchy Process.* New York, New York: McGraw-Hill International.

Samuelson, P. 1986 — *Toward a Reform of the Defense Department Software Acquisition Policy,* CMU/SEI-86-TR-1. Pittsburgh, Pennsylvania: Software Engineering Institute.

Samuelson, P., and K. Deasy 1987 — *The Effect of Software Support Needs on the Department of Defense Software Acquisition Policy, Part 1: A Framework for Analyzing Legal Issues,* CMU/SEI-86-TR-1. Pittsburgh, Pennsylvania: Software Engineering Institute.

Samuelson, P., K. Deasy, and A. Martin 1986 — *Proposal for a New "Rights in Software" Clause for Software Acquisitions by the Department of Defense,* CMU/SEI-86-TR-2. Pittsburgh, Pennsylvania: Software Engineering Institute.

Software Productivity Consortium 1992a — *Introducing Systematic Reuse to the Command and Control Systems Division of Rockwell International*, SPC-92020-N. Herndon, Virginia: Software Productivity Consortium.

1992b — *Process Engineering With the ESP Model*, SPC-92079-C. Herndon, Virginia: Software Productivity Consortium.

1992c — *Software Measurement Guidebook*, SPC-91060-CMC. Herndon, Virginia: Software Productivity Consortium.

1992d — State of Reuse Practice Study. Notes by Fred Hills from interviews with software developers. February and March 1992.

1993a — *Managing Process Improvement: A Guidebook for Implementing Change*, SPC-93105-CMC. Herndon, Virginia: Software Productivity Consortium.

1993b — *Reuse-Driven Software Processes Guidebook*, SPC-92019-CMC. Herndon, Virginia: Software Productivity Consortium.

1993c — *Reuse Economics Spreadsheet Model User Manual*, SPC-91158-CMC. Herndon, Virginia: Software Productivity Consortium.

1993d — *Using New Technologies: A Technology Transfer Guidebook*, SPC-92046-CMC. Herndon, Virginia: Software Productivity Consortium.

Software Technology for Adaptable, Reliable Systems (STARS) 1991 — *Reusable Software Acquisition: Current FAR and Budget/Finance Requirements*, GR-7670-1226(NP). Reston, Virginia: Unisys Defense Systems.

1992a — *Proceedings of STARS '92*. Arlington, Virginia: STARS Technology Center.

1992b — *STARS Reuse Concepts—Conceptual Framework for Reuse Processes*. Hanscom Air Force Base, Massachusetts: U.S. Air Force Material Command, Electronic Systems Center.

1993 — *Reuse Strategy Model: Planning Aid for Reuse-Based Projects*. Hanscom Air Force Base, Massachusetts: U.S. Air Force Material Command, Electronic Systems Center.

U.S. Air Force
1988

Acquisition Management: Software Risk Abatement, AFSC/AFLCP 800-45. Air Force Systems Command and Air Force Logistics Command.

U.S. Department of Defense (DoD)
1988

Military Standard: Defense System Software Development, DOD-STD-2167A. Washington, D.C.: Department of Defense.

1992a

Common ATCCS Support Software (CASS) Architecture Description and Operational Concept, ACCS-A1-302-002. Fort Monmouth, New Jersey: U.S. Army CECOM.

1992b

Deriving and Managing Functional Commonality for Software Reuse: An RNTDS Experience, Jim Ardis. *Crosstalk*, February 1992.

1992c

DoD Software Reuse Initiative: Vision and Strategy. Washington, D.C.: Department of Defense.

Wartik, S., and R. Prieto-Diaz
1992

Criteria for Comparing Reuse-Oriented Domain Analysis Approaches. *International Journal of Software Engineering and Knowledge Engineering*, September 1992.

INDEX