CDRL: B002
28 February 1994

UNISYS

Engineer's Handbook
Central Archive for Reusable Defense Software (CARDS)

Informal Technical Data

Central Archive for Reusable Defense Software

STARS-VC-B002/002/00
28 February 1994

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	28 February 1994	Informal Technical Report

4. TITLE AND SUBTITLE

Engineer's Handbook
Central Archive for Reusable Defense Software (CARDS)

5. FUNDING NUMBERS

F19628-93-C-0130

6. AUTHOR(S)

Chris Fronczek
George Jackelen
Stephen Riesbeck

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

Unisys Corporation
10210 Sunrise Valley Drive
Reston, VA 22091

8. PERFORMING ORGANIZATION REPORT NUMBER

STARS-VC-B002/002/00

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

Department of the Air Force
Electronics Systems Center
Hanscom AFB, MA 01731-2816

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

B002

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution "A"

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This Engineer's Handbook was developed under the Central Archive for Reusable Defense Software (CARDS) Program to help facilitate advances in software reuse techniques and technologies. This document provides guidance to government System Program Office (SPO) Engineers on envisioned changes to their duties and responsibilities as domain-specific software reuse becomes incorporated into mainstream DoD system/software acquisition and engineering processes.

14. SUBJECT TERMS

15. NUMBER OF PAGES
51

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	SAR

CDRL: B002
28 February 1994

INFORMAL TECHNICAL REPORT
For The
SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS
(STARS)

Engineer's Handbook
Central Archive for Reusable Defense Software
(CARDS)

STARS-VC-B002/002/00
28 February 1994

Data Type: Informal Technical Data
Contract NO. F19628-93-C-0130
Line Item 0002AB

Prepared for:

Electronic Systems Center
Air Force Material Command, USAF
Hanscom AFB, MA 01731-2816

Prepared by:

Azimuth, Inc.
Electronic Warfare Associates, Inc.
Strictly Business Computer Systems, Inc.
under contract to
Unisys Corporation
12010 Sunrise Valley Drive
Reston, VA 22091

CDRL: B002
28 February 1994

INFORMAL TECHNICAL REPORT
For The
SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS)

Engineer's Handbook
Central Archive for Reusable Defense Software
(CARDS)

STARS-VC-B002/002/00
28 February 1994

Data Type: Informal Technical Data

Contract NO. F19628-93-C-0130
Line Item 0002AB

Prepared for:

Electronic Systems Center
Air Force Material Command, USAF
Hanscom AFB, MA 01731-2816

Prepared by:

Azimuth, Inc.
Electronic Warfare Associates, Inc.
Strictly Business Computer Systems, Inc.
under contract to
Unisys Corporation
12010 Sunrise Valley Drive
Reston, VA 22091

CDRL: B002
28 February 1994

Data Reference: STARS-VC-B002/002/00
INFORMAL TECHNICAL REPORT
Engineer's Handbook
Central Archive for Reusable Defense Software
(CARDS)

Distribution Statement "A"
per Dod Directive 5230.24
Approved for public release, distribution is unlimited

Developed by: Azimuth, Inc., Electronic Warfare Associates, Inc., and Strictly Business Computer Systems, Inc. under contract to Unisys Corporation

This document, developed under the Software Technology for Adaptable, Reliable Systems (STARS) program, is approved for release under Distribution "A" of the Scientific and Technical Information Program Classification Schema (DoD Directive 5230.24) unless otherwise indicated by the U.S. Sponsored by the U.S. Advanced Research Projects Agency (ARPA) under contract F19628-93-C-0130 the STARS program is supported by the military services with the U.S. Air Force as the executive contracting agent. The information identified herein is subject to change. For further information, contact the authors at the following mailer address: delivery@stars.reston.paramax.com no event shall the Government (prime contractor or its subcontractor) be liable for any special, indirect, or consequential damages or any damages whatsoever resulting from the loss of use, data, or profits, whether in action of the contract, negligence, or other totious action, arising in connection with the use or perfomance of this document.

CDRL: B002
28 February 1994

Data Reference: STARS-VC-B002/002/00
INFORMAL TECHNICAL REPORT
Engineer's Handbook
Central Archive for Reusable Defense Software
(CARDS)

Principal Author(s):

Chris Fronczek *Date*

George Jackelen *Date*

Stephen Riesbeck *Date*

Approvals:

System Architect: *Kurt Wallnau* *Date*

Program Manager: *Lorraine Martin* *Date*

(Signatures on File)

Data Reference: STARS-VC-B002/002/00
INFORMAL TECHNICAL REPORT
Engineer's Handbook
Central Archive for Reusable Defense Software
(CARDS)

PREFACE

This release of the Engineer's Handbook would not have been possible without the guidance and efforts of authors of earlier versions. Sincere thanks to the following individuals for their valuable contributions: Terry Huber of DSD Laboratories, Paul Kogut of Unisys, Jay Reddy of D.N. American, Robert Saisi of DSD Laboratories, Kerrin Smith of Azimuth, Charlie Snyder, formerly of Unisys, and Kurt Wallnau of Unisys.

This Handbook is a "living" document, and will be periodically updated. Comments on this document are welcomed and encouraged.

CDRL: B002
28 February 1994

Data Reference: STARS-VC-B002/002/00
INFORMAL TECHNICAL REPORT
Engineer's Handbook
Central Archive for Reusable Defense Software
(CARDS)

ABSTRACT

This Engineer's Handbook was developed under the Central Archive for Reusable Defense Software (CARDS) Program to help facilitate advances in software reuse techniques and technologies. This document provides guidance to government System Program Office (SPO) Engineers on envisioned changes to their duties and responsibilities as domain-specific software reuse becomes incorporated into mainstream DoD system/software acquisition and engineering processes.

The intended audience of this Handbook is SPO Engineers who are responsible for pre-Request for Proposal (RFP) engineering activities, proposal evaluation, monitoring of engineering activities after a contract is awarded, and monitoring of ongoing sustaining engineering efforts (or maintenance) of fielded products.

To fully utilize the concepts in this Handbook, it is recommended that the reader be familiar with software development techniques and methodologies, existing government regulations and standards (such as DOD-STD-2167A, MIL-STD-499, MIL-STD-1521B, and emerging DOD-STD-498/SDD), and the acquisition process.

Table of Contents

Appendices

1 INTRODUCTION

In an effort to improve software quality and cost effectiveness, the Department of Defense (DoD) is actively endorsing software reuse, the process of implementing new systems by using existing software products and information [DOD92]. DoD advocates domain-specific software reuse, which focuses on a well-defined functional area called a domain. Domain-specific reuse provides a framework for generating systems within that area and also provides greater leverage of reusable assets.

As emerging principles and technology associated with domain-specific software reuse become incorporated into DoD system/software acquisition and engineering processes, the duties and responsibilities of personnel involved in system/software acquisition, development, and maintenance will change. This Engineer's Handbook focuses on changes in the duties and responsibilities of System Program Office (SPO) Engineers.

An example of envisioned change is that SPO Engineers will interface with two "new" activities called domain management and domain engineering. For example, a contract will no longer be issued to develop a stand-alone, "built from scratch" system which does not consider previous system/software engineering efforts, but instead will be expected to leverage domain engineering technology. Domain engineering technology can include reuse libraries, domain/architectural models, reuse tools and processes, reuse guidelines and handbooks, and so forth. SPO Engineers will be expected to extensively reuse existing components (also referred to as assets or artifacts), and will be expected to create new components for contribution to the domain technology base. This will require changes in the way SPO Engineers perform contractor technical oversight; in addition, there will be significant differences in their duties during preparation of Request for Proposals (RFPs) and proposal evaluation.

1.1 Purpose

The purpose of this Engineer's Handbook is to provide SPO Engineers with:

- expected changes in duties and responsibilities, as provisions for domain-specific software reuse are incorporated into system/software acquisition and engineering processes,
- examples of how these expected changes will impact their current duties, and
- recommendations for effective use of domain engineering technology.

The intended audience of this Handbook is SPO Engineers who are responsible for:

- pre-RFP engineering activities,
- proposal evaluation,

- monitoring of engineering activities after a contract is awarded, and
- monitoring of ongoing sustaining engineering efforts (or maintenance) of fielded products.

The term "SPO Engineers" is used in this Handbook to include engineers working at or below a program management level in a DoD organization (see Figure 1-1, below, and Figure 2-2, page 10). This may include engineers working at a Central Design Activity (CDA) level. (A Central Design Activity is generally a large DoD activity, with an average of 50+ personnel, associated with software design, development, re-engineering, maintenance, systems integration, and common support activities. Common support functions include workload control, systems development guidance and tools, data administration, software repositories, and application development process and assessment improvement programs.) In general, these engineers are sometimes referred to as "government" engineers.

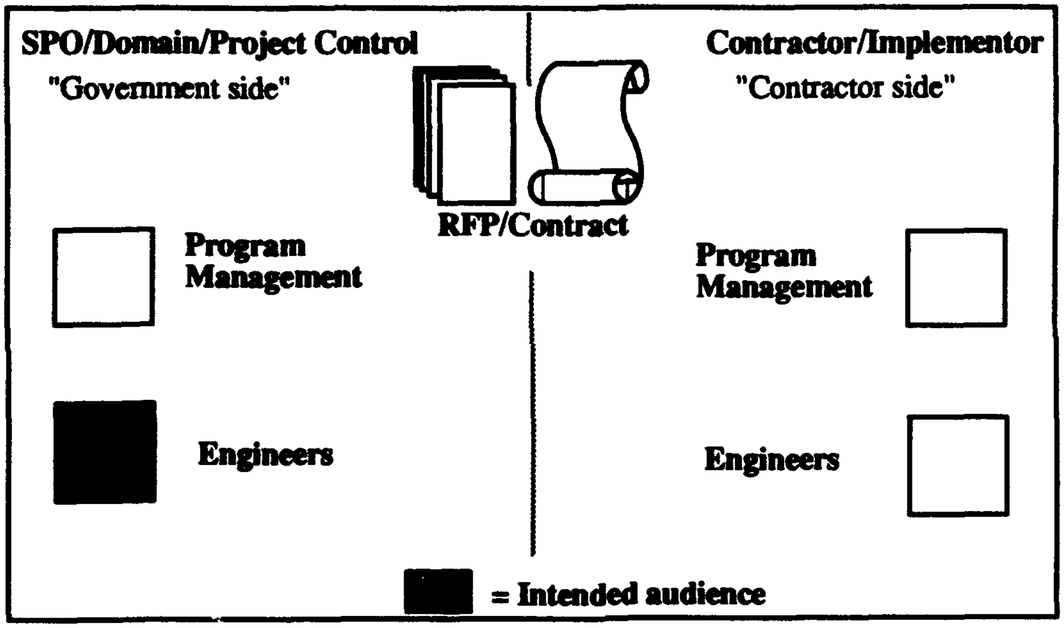

Figure 1-1 Intended Audience for the Engineer's Handbook

1.2 Scope

This Handbook is intended to increase awareness of envisioned changes in the activities of SPO Engineers resulting from DoD adoption of domain-specific reuse techniques; it provides a range of information on reuse topics such as domain engineering, domain analysis, and system development. However, this Handbook is not designed to function as a tutorial on domain-specific software reuse (see section 1.4.1 *Relationship to Other Documents*). In-depth examination of issues such as domain creation, domain management, and acquisition strategies are outside the scope of this document.

It is important to note that the focus of this document is on envisioned changes that will occur in SPO Engineers' duties. This Handbook describes these changed duties both from the perspective of a hypothetical, established domain engineering activity which SPO Engineers will interface with, and from the perspective of a DoD organization in transition to domain engineering, with an emphasis on the incremental, systematic changes that will be occurring during this transition.

This version of the CARDS Engineer's Handbook replaces the previous version (STARS-VC-B002/001/00), dated 30 September 1993 [CARDS93f].

1.3 Background and Assumptions

The Central Archive for Reusable Defense Software (CARDS) Program is a concerted DoD effort to transition advances in the techniques and technologies of domain-specific software reuse into mainstream DoD software procurements. The CARDS Program goals are to:

- produce, document, and propagate techniques to enable domain-specific reuse throughout the DoD,
- develop and operate a domain-specific library system and necessary tools,
- develop a Franchise Plan which provides a blueprint for institutionalizing domain-specific, library-centered reuse throughout the DoD,
- implement the Franchise Plan with users and provide a tailored set of services to support reuse.

In addition to CARDS, there are various efforts in the software community currently addressing organizational, business, and technical aspects of software reuse.

- The Defense Information Systems Agency (DISA) has, among other things, begun several process development initiatives in the domain analysis and reuse metrics collection areas. To date, DISA has defined the domains which exist within DoD [DISA92], developed guidelines for conducting domain analyses [VITA92], and developed a method for defining and collecting reuse-related metrics.
- The Software Engineering Institute (SEI) has developed and applied a domain analysis process (FODA, Feature-Oriented Domain Analysis) and currently is developing a reuse-based software development methodology that is based on DOD-STD-2167A and focuses on identifying and applying reusable resources [KANG92].
- The Advanced Research Projects Agency (ARPA) sponsored Software Technology for Adaptable, Reliable Systems (STARS) Program has been developed to increase software productivity, reliability, and quality by integrating software processes and reuse concepts. Specifically, the STARS reuse concepts provide a conceptual foundation, framework, and requirements for reuse technology processes and supporting tools. Their approach is generic with respect to organizations, software engineering

methodologies, technologies, and environments. It advocates the use of domain specific software architectures (DSSAs), which provide a framework for creating components and constructing systems within a domain [STARS92].

The above efforts (and others) have captured and produced a wealth of knowledge which can assist SPO Engineers. In addition to documentation currently available, these efforts will continue to provide SPO Engineers and others with the latest advances in the rapidly changing and emerging field of software reuse.

To fully utilize the concepts in this Handbook, it is recommended that SPO Engineers be familiar with software development techniques and methodologies, existing government regulations and standards (such as DOD-STD-2167A, MIL-STD-499, and MIL-STD-1521B), and the acquisition process. Section 1.4.1, Relationship to Other Documents, outlines additional sources for consultation.

1.4 Document Overview

1.4.1 Relationship to Other Documents

The domain-specific reuse knowledge gained during the CARDS effort will be conveyed to the DoD and software reuse communities via a Franchise Plan and three sets of documents: Reuse Adoption Handbooks, CARDS Command Center Library development and operation related documents, and training and educational material.

The Franchise Plan provides a description of reuse processes and instructions for tailoring development processes to incorporate domain-specific reuse. It describes, in precise steps, a scenario for an organization to establish a domain-specific reuse capability.

The Reuse Adoption Handbooks consist of the Engineer's, Component Provider's and Tool Vendor's, Acquisition, and Direction Level Handbooks. Together these four handbooks address software development, program management, and executive planning within the context of software reuse. As a complement to the Engineer's Handbook, the Component Provider's and Tool Developer's Handbook provides government software developers and industry vendors with guidance for building/creating domain-specific reusable components and tools. The goal of the Component Provider's and Tool Developer's Handbook is to stimulate the development and commercialization of large scale components and tools for vertical domains [CARDS93c]. The Acquisition Handbook assists government Program Managers and their support staff in incorporating software reuse into the acquisition and maintenance portions of the life-cycle process. The Acquisition Handbook provides guidance in planning the acquisition strategy, contract award, managing the effort, and follow-on support [CARDS93a]. The Direction Level Handbook offers a framework to assist government acquisition executives in establishing plans to manage software reuse across their systems. Importance is placed on the policy and business issues (e.g., regulations, incentives, funding, cost/benefit, education and training, and ownership of components) that act as the support structure for reuse [CARDS92a].

Although some of the CARDS library development and operation documents are specific to the CARDS Command Center Library, they can be used by other organizations interested in technical and operational processes used by a domain-specific reuse library in the command and control domain. These CARDS documents address the library's operations procedures [CARDS94c and CARDS94d], library development issues and methodologies [CARDS93b], the technical concepts [CARDS94b], project management plans, as well as describing the Command Center Library model [CARDS93h].

The CARDS training effort includes a training plan [CARDS94a], course outlines, and sample course materials. These courses include *Introduction To Software Reuse for System/Software Engineers* and *Application Engineering With Domain-Specific Reuse.* They are geared to educate the software professional and support the reduction of cultural barriers to reuse. They can be tailored to meet the needs of varying audiences.

Also, this Engineer's Handbook has been developed to function as a complement to additional publications. Specifically, these publications include the following:

- The STARS *Conceptual Framework for Reuse Processes* (CFRP) defines a context for considering reuse-related software development processes, their interrelationships, and their composition and integration with each other and with non-reuse-related processes to form reuse-oriented life cycle process models [STARS92].
- *SDP-2000: A Guide to Project Implementation of Megaprogramming* describes a vision of the software industry as it may exist under megaprogramming (when superior practices in software engineering are synthesized), and describes transition steps that will be required by the government and contractors alike [POORE93].
- *A New Process for Acquiring Software Architecture* outlines a process that can be used to ensure that system acquisitions include attention to the software architecture [SAUNDERS93].

It is recommended that these publications be consulted for additional, in-depth information on issues discussed in this handbook.

1.4.2 Document Organization

This handbook is organized into six chapters and four appendices.

Chapter One, *Introduction*, provides a general introduction to the document.

Chapter Two, *Acquisition Process Summary*, is a brief overview of the DoD acquisition and program management structure, and is designed to establish a common level of understanding for the audience.

Chapter Three, *Domain Engineering*, gives an overview of current concepts in domain engineering, domain management, and domain analysis with emphasis on the impact on DoD procurement.

Chapter Four, *Domain Engineering During the Acquisition Process*, describes the detailed changes in day-to-day activities and responsibilities of pre-contract award SPO Engineering efforts due to the impact of domain-specific reuse.

Chapter Five, *Engineering During the Development Phase*, outlines the detailed changes in day-to-day activities and responsibilities of post-contract award SPO Engineering efforts due to the impact of domain-specific reuse.

Chapter Six, *Sustaining Engineering Efforts*, describes the detailed changes to any sustaining SPO Engineering (maintenance) activities that will result from the impact of domain-specific reuse.

Appendix A is a list of references used in the development of this document, and a list of recommended readings. The recommended reading list is provided for SPO Engineers who may not be familiar with the acquisition process, and does not include information on reuse. For information on reuse, SPO Engineers should consult sources listed in the References section of Appendix A, and sources listed in section 1.4.1 of this document.

Appendix B contains Domain Engineering Evaluation Criteria Examples, designed to help SPO Engineers identify specific data required in proposals to evaluate technical and management reuse approaches, as described in Chapter Four.

Appendix C, *Domain Engineering Pre-Award Survey Suggestions*, outlines a list of suggested questions which SPO Engineers may want to ask bidders in a pre-contract award survey.

Appendix D is a glossary.

2 ACQUISITION PROCESS SUMMARY

2.1 Introduction

This chapter is a brief overview of the DoD acquisition/source selection process to ensure readers understand some of the mentioned terminology and activities. The purpose of this chapter is to provide SPO Engineers with enough information to understand Chapter Four of this handbook and to have an idea of their role during the acquisition process (i.e., prior to contract award). If the reader is familiar with the acquisition process, but not with the impact domain engineering will have, then this chapter may be skimmed.

The described acquisition process is based on the Federal Acquisition Regulation (FAR), Subpart 15.6, *Source Selection for Major Acquisitions* [FAR]. This states that the acquisition process is a procedure through which contracts are awarded as a result of competitive negotiations. Any procedure not involved in sealed bids or auctions, and is competitive, can be considered an acquisition process.

2.2 Acquisition Process Objectives and Procedures

In summary, the government has established objectives and procedures to ensure fair competition for government ordered products. The specific objectives are to maximize competition, minimize the complexity of acquiring new government systems, assure impartial evaluation of offers, and guarantee the selection of the offer in terms of stated requirements.

The FAR describes two types of acquisition processes:

- Formal: Any acquisition process using a:

    1. Specific evaluation group structured and established to evaluate proposals.

    2. Person (Source Selection Authority (SSA)) who is at a management level, above that of the contracting officer, to select a winner.

- Informal: Any acquisition process that is not formal.

The following information is provided (as reference material) for SPO Engineers who need more details than this handbook provides. The FAR does not go beyond a high-level of specification and definition. Most of the details are left to the different government agencies (e.g., DoD, Treasury, and Veterans Affairs) to develop, document, and implement. As a result, DoD has created the Defense FAR Supplement (DFARS). Each service/department has supplemented the DFARS, e.g., AFFARs (Air Force FAR) and AFARs (Army FAR). These service DFARSs are further described in regulations, e.g., USAF Regulation (AFR) 70-15, which set policy, assign authority, and prescribe procedures for solicitations and evaluations of proposals. The next level of regulation, for example the AF, is called the AF Material Command (AFMC) Supplement.

Finally, for example, each AFMC center (e.g., Electronic Systems Center - ESC) has additional supplements to describe their policies and procedures. Figure 2-1 summarizes this hierarchy of regulations. It must be noted that the General Accounting Office and other organizations are looking at possible recommendations to this process to address reuse.

2.3 Organization

Figure 2-2 and the following provide information about the DoD organizational structure for major acquisitions. However, this Engineer's Handbook can also be used for non-major acquisitions. Note that some of the services may use different organizational terminology.

The Program Executive Officers (PEOs) oversee major program execution, screen staff reviews, report only to their Service Acquisition Executive (SAE) for program matters, and review baselines, e.g., cost. For non-major programs, the Air Force uses the term Designated Acquisition Commander (DAC) in place of a PEO.

For major programs, the SPO/Program Manager manages and executes the program, reports to the PEO for program matters, formulates baselines for costs, etc. The Program Manager implements the funding method, type of contract, and determines program requirements.

The contracting officer (CO) is the only individual authorized to enter into contracts on behalf of the government and to direct the contractor's efforts within each contract. This role is broken down into two areas:

- The Procuring Contracting Officer (PCO) performs all negotiations, some legal reviews, and makes contract awards.

- The Administrative Contracting Officer (ACO) performs contract administration (day-to-day contract affairs after contract award) for the PCO (e.g., ensures contractor performs as contractually required) but is not authorized to negotiate. ACOs normally cannot reschedule work or change contract scope. ACOs receive authority from the PCO and are the contractor's single point of contact on contract issues.

The Contracting Officer's Technical Representative (COTR) is someone from the program office (e.g., could be the Program Manager or designate) who works day-to-day with the contractor to ensure technical compliance. If there is a contractual/technical problem, the COTR notifies the ACO or PCO for a ruling and/or action.

The Legal Office works with the Program Manager and PCO to ensure legal compliance by both the contractor and the government.

SPO Engineers ensure the requirements (e.g., for this document, domain engineering requirements) are part of the Request for Proposal (RFP) and contractor's proposals, and implemented by the contractor.

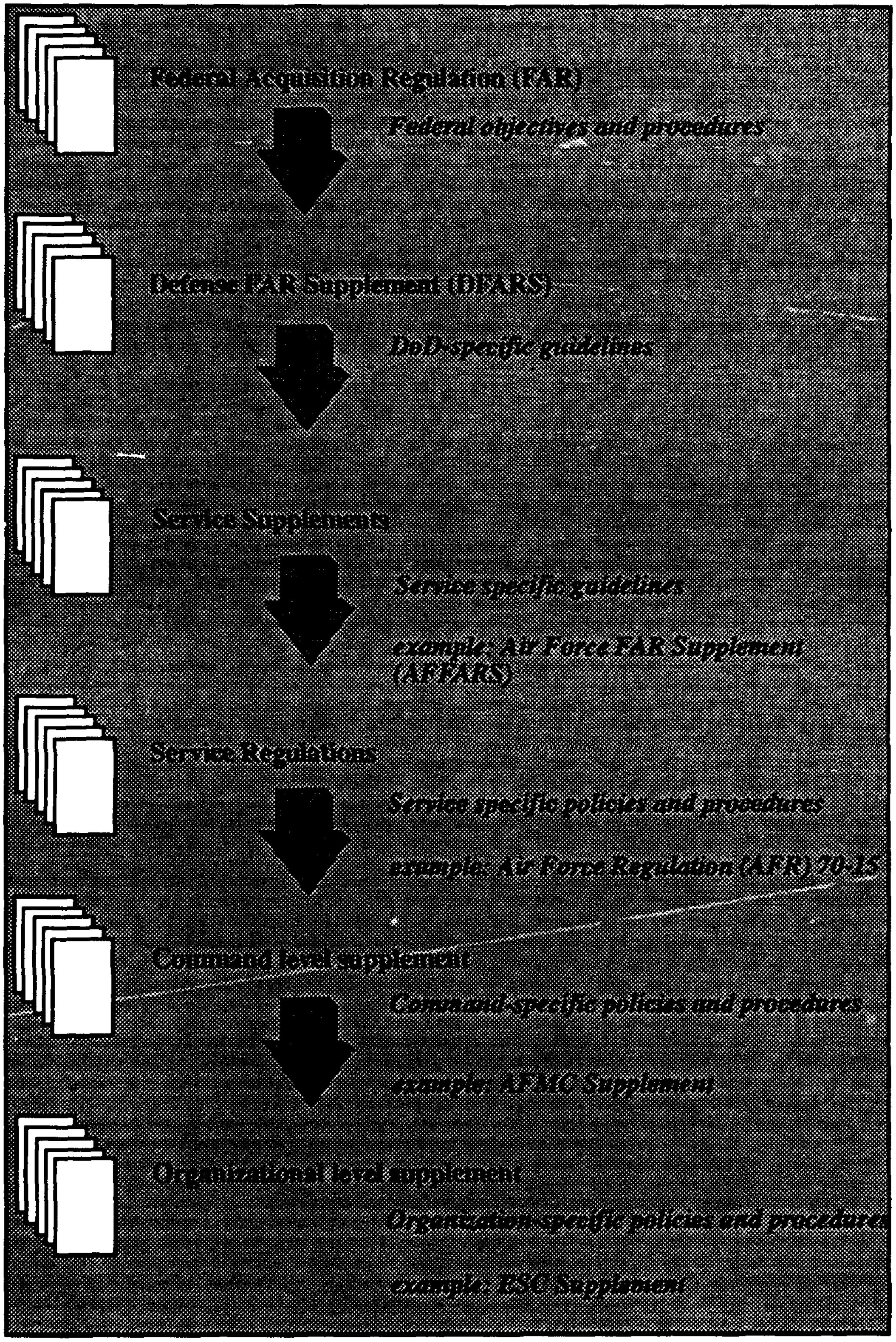

Figure 2-1 Sample Regulations The Air Force Uses To Implement The FAR

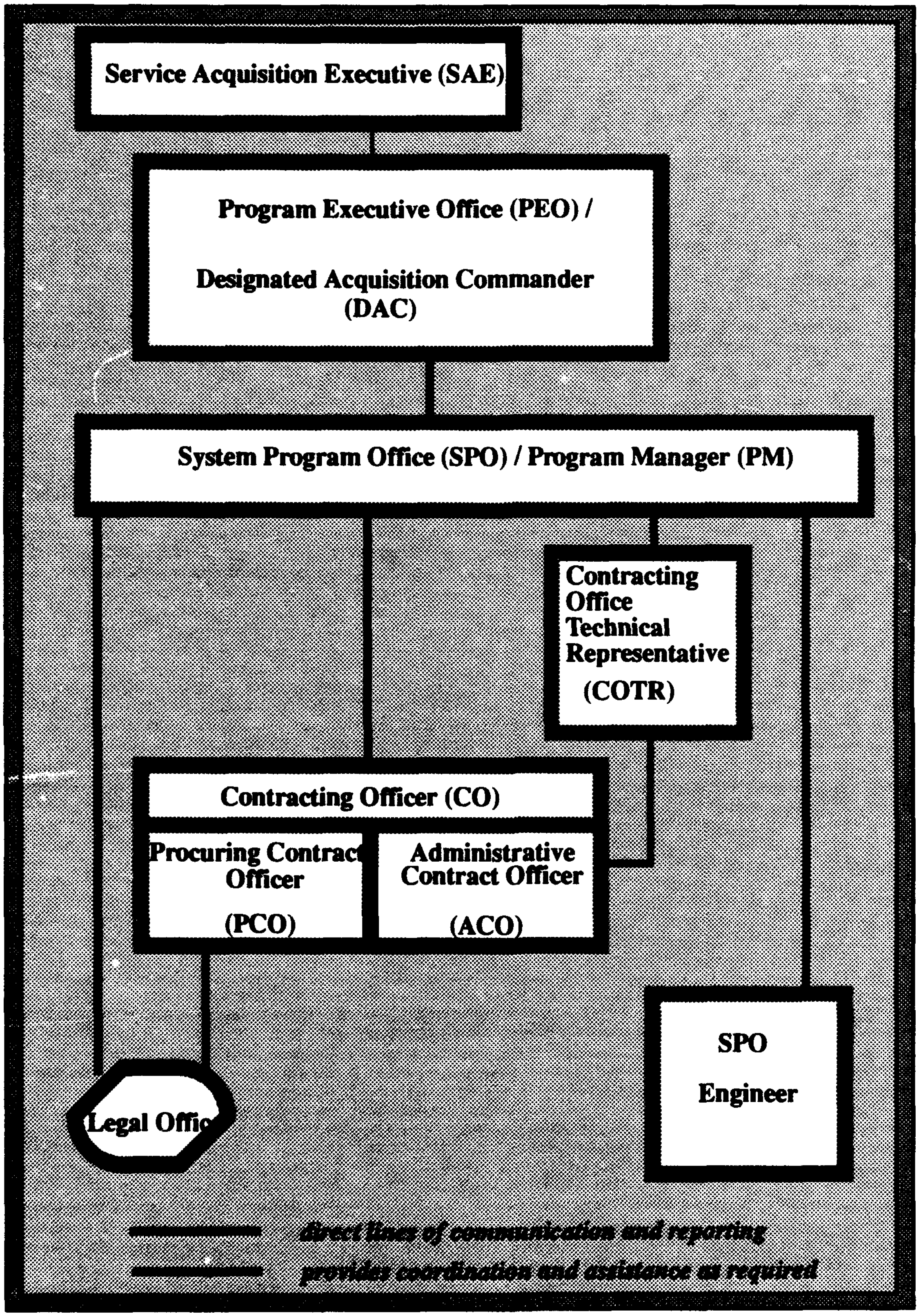

Figure 2-2 DoD Provides A Flexible Organization To Assist The PM In The Acquisition Process

2.4 SPO Engineer Acquisition Process Description

Government acquisition of a major (defined in each service's DFARS supplement in terms of cost/budget) system is characterized by a highly formal and restricted interaction between the government and the bidders/offerers, i.e., contractors who are/may be interested in bidding for a contract/program. Minor procurements may have different interaction requirements.

An acquisition process (Figure 2-3) can consist of 15 chronological steps which can be grouped into four activities. A brief summary is given below. (SPO Engineering activities are emphasized.) As in any acquisition, deviations to the described activities can occur.

2.4.1 Activity I - Pre-Solicitation

Government pre-solicitation activities overlap with several bidder activities, e.g., marketing, soliciting information, and preparing pre-proposals. This activity is in preparation for the RFP that the government is planning to issue to potential bidders.

2.4.1.1 Acquisition Plan

The Acquisition Plan is described in FAR Part 7 and requires planning for every procurement. The Acquisition Plan has three parts: Acquisition Background and Objectives, Plan of Action, and Acquisition Milestones. The emphasis of the Acquisition Plan is on cost considerations and the type of source selection (formal or informal). Specific preparation instructions are provided by the service regulations.

SPO Engineers may be expected to assist in Acquisition Plan preparation in the areas of (see Chapter Four):

- Costing. Is there an added cost to do reuse? What is the cost? What are the cost factors, e.g., start-up versus already existing domain reuse library?

- Trade-off studies. What reuse libraries exist that the contractor can use? What reuse components are available and where? What reuse tools are available and where? [CARDS93c] provides an examination of these issues.

- Risk studies. What companies have experience in the domain and have (or may be able to) implemented reuse? Are there enough reusable components and tools available to require reuse? Will additional money be needed to provide reuse in new components and/or tools?

- Product descriptions. Will a Reuse Implementation Plan (used to describe how the contractor will implement reuse into the developed system) be required?

- Contracting considerations, e.g., use of reuse libraries. Will contractor or government have a separate contract with a reuse library? Is there a need for a contractual incentive award to encourage reuse?

- Make or buy, e.g., how much reuse can be expected. Will reuse be a major or minor part of the contract? How many new components should be developed with reuse built-in? What components are available that need to be modified for reuse?

- Test and evaluation. Will reuse provide any test and evaluation savings? Will reuse accelerate the testing schedule? How will reuse ensure test result satisfaction?

2.4.1.2 Source Selection Plan

The Source Selection Plan sets forth principal considerations influencing the evaluation of competitors and includes, for example (SPO Engineers can become heavily involved with the last two items):

- A description of the source selection organizational structure.

- Proposed pre-solicitation activities, e.g., a pre-solicitation briefing to industry. A briefing will be needed to encourage reuse and to show that the government is serious about its implementation.

- Statement and description of proposed evaluation factors and their relative importance. Since reuse may be new to some contractors, describe some of the reuse evaluation factors.

This process also includes the preparation of several other documents SPO Engineers could help with, including:

- Statement of Work (SOW) - MIL-STD-245 - contractor work items to be accomplished. What reuse issues (e.g., new components (or a government approved list) shall be reusable and shall be tested to provide reusability) must be included in the SOW? If a Reuse Implementation Plan is required, will it be part of the proposal, a contract deliverable, or both?

- Work Breakdown Structure (WBS) - MIL-STD-881 - detailed separation of tasks by work activity. Will reuse efforts be considered a separate task that therefore must have its own cost and schedule reporting?

1. Acquisition Plan (AP) development by sponsoring agency.
2. Source Selection Plan (SSP) development by sponsoring agency.

Activity I - Pre-Solicitation

RFP

3. Procuring Contracting Officer (PCO) distributes sponsoring agency's Request For Proposal (RFP).
4. Contractors submit proposals to sponsoring agency.

Activity II - Solicitation and Proposal Preparation

Response to RFP

5. Source Selection Evaluation Board (SSEB) performs evaluation.
6. PCO/Source Selection Authority (SSA) determine competitive range.
7. PCO notifies bidders outside of competitive range.
8. Remaining bidders informed of deficiencies.
9. Final bidders submit best and final offer (BAFO).
10. Final Government evaluation is performed.
11. SSEB chairperson presents final evaluation results to Source Selection Advisory Council/Committee (SSAC).
12. SSAC advises SSA on conduct of source selection process and prepares comparative analysis of SSEB evaluation results.

Activity III - Analysis, Evaluation, and Negotiation

Winner selected

13. SSA makes selection.
14. PCO awards contract and notifiees unsuccessful bidders.
15. Unsuccessful bidders may request and receive debriefing.

Activity IV - Selection and Award

Figure 2-3 The Acquisition Process Involves Four Activities and 15 Steps

- Specifications - FAR Part 10 - detailed set of system requirements. How do the reuse requirements fit in with the overall system requirements?

- Evaluation criteria - FAR 15.605 - high level factors to be used in the evaluation process to select a winner. Evaluation criteria are the specific criteria (not revealed to contractors) that evaluators use to determine if (or how well) contractors satisfy evaluation factors. What are the reuse evaluation criteria?

- Proposal Preparation Instructions/Instructions for Proposal Preparation (PPIs/IFPPs) - FAR 15.406-5(b) - specifies the proposal content, e.g., format, style, and what the bidders must address. Are there any special instructions contractors need to know to satisfy the government's RFP reuse requirements (e.g., reuse implementation is to be described in the management and technical portions of the proposals)? The RFP's PPIs include the date, time, and place the proposals must be submitted. The PPIs are strictly enforced, e.g.:

  - A proposal is rejected for late delivery.

  - Exceeding the page limitations causes the extra pages to be removed from the proposal prior to start of the evaluation.

The above documents provide bidders with goals for their proposal, which should be structured to address the evaluation criteria. The evaluation criteria correspond to the SOW, WBS, and specifications.

Evaluation criteria development is based on four items (which need to involve SPO Engineers since results of this development will indicate to the bidders the importance of reuse and domain engineering):

- What to evaluate. What are the critical areas that must be evaluated to determine if the government will receive quality products from the bidders and what are the criteria that can be used to discriminate between the bidders?

- How to evaluate. During the evaluation process, what do the evaluation criteria mean and how are results (and documentation leading to these results) scored and recorded? Some SPOs use a color system (e.g., red means failed the criteria, yellow means meets most of the criteria, green means meets all the minimum criteria, and blue means exceeds at least some criteria) and others use a numbering system (e.g., 1 means exceeds criteria).

- Relative weighing. Since each evaluation criterion does not have the same level of importance, each criterion is given a weighing factor (a fraction) which is multiplied against each score. The sum of the weighing factors must equal 1.

- Minimum satisfactory levels. The evaluation criteria do not always include measurable requirements, e.g., must be at least 95 percent reliable. As a result, evaluators must understand the minimum(s) needed for a bidder to meet the criteria. Many times this requires a subjective evaluation since the bidder may not clearly state this minimum(s) acceptable requirement.

2.4.2 Activity II - Solicitation and Proposal Preparation

2.4.2.1 Request For Proposal (RFP) Development

Documents prepared for the Source Selection Plan provide the main inputs for developing RFPs. The RFP format is described in FAR 15.406-1. The functions of the RFP are to:

- Describe the requirements. (SPO Engineers could assist by helping the contractors understand what the government wants in terms of reuse (not the "how" which must be described in the proposal) and any vision on how developed components will be reused in future efforts.)

- State the contract forms.

- Establish the evaluation criteria. (SPO Engineers should participate as described above.)

- Set the proposal format.

- Provide information on the acquisition process.

2.4.2.2 Proposal Preparation

During this step, bidders prepare their proposal (which usually includes cost, management, and technical information) and formally (in writing or at meetings of all bidders) ask questions. Questions and answers are documented and sent to all bidders by the government. This is done to ensure a bidder does not have an advantage over other bidders. SPO Engineers may participate in these meetings by answering contract questions and/or advising other government team members.

2.4.3 Activity III - Analysis, Evaluation, and Negotiation

Here, SPO Engineers and others (e.g., cost analysts, management, users, and technical specialists) read the proposals, provide evaluations based on the evaluation criteria, and document proposal deficiencies and clarifications. This is done through a Source Selection Evaluation Board (SSEB) consisting of personnel representing various government functional, operational, management, and technical disciplines relevant to the acquisition.

Evaluation involves many people and can require weeks or months of effort. The evaluation process is highly structured, monitored, and controlled to ensure the government provides fair and equal treatment to the bidders. Depending upon the PPIs and evaluation criteria, SPO Engineers may be evaluators on domain engineering and other technical areas as needed.

As proposal evaluators, the main role of SPO Engineers is to understand the SOW, specifications, domain-specific requirements and evaluation criteria; to clearly understand bidders' proposals for handling domain engineering issues; and to develop documentation on deficiencies, concerns, and evaluation logic and results.

The list of deficiencies, strong points, and weak points constitute the basis for the evaluation and the foundation of the narrative summary. A score is assigned mainly on the basis of this summary.

During this activity, SPO Engineers do not become involved with cost evaluation. Instead, SPO Engineers help evaluate management and technical proposal issues.

The next step is the Best and Final Offer (BAFO). This is when the bidders still in the running are given the opportunity to revise their offer (usually based on cost, but changes in architecture can occur). SPO Engineers should not be involved with the BAFO.

2.4.4 Activity IV - Selection and Award

This activity begins when the Source Selection Authority (SSA) makes an award decision. This decision may or may not agree with the Activity III results. After this selection, the PCO awards the contract and notifies the unsuccessful bidders.

Unsuccessful bidders may request a government debriefing - which must be honored. But, the amount of information the government can provide is limited and will not include an item-by-item comparison of the proposals and must not reveal relative standings of the bidders or the scoring results. SPO Engineers may provide assistance in preparing for this meeting, but probably will not participate.

In addition, a bidder may file a protest, in which case SPO Engineers may have to help support the government's response by describing how the evaluation process and ratings satisfied the level criteria which corresponds to the RFP's listed evaluation factors and weights.

2.5 Summary

The result of the acquisition process is that there is now a legal contract between the government and one or more contractors. The acquisition process can take several years to complete, based on funding, program complexity, etc. However, the work is required to ensure a fair acquisition process, the government has identified its requirements, and the winning contractor understands the government's requirements. As a result, the acquisition process sets the stage for future cooperation between the government and the contractor(s).

The next chapter (Chapter Three) is a brief tutorial on the domain engineering procedures. Chapter Four relates Chapter Three with domain engineering duties mentioned in this chapter.

3 REUSE ENGINEERING AND DOMAIN ENGINEERING

Reuse of software has been very common among developers of software systems. Use of macros and subroutines is a simple example of software reuse. Recent reuse efforts are concentrating on expanding the role of reuse and formally institutionalizing reuse processes among software practitioners. The current emphasis is not only on reuse of software code, but also on reuse of software life-cycle components/assets/artifacts (such as requirements, architectures, test suites, etc.). To maximize reuse potential when building new systems, these components must be created, managed, and used effectively.

Domain-specific reuse refers to the process of reusing software components (both specialized and general purpose) that are applicable to a class of related systems or applications (i.e., a domain). Examples of domains include command and control systems and airborne weapons systems. Requirements and architectures (as well as other software components, including code) of existing systems within that domain are reused in constructing new systems. Such software components have to be amenable for reuse and generic enough to apply to multiple systems in the domain of interest. The focus of domain engineering is to create these reusable software components for a domain.

In traditional systems acquisition (see Figure 3-1), SPO Engineers have been largely concerned with the acquisition of a single (or a number of unrelated) system; SPO Engineers were neither required to use existing components nor required to create components to be used in future acquisitions. This does not imply that reuse did not occur in traditional systems acquisition, but that such reuse may have been informal or ad hoc. Even if an organization had a formal reuse process, it was probably internal to the organization and, as such, related acquisitions outside of the organization could not make use of the component base except through informal or ad hoc mechanisms.

The emerging fields of domain-specific reuse, coupled with the keen interest and commitment shown by DoD, will institutionalize reuse, making reuse a standard (and potentially mandatory) practice in future systems acquisitions. There are already some examples of DoD acquisitions requiring the reuse of existing components and the identification/creation of new components for use in future acquisitions. Several organizations are initiating domain engineering endeavors for several domains in preparation for impending changes in acquisitions. In expectation of these changes, *SDP-2000: A Guide To Project Implementation Of Megaprogramming* describes that:

> *"In the DoD acquisition arena, where the government is the domain manager, all parties will be driven by incentives that are contingent on use of a conventional domain model and architecture. This boundary on the objects of production and consumption will provide a stable environment for. . . government acquisition of systems of increasing capability" [POORE93].*

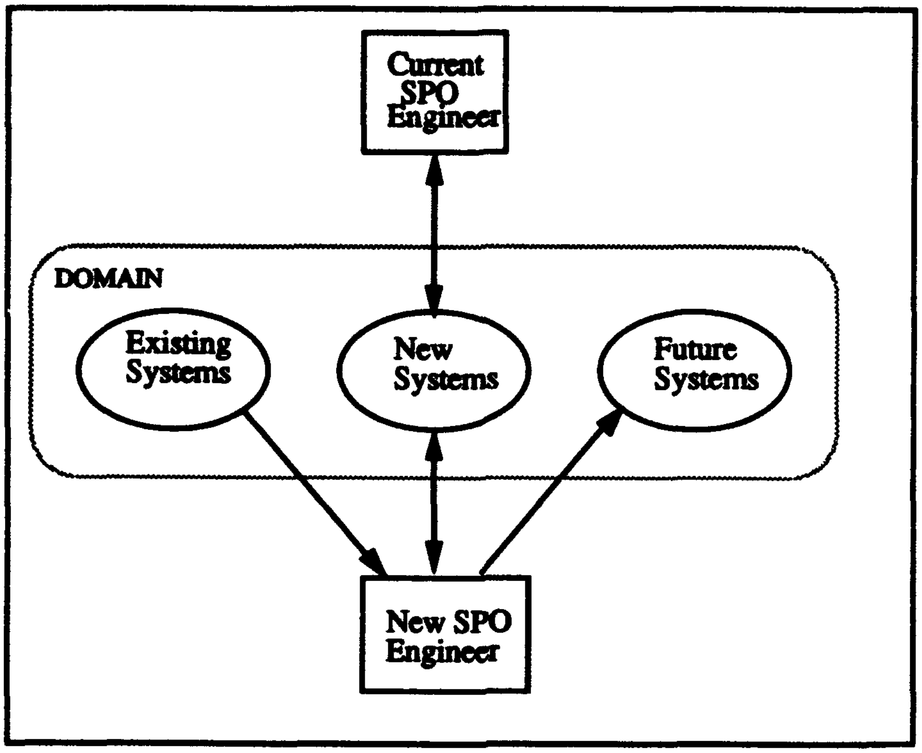

Figure 3-1 Current and New Tasks for SPO Engineers

As the government becomes the driving force for reuse through changes in the acquisition process, reuse-related technological innovations and their subsequent insertions into organizations will alter existing job definitions and responsibilities. Figure 3-1 represents the current and future tasks for SPO Engineers. Currently, typical SPO Engineers are concerned with the acquisition of a single system which may or may not consider existing components or future system needs; in the future, they will need to consider leveraging existing components and will need to keep future systems (and enhancements) in mind.

The following sections provide a general, high-level overview of reuse engineering and domain engineering processes. The intent of this chapter is to highlight perceived impacts of reuse engineering and domain engineering on SPO Engineers. Details of some of the concepts introduced are elaborated in other handbooks [CARDS93a] and [CARDS93b]. As a result, this chapter may be skimmed if the reader is familiar with reuse engineering and domain engineering.

Due to the nature of the engineering processes described in this chapter (e.g., interacting activities with feedback loops, etc.), the descriptions are presented as a continuum of related activities. This is in contrast to the style used in the previous chapter describing a sequential set of activities.

3.1 Reuse Engineering

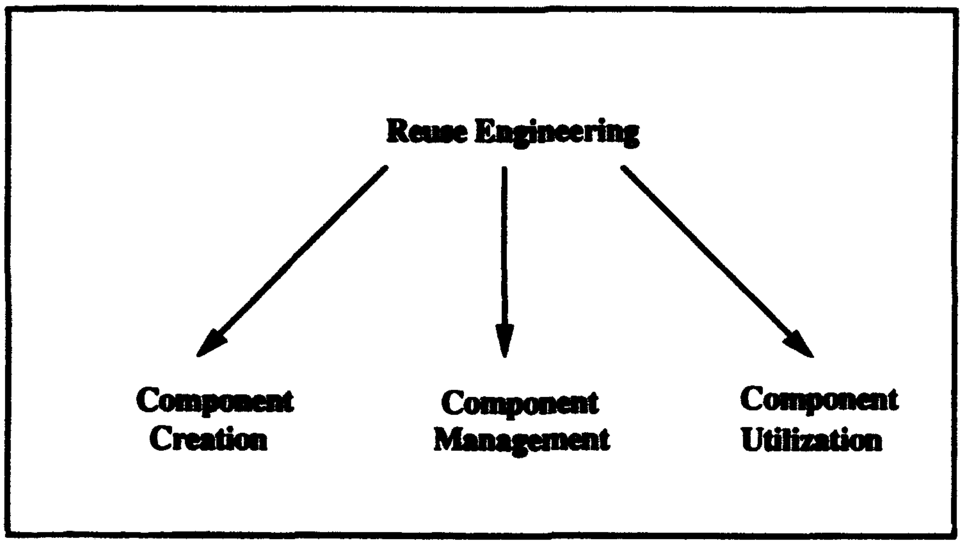

Figure 3-2 Reuse Engineering Processes

For SPO Engineers, reuse engineering can be described to encompass three processes [STARS92]: Component Creation, Component Management, and Component Utilization (see Figure 3-2).

- **Component Creation** produces and evolves domain components such as: domain models, domain architectures, application generators, software components, etc.

  Organizations chartered to create a component base for a specific domain enact the component creation process.

- **Component Management** acquires, evaluates, and organizes components produced by the component creation process. Component management acts as a brokering mechanism between the component creators and component utilizers.

  The CARDS Command Center Library (CCL) is an example of a library acting as a broker who enacts the component management process. The CARDS CCL manages components by providing the requisite services for the acquisition, evaluation, and organization of components. Normally, any process embodying data management and evolution functions of domain information can be classified as a component management process [CARDS93b].

- **Component Utilization** uses components made available by the component management process (and produced by the component creation process) to identify, select, and tailor desired components and integrate them to create application systems within the target domain.

Any organization chartered with the acquisition of a new system in the target domain is an example of an component utilization agent.

It is conceivable that a single organization may enact more than one (or all) of the above processes. However, a clear delineation is necessary to support the concept of multiple stake-holders in a domain: producer, broker, and consumer. These roles may be assigned to different organizations within and outside the DoD.

3.2 Domain Engineering

Domain engineering is primarily concerned with creation of a component base which can then be managed and used. Domain engineering activities enact the component creation process of reuse engineering. Figure 3-3 shows the context of domain engineering within the reuse engineering process. Each of the domain engineering activities is described in subsequent paragraphs.

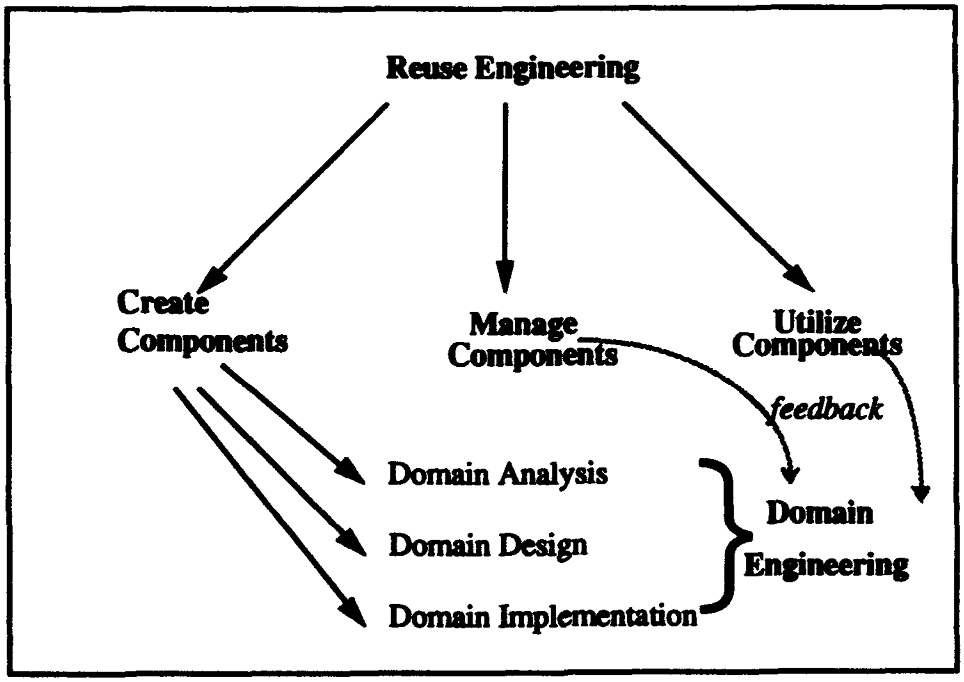

Figure 3-3 Context of Domain Engineering

The dotted arrows from the "manage components" and "utilize components" idioms of the reuse engineering process in Figure 3-3 indicate refinement and feedback loops to show that the component base is constantly evolving.

Domain engineering activities can be viewed as analogous to application engineering activities. However, it is important to note that domain engineering and application engineering are two distinct processes. The products of domain engineering activities can be (and are meant to be) utilized in an application engineering activity. Application engineering activities may provide feedback influencing future domain engineering activities.

The focus of application engineering is a single system (e.g., F-22 avionics), whereas the focus of domain engineering is on multiple related systems (e.g., avionics for all fly-by-wire aircraft) within a domain. Domain engineering activities can be summarized as:

- Analyzing and recording existing systems and future requirements in the domain (i.e., define the "problem space").

- Based on this analysis, proposing/designing a generic architecture that meets a large majority of application requirements within the domain (i.e., propose a "solution space").

- Implementing components (develop, re-engineer, or identify existing components) satisfying elements of the architecture (i.e., implement the proposed "solution space").

Thus, domain engineering can be described in terms of three interconnected activities: domain analysis, domain design, and domain implementation. These activities possess considerable interaction and feedback, and are analogous to the analysis, design, and implementation activities of application engineering (see Figure 3-4).

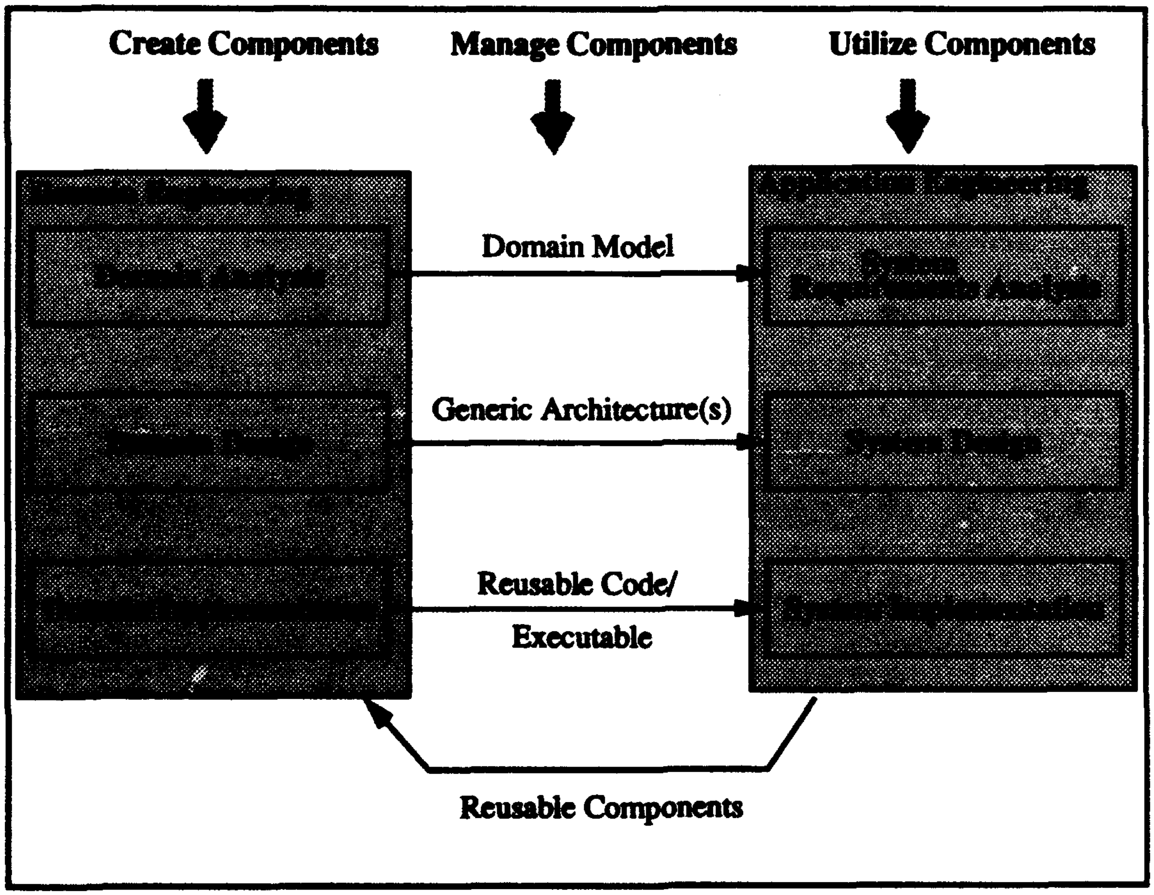

Figure 3-4 Domain and Application Engineering

The component base is constantly evolving through feedback from the component management and utilization processes.

The products of domain engineering activities (which will be elaborated upon in subsequent paragraphs) can be used at each stage of an application engineering process.

3.2.1 Domain Analysis

Domain analysis is the process of identifying, collecting, organizing, analyzing, and representing the relevant information in a domain, based on the study of existing systems and their development histories, knowledge captured from domain experts, underlying theory, and emerging technology within the domain.

> *". . . [D]omain analysis is concerned with knowledge acquisition, and with methods to make use of that knowledge. . . The ideal domain analysis approach would define methods that tell the developer everything he needs to know about reuse: which component best matches the specification, how to adapt it if it does not meet the specification exactly, etc." [WARTIK92].*

Current literature and schools of thought define the scope of domain analysis to varying degrees. Some define domain analysis to encompass all domain engineering activities (analysis, design, and implementation). Others define domain analysis to encompass defining the "problem space" and proposing a "solution space" (skipping implementation). Still others define domain analysis as an activity concerned only with the definition of the domain's "problem space." Some methods restrict the "problem space" to requirements only, while others include all the information derived from existing systems (architectures, design rationale, trade-offs, etc.).

For this handbook, the term domain analysis deals exclusively with defining a domain's "problem space" and the "problem space" is not restricted to domain requirements. Such a definition clearly recognizes the importance of all information, not just requirements, that can be derived from existing systems. This definition of domain analysis also delineates the domain's "problem space" from its "solution space." Such a distinction supports the concept of multiple stakeholders for a given domain. For example, a government agency may retain control of the domain's "problem space" and let independent contractors (or another agency) implement the "solution space."

There are several methods for conducting domain analysis [WARTIK92, SIMOS93, and DISA93]. Irrespective of the chosen method, it is important to capture not only the commonalities among existing systems but also the variability existing among them. Any rationale or trade-offs used while choosing among alternatives must also be recorded in a domain model (also referred to as "descriptive model" [SIMOS93]).

The domain model can be used during the requirements analysis of a new system. The domain model can provide a clear basis for clarifying system requirements and making informed decisions about alternatives (and associated trade-offs) existing for certain requirements. Any new system requirements that are not part of the domain model should be assessed and the domain model updated as part of domain management function.

The domain model also forms the basis for the domain design activity.

3.2.2 Domain Design

Domain design is primarily concerned with the generation of a high level generic design solution that can be applied to multiple systems within the domain. Such a design (referred to as Domain Specific Software Architecture (DSSA) [DISA93], domain architecture model [STARS92], or generic architecture [CARDS94b]) usually consists of a set of required subsystems along with relationships among them. The generic architecture satisfies the domain requirements that are elicited in the domain model. As with the domain model, the generic architecture contains common as well as variant parts (which usually, but not necessarily, map to common and variant requirements in the domain model). Traceability between the domain model and elements of the generic architecture is necessary to ensure ease of maintenance and consistency. [CARDS94e] provides more information on software architectures.

The domain design activity may produce several generic architectures at varying levels of detail/abstraction. The architectures generated from this activity reflect the user's needs/perspectives. Typical architecture views are entity-relationship diagrams (ERDs), function/object decomposition models, data/control flow models, etc.

When building a new application, a generic architecture provides the requisite framework for designing the new system. As with the domain (requirements) model, several design options can be explored to meet the new system's design constraints. If the design rationales and trade-offs are documented in the generic architecture(s), it would facilitate informed decisions (based on prior experiences and knowledge about similar systems in the domain) about design alternatives.

As part of the domain implementation, it may be feasible to provide application generators so that requisite components/subsystems can be automatically generated to provide desired functionality. The feasibility of developing application generators largely depends on the domain of interest (rigidity of specification and availability of technologies).

3.2.3 Domain Implementation

The domain implementation activity is concerned with the acquisition (to include purchase, development, and re-engineering) of reusable components supporting the generic architectures created during the domain design activity and which are consistent with the constraints inherent in these architectures. Multiple components providing the same functionality may be acquired to implement a specific element of the domain architecture, thus providing variability for the user in selecting the appropriate configuration for application needs.

Components to support the domain architecture(s) can come from a variety of sources: public domain, Commercial Off-The-Shelf (COTS), Government Off-The-Shelf (GOTS), legacy systems, etc. If there are no existing components or if existing components can not be re-engineered, new components may have to be developed. When developing new components, it is important to consider the reusability aspects for the domain: compatibility and interface requirements, performance and storage requirements, licensing issues, etc.

3.2.4 Summary

Although the domain engineering activities enumerated in paragraph 3.2 (domain analysis, domain design, and domain implementation) imply a sequence of activities, there is considerable feedback (and evolution based on this feedback) among these activities. Additionally, component management and component utilization processes of the encompassing reuse engineering process provide valuable feedback to evolve the component creation process and the component base.

In traditional systems acquisition, SPO Engineers have been largely concerned with the acquisition of a single system without much emphasis on reuse; SPO Engineers were neither required to utilize existing components nor required to create components to be used in future acquisitions. This does not imply that reuse does not occur in traditional systems acquisition, but that such reuse may have been informal and ad-hoc (i.e., no defined methods for performing reuse). The new SPO Engineers will be involved in all of these activities and may direct the reuse engineering efforts for the government. The perceived changes to the functions of a SPO Engineer will be examined in greater detail in subsequent chapters. Where possible, transitional activities will also be highlighted.

4 DOMAIN ENGINEERING DURING THE ACQUISITION PROCESS

4.1 Introduction

This chapter combines domain engineering points of Chapter Three with specific points of Chapter Two where SPO Engineers become involved with the DoD acquisition process, i.e., activities DoD assigned personnel perform prior to a contract award. Helpful information can also be found in the CARDS Acquisition Handbook [CARDS93a].

The purpose of this chapter is to reinforce the following:

> *"Software reuse and designing for reuse can yield substantial improvements in productivity and maintenance within the Air Force and the Department of Defense (DoD). Consequently, we should encourage our contractors to use existing software and to design new software for reuse" [USAF93M007].*

The following sections step through the acquisition process which Chapter Two identified as areas SPO Engineers need to support. The basic goal is to raise some questions and/or provide answers to what the SPO Engineers should do during the acquisition process. It is difficult to provide many specifics since contract requirements, needs, and visions vary with each contract.

4.2 Acquisition Plan Development

Based on the PM's guidance, SPO Engineers can provide Acquisition Plan support. [CARDS93a] provides extensive information on preparing an Acquisition Plan with reuse in mind.

4.2.1 Costing

An Acquisition Plan is concerned with funding plans and cost estimates. SPO Engineers, working for domain management, can help estimate the cost of domain engineering and reuse. Costing databases normally do not include reuse as a costing factor. For example:

- What will be the cost to add reuse capabilities to current and planned components and/or organizations? This could require a large up-front cost to the government and/or winning contractor. Will cost be beneficial? How long will it take to "break even"? Include costs for: reuse training, reuse process plans, setting up (or connecting to) a reuse library, telecommunications, facilities, computer hardware and software, etc. What near term cost savings will be offset by infrastructure investments?

- What will be the estimated reuse software component savings across programs within, or across, a domain? This could be an area of great savings. This requires knowledge about what other programs within a domain are doing. Who has this knowledge?

- What is the estimated cost/savings for reuse library utilization? This must be examined not just from a program-specific view, but also from a domain-specific view. If a domain-specific reuse library already exists, this can reduce start up and operational costs. What services/material does the reuse library provide/not provide? What is cost of reuse library interoperation?

- What is the estimated savings by having the bidders understand the program's vision statement? A long-term vision is important, as many studies have shown that maintenance is the biggest life-cycle cost factor. Therefore, this could be an important cost justification for reuse. How reliable is this vision? Is the vision funded?

- How will reuse affect life-cycle costs? If there is limited or no program/system life-cycle cost savings, then examine possible domain life-cycle cost savings.

- What type of funds (research and development (R&D), production, or operations and maintenance (O&M)) can/must be used?

- How will system development and maintenance be improved, e.g., reduce development time, risk, and cost; and reduce maintenance risk and cost?

4.2.2 Trade-off Studies/Risk Assessments

The purpose of trade-off studies/analysis is to identify the strengths, weaknesses, and risks of alternatives and to identify the preferred solution. [CARDS93a] provides guidelines to focus on trade-off criteria within the context of reuse. Working with domain management, the following must be addressed to determine if domain engineering and reuse will provide advantages to the program and domain:

- What reuse opportunities are predefined? Are there specific processes for capitalizing on these opportunities?

- Has the domain been identified by DoD to have reuse opportunities? Has the application of reuse techniques also been identified?

- Are system products/components suitable for reuse? Do criteria exist to validate these components for new applications?

- Does the acquisition process need to be modified to integrate reuse into each phase of the acquisition process and into the overall system life cycle?

- What are the business issues of reuse? Are novel strategies required in the acquisition approach to support reuse?

- What procedures are needed to collect metrics to measure the payoff from the reuse initiative and to aid developers in the selection of reusable components?
- Are there near-term products and services facilitating movement to a reuse-based paradigm?
- What training is needed to ensure practitioners capitalize on the reuse initiative?
- What technology-based investment strategy is needed to identify, track, and transition appropriate reuse-oriented processes and product technologies into this system?
- What do lessons-learned reports (from other reuse efforts) indicate can be used by this contract?
- For the proposed system, what companies already have a reuse-based business strategy? Does this strategy include systematic (planned), not opportunistic (ad hoc), reuse? Are multiple reuse approaches used?
- If this is a new reuse domain, what should the domain boundaries be? (See [DoD92], paragraph 3.1, Establish Domains, for some assistance.)
- Based on other Acquisition Plan questions and answers, is a reuse trade-off study or risk assessment needed to determine whether reuse is beneficial? If yes, the trade-off study might also address how to do reuse.
- How can other programs and domains benefit from this reuse effort? Reuse should be done for a domain rather than just for a program.
- Identify relationships between domains to facilitate communications between domain managers and enhance each domain manager's understanding of their domain.
- Determine types of components the government wants to own and under what circumstances, including cost.
- Determine what level of ownership or intellectual rights should be pursued to maximize benefits to the government and its contractors.
- Help provide guidelines to enable domain managers to do a trade-off study on requirements, e.g., does the initial cost to do reuse justify its use within a domain.
- What cost and schedule risk analysis is needed? Initiating reuse can affect cost and schedule.

- Are there possible risks (liabilities) for reuse libraries? Are there proposed solutions to decrease those risks? This needs to address, for example, legal aspects of ownership and liabilities. [CARDS93a] and [CARDS93d] address reuse legal issues.

- Have a set of potential reuse libraries and/or assets been identified? [CARDS93c] provides a partial list of libraries for reusable components.

- Is there a need to develop a reuse library(ies)? If a current or planned reuse library is not feasible for this program, then the government and the winning contractor must consider implementing a new reuse library. [CARDS93b] can help in this effort.

- What is the criteria for reuse library establishment? Is a domain-specific library required or can a more generic reuse library be used?

- What reuse-oriented architectures already exist?

4.2.3 Product Descriptions

Since domain engineering and reuse will have an impact on the final products, i.e., deliverable components:

- What are the current and future product requirements being considered by domain management? The final products should be reusable within the domain and maybe within other domains. A vision statement needs to be developed for the Acquisition Plan and incorporated later into the RFP package (see paragraph 4.4.3).

- How are products expected to change over time due to advanced technology, change in operational environment, personnel, etc.? This requires a knowledgeable vision statement to help ensure future reusability.

- Who owns, maintains, and distributes (i.e., "creates, manages, and uses") the domain architecture? When the work is finished, the government must implement a maintenance program to extend the life of the reusable component and to expand the component's use within the domain.

- Where is the domain architecture? Prior to issuing an RFP, the government should have a domain architecture(s) already in place. This ensures government control over any changes to the architecture.

- How will current reuse components affect product requirements and design, e.g., will reuse cause a change in standards or specifications? Feedback is needed to keep processes up-to-date and compatible with work being done by others.

- What tools (e.g., software reuse insertion tools and software reuse adoption handbooks) will be needed to implement software reuse? The use of tools will result in a more effective reuse program and help people implement the reuse concept.

- The bidders, as part of the proposals, need to define how a reusable component is accepted into a reuse library.

- Domain models (what a domain does), software architectures (how a domain works), product designs (how the domain is built), and implementation components (what is built) should be final products.

- Guidelines (who will develop them?) will be used to provide component design characteristics and evaluation criteria for certifying (and qualifying, if necessary) components.

4.2.4 Contracting Considerations

[CARDS93d] discusses how reuse impacts the contracting legal issues. SPO Engineers must work with domain management to determine some contractual reuse requirements, such as:

- Will reuse within, or across, domains be a factor in determining the winning bidder? Bidders having knowledge of the domain, versus knowledge of only the program, should help reduce overall cost and risks.

- Are there guidelines/regulations/contract clauses for making reusable component ownership (e.g., proprietary and intellectual) decisions? This is a critical reuse issue and must be resolved prior to issuing the RFP. The Army is currently working this issue.

- If bidder developed reuse components are used by the program, what are the legal aspects, e.g., patent, licensing, and copyrights? How much of the developed components does the government need to own? Should the government share ownership with the developer? A "clear" component title transfer to the government is needed to assure the government that the component does not infringe on the copyrights on any other component (i.e., who owns the component and what are the ownership criteria?).

- Prior to government acceptance of a component, the supplier must state if there is any legal liability for operational deficiencies.

- What needs to be developed for negotiating license and maintenance agreements? For issues that can not be resolved in advance, what negotiating process will the government use? There may be a need to develop guidelines for negotiating terms and conditions of library/subscriber agreements and library/supplier agreements.

- How will contract changes impact reuse issues? Since the government and bidders end up negotiating the final contract, changes must be examined for their possible impact on reuse.

- Does the contract clearly state that reuse contractual issues must be passed onto any subcontractors? Without this clause, subcontractor developed components may not have to comply with government reuse requirements.

- Is there a need to outline approaches to encourage reuse investment by contractors? To have industry become more active in DoD's reuse efforts, the government may have to implement an incentive program, or standard award/incentive clauses may be enough.

- Will particular reuse libraries and/or set of components be mandatory? The government can reduce risk and cost if it identifies applicable reuse libraries and components prior to issuing the RFP.

4.2.5 Make or Buy

SPO Engineers must also help determine what needs to be developed (make) or acquired (buy). For this area of the Acquisition Plan:

- How much freedom will a contractor have in determining what can be developed, bought, or reused? Contractors will be looking for the most profitable way of doing business. The government must determine the checks and balances it needs.

- How much impact (e.g., cost, schedule, and reliability) will reuse have on the make or buy decision? This could be part of a trade-off study or risk assessment.

- What is the impact of reuse on the performance evaluation of the bidders? This will be used to help determine evaluation criteria and evaluation weighting factors.

4.2.6 Test and Evaluation

Based on the use of reusable components:

- Domain engineering and reuse should improve test and evaluation planning by reducing the planning complexity and time required to implement.

- Reuse will reduce unit testing since the reusable component(s) should require less testing.

- Integration testing should be faster and more reliable since the interfacing to reusable components should be well defined and tested.
- Reuse certification and qualification procedures should also ensure reliability, etc., of components by reducing risk and liability.
- In general, the more an component is reused (without any modifications) by others, the less testing is needed. In a reuse library environment, the library maintainers are very concerned about integrity and ease of updating components.

4.3 Source Selection Plan Development

The Source Selection Plan sets forth principal considerations influencing the evaluation of bidders. As a result, SPO Engineers must provide inputs to assist in the evaluation process. [CARDS93a] provides detailed information about Source Selection Plan development.

4.3.1 Pre-solicitation Activities

Working with the legal staff, SPO Engineers must ensure reuse legal issues are listed, addressed, and resolved prior to issuing the RFP. This includes exhaustive, complete, and clear statements on ownership rights of reused components, including modified, reused components. Also, SPO Engineers must help government management understand and support reuse. SPO Engineers may prepare for and participate in pre-solicitation briefings to industry.

4.3.2 Work Breakdown Structure (WBS) Development

Here, the key issue centers around depicting domain engineering and reuse on WBS charts. For instance, will reuse libraries have their own WBS limb/leaf? The amount of domain engineering and reuse costing information the government wants has an impact on the WBS format, size, and contains. But, SPO Engineers must remember that there is a trade off between overhead cost and the amount of information required by the WBS. The impact of reuse software on the WBS must be addressed in accordance with MIL-STD-881.

4.3.3 Proposal Preparation Instruction (PPI) Development

SPO Engineers will provide limited, if any, PPI information. But, they must be prepared to assist in this area to ensure bidders understand the importance of reuse in the proposals and final products.

4.4 Request for Proposal (RFP) Development

While preparing the RFP, SPO Engineers need to ensure the RFP includes a specification, Statement of Work (SOW), and an architecture vision addressing domain engineering and reuse. This is done to ensure reuse is required as an integrated part of a bidder's developmental approach.

The specification and SOW are technical documents conveying sufficient understanding of reuse requirements necessary for bidders to prepare estimates and results pursuant to the contract. The specification and SOW must be adequate for the solicitation and award process.

The architecture vision [SAUNDERS93] is used to identify to the bidders items that will/should/may be needed in the future. For instance, since government reuse standards do not exist, the bidders may be asked to recommend product and/or process evaluation standards.

The following paragraphs provide SPO Engineers with enough information to ensure bidders understand that domain engineering and software architecture are mandatory parts of the government's acquisition process. [CARDS93a] should also be referenced.

4.4.1 Specification Development

For SPO Engineers, the specification can be used to indicate components and domain libraries known to the government that could/must/may be used by bidders. This will assist bidders in identifying specific current requirements. However, the specification should not imply a particular architecture/solution, but instead should indicate that bidders are encouraged to provide their own documented proposal recommendations/solutions for an architecture.

Some specifications are deliberately developed to give details on what is to be done in terms of physical characteristics (a design), e.g., size and shape. In this case, a compromise must be reached on how much reuse design detail the specification will contain versus containing pure requirements and no design.

The specification can also be used by SPO Engineers to indicate:

- Systems in the domain of interest that can impart knowledge about the domain and feed domain analysis or reengineering efforts to produce domain components or new application systems.

- Tools that can contribute to the reuse infrastructure within an organization and can be applied to automate reuse processes.

- Developed components submitted to target reuse library(ies) for future use within the domain or by other domains.

- Component requirements for identifying (e.g., characteristics, COTS, GOTS, and types of components suitable/desirable for reuse), selecting (criteria to validate selected components), and tailoring (re-engineer legacy systems to yield reusable components).

- Requirements for component integration testing.

- Specific usage of generic components in system acquisition and incorporation within future system acquisitions.

Since government requirements for reusable components are relatively new, SPO Engineers and the Program Manager may want to consider requiring the bidders to bid separately on needed and desirable (or goals) parts of the specification. The winning bidder then proposes which of the desired capabilities can be supplied with any remaining money.

The specification should require a demonstration/prototype of the extent of the proposed system's reuse maturity during the source selection process. An acceptable bidder's management plan (e.g., Reuse Implementation Plan), explaining how the system would be evaluated, should be required by the RFP package.

4.4.2 Statement of Work (SOW) Development

The SOW's Contract Data Requirements List (CDRL) must be used to show bidders that software reuse will be continuously reviewed (at preliminary design reviews, etc.) throughout the contract's life to ensure compliance with the winner's (i.e., the contractor's) proposed reuse methodology, tools, etc. For example, any definition or actual changes to what is a reusable component must first be approved by the government. In addition, each review will require the contractor to give a status on reuse and to demonstrate how the components will be controlled and prepared for future reuse, even for other contracts. Another reuse issue is the enforcement of standards to ensure the components will be reusable on future acquisitions.

The reviews should include audits to ensure updates to management plans (e.g., Software Development Plan (SDP), Software Reuse Plan, Program Management Plan (PMP), or System Engineering Management Plan (SEMP)) correctly describe the reuse process (including controls) the contractor is using. To assist in proposal evaluations, management plans should be required as part of the proposal and include the contractor-provided components [STARS92]:

- Business strategies, policies and procedures, expertise, technologies, cultural legacies, etc.

- Reuse planning process goals, strategies, and objectives for reuse within and across selected domains; planned infrastructure capabilities to facilitate reuse performance and evolution; and selected, tailored, and configured processes to be applied to satisfy the goals and strategies.

- Reuse enactment processes to manage active reuse programs (e.g., allocate resources to these programs, initiate and retire components, and monitor and regulate component day-to-day performance) and ensure a reuse infrastructure is established and maintained sufficient to meet program needs.

- Reuse learning processes to evaluate reuse program performance relative to local and global objectives, and explore and recommend approaches to effect evolutionary or revolutionary enhancements in reuse plans. Reuse plan enhancements can be viewed as an institutional mechanism for managing improvement and innovation.

- Creation processes to produce and evolve domain components, including domain models and architectures, application generators, and software components.

- Management processes to acquire, evaluate, and organize components produced by the component creation processes, and make those components available via some form of library acting as a brokering mechanism between component creators and component utilizers.

- Utilization processes to reuse the components made available by the component management processes by identifying, selecting, and tailoring desired components and integrating them appropriately to construct application systems within the target domain(s).

- Examination of tool reusability.

The SDP should also be part of the final contract agreement and placed under immediate change control. As with the SDP, an initial list of contractor components should also become part of the proposal. The components themselves should be part of the final contract agreement.

Periodic Technical Interchange Meetings (TIMs) should be included in the CDRL to help ensure the government's SPO Engineers and the contractor's engineers have the same expectations about the resulting product's reusability.

The CDRL should also state that the Functional Configuration Audit/Physical Configuration Audit (FCA/PCA) acceptance depends upon proof of reused/reusable components.

The government must identify contract Data Item Descriptions (DIDs) to permit reuse tracking.

Since reuse development and implementation is relatively new, another SOW option is to allow the bidders to provide revisions to the SOW. This could cause problems during the proposal evaluation process, but it would provide more meaning to the final SOW and will become more adaptable to the bidder's reuse process and proposed solution. The Program Manager and SPO Engineers must:

> *". . . make a conscious decision as to the type of component owned and managed by the government. Each domain's reuse business strategy [must] identify the level at which government ownership is prudent: requirements (what it does), architecture (how it works), or implementation (what is built). This decision will vary by domain and may vary within a domain. It will be driven by the technical factors, the potential*

market for a particular component, the health of the commercial marketplace, and the acquisition strategy with the domain" [DoD92].

4.4.3 Vision Statement

To ensure bidders understand the government's future vision and the need for domain engineering during maintenance, the Program Manager and SPO Engineers must provide bidders with a sense of direction for future capability growth. This allows the government (domain managers) to oversee the domain and ensure the system implementation preserves the architecture's desirable aspects [SAUNDERS93]. As a result, bidders must use this vision to emphasize how their components will accommodate changes in technology and the envisioned mission. [SAUNDERS93] provides examples of what can/should be included in a vision document.

Another area that should be in the vision document is the government's vision of reuse library controls, usage, etc. This will show the long range planning being done by the government, e.g., new technologies and standards.

The emphasis of the vision document is to:

- Identify existing technology base.
- Identify current and projected architecture and reuse technology needs.
- Identify possible sources of architectures and reuse technology.
- Capitalize upon available technologies, environments, etc., facilitating reuse and architectures.
- Devise a strategy to capitalize upon current technologies and plan a future approach to utilize promising new developments (short-term goals).
- Identify a strategy for assessing architectures and reuse technology.
- Define criteria to identify software technology having potential use within the program and/or domain(s).

It must be noted that to provide bidders with this vision information, the SPO Engineers must begin work during the Acquisition Plan development (see paragraph 4.2.3).

4.4.4 Statement and Description of Proposed Evaluation Factors and Their Relative Importance

The previously mentioned trade-off studies and risk assessments can be used during the RFP preparation to help complete the proposal evaluation factors and criteria. This includes establishing weights to determine relative importance. SPO Engineers must work with the Program Manager and others to determine the evaluation factors and weights. This information is then fed into the next step, evaluation criteria. To ensure bidders understand the importance of reuse and domain engineering, reuse and domain engineering must be evaluation factors and have significant weights. The PM may decide to split domain engineering into two or more factors, e.g., in the areas of technical, management, and cost.

[DOD92] states that "[L]ong term strategy must lead to the creation of a true 'black box' components' industry. . . This concept of 'black box' components implies a library of interchangeable parts that can be tied together to create new systems." An on-site visit can show government personnel whether companies are implementing this long term strategy and if they are succeeding. This strategy needs to involve "experimenting, capturing experience, setting the policy and procedures, and establishing organizational structure and mechanisms supporting a reuse-based software engineering process" [DOD92]. Four fundamental principles intristic to this concept are: domain-specific reuse, process-driven reuse, architecture-centric investment, and interconnected reuse libraries [DOD92].

4.5 Evaluation Criteria

Another acquisition process area to emphasize is evaluation criteria. SPO Engineers must identify specific data required in proposals to evaluate technical and management reuse approaches, e.g., do the bidders understand that reuse is a process, not an end product; that libraries facilitate, but do not enable, reuse; and do bidders show evidence of practicing reuse.

Appendix B contains some suggested domain engineering evaluation criteria, e.g., a primary focus should be on domain analysis, models, and architectures. [CARDS93a] also provides helpful information.

4.6 Pre-award Survey

To help ensure proposals match what actually happens within each contractor's organization, the government can perform pre-award surveys. Pre-award surveys can become an important evaluation criteria and should consist of on-site visits (not just a questionnaire) to see stated processes in action and proof-of-past process implementations and improvements.

An evaluation factor may be the company's technology and methodology to represent domain knowledge and reusable software development tools. What are the risks for each of these, e.g., what already exists and what must be developed?

Effective 4 June 1993, the AF requires regular use of "software development capability evaluations for software intensive systems in conjunction with source selections" [USAF93M003]. This AF acquisition policy authorizes two methods for use in AF source selection evaluations. These methods present some reuse acquisition issues and must be used for ". . . all software-intensive systems, and major modifications to existing software-intensive systems, if any of the [stated conditions] is met. . ." [USAF93M003].

Appendix C provides a list of possible reuse questions that may be used as part of a pre-award survey.

4.7 Proposal Preparation

Since proposal preparation is done by bidders, there isn't much SPO Engineers can do, except to prepare for the proposal evaluation.

Based on the SPO Engineer's inputs to the RFP, the bidder should supply a System/Segment Design Document - SSDD (including architecture description), SDP (including methods and tools to support and preserve the architecture), and a list of reusable components including descriptions. For SPO Engineers, a major concern is that each proposal has an architecture with appropriately selected components so the system can be expected to have a long and stable life [SAUNDERS93].

4.8 Proposal Evaluation

During this step, the SPO Engineer must help assess architecture attributes and characteristics, and preservation schemes. This work can be easy or very difficult depending on the RFP package, e.g., Proposal Preparation Instructions, specification, SOW, and vision. The proposed SDP will provide insight into the bidders' intention to manage architectures, components, and domain libraries. If trade-off studies are required, bidders need to describe their trade-off study process, e.g., what criteria initiates a trade-off study.

The SPO Engineer should consider the following during the proposal evaluation process:

1. Does the schedule address timely implementation of reuse? If reuse is to be implemented, then reuse must be part of the schedule.

2. Are adequate resources available to execute the reuse effort, e.g., training, skills, and experience? Based on the skill and knowledge levels of contractor personnel (as shown in the proposal's management plan), these should match the proposed schedule.

3. What is the allocation of resources to the reuse effort? Is it adequate?

4. Assignment of responsibilities, e.g., will the contractor's reuse advocates have an adequate channel to voice their plans (to contractor and government managers) to implement reuse versus development of new components?

5. Is there a quantification of risks associated with the reuse tasks or the process itself?

6. What quality control measures will be employed throughout the process?

7. Is there adequate provision for the environment and infrastructure?

8. How will a bidder's capability to do reuse be evaluated against reuse evaluation criteria?

9. Do the bidders have enough qualified staff, tools, facility, and equipment to maintain needed reuse components and/or libraries?

4.9 Summary

This chapter, and appendices B and C, provided the SPO Engineer with specific reuse questions and information to help prepare an RFP package and perform proposal evaluations. The end result of this work will be a contract between the government and one or more contractors. The next chapters address what the SPO Engineer must do after contract award to ensure compliance with the contract and the implementation of a reuse program.

5 ENGINEERING DURING THE DEVELOPMENT ACTIVITY

5.1 Introduction

SPO Engineers are the interface between the government domain activities and the work performed by a contractor in creating a specific instance of a domain component. Chapter Three described the activities performed by the domain managers and domain engineers in establishing and maintaining a domain-specific reuse capability. Chapter Four described the SPO Engineer's activities in assisting with the RFP and in evaluating the technical compliance of the submitted proposals. This chapter describes SPO Engineer duties as monitors of the contractor's performance during the span of the development contract. SPO Engineer activities under a sustaining engineering or maintenance contract are described in Chapter Six.

This chapter deals with the situation where a contractor is providing implementation services and the SPO Engineers are monitoring the contractor's performance. In addition to this situation, there are many development efforts completely performed within government facilities by the SPO Engineers. For the purpose of this document, the latter situation is considered as part of a sustaining engineering effort and is addressed in Chapter Six.

Contractors may have their own domain management capability. They may concentrate their business efforts on specific lines-of-business (domains) and may themselves create systems out of existing domain components. The specific procurement that has occurred (Chapter Four) will describe the manner in which contractor domain components are included into government domain libraries, if desired or required. This can only occur where the government domain growth matches, or can benefit from, the inclusion or incorporation of the contractor's domain-specific activities, and where other non-technical considerations, such as ownership and licensing, have been addressed.

SPO Engineers oversee the production efforts of the contractor in complying with the contract issued as a result of a procurement effort as described in Chapter Four. That procurement effort has incorporated the reuse-specific and domain-specific contract clauses to allow the oversight efforts by the SPO.

New component production within the framework of established domains and their developed domain management and engineering support will likely require the use of automated tools to support the production, management, and quality control of the software and other components [CARDS93c]. In-progress reviews are likely to consist of electronic on-line review using high-level Computer Aided Software Engineering (CASE) and modeling tools. Software Quality Assurance (SQA) oversight focuses on verifying that the contractor is following defined processes, as well as validating the product meets its requirements.

There are currently few government organizations having the domain management capabilities described in Chapter Three. There is a significant focus within the DoD on the transition to such a domain-specific reuse based product line structure. The focus of this chapter is on the changes that will occur in the SPO Engineer's duties because of reuse rather than changes that have occurred, since the environment of the typical SPO Engineer is in transition. These changed duties will be described both from the perspective of a hypothetical, existing, implemented domain engineering activity with which SPO Engineers interface, and from the perspective of an organization in transition to domain engineering with the focus on the incremental changes that will be occurring.

5.2 Role of the SPO Engineers

SPO Engineers are the technical representatives for the government during the development activity. They provide the technical direction and oversight to the contractor. The nature of that oversight changes as domain engineering, domain analysis, and domain-specific reuse become more widespread throughout the DoD.

Current SPO engineering development efforts focus on interactions with contractors at specific, defined intervals during the performance of the contract, supplemented by occasional technical interchange discussions to address issues. The purpose of the reviews and interchange meetings is to monitor the technical progress of the program and to address technical issues as they are recognized.

Under a development organization working within a program management office which emphasizes domain-specific reuse, the contractor interactions will change. SPO Engineers will work closely with the contractor participating in the technical development of the product. Technical reviews will occur throughout the development in addition to the contractually mandated activities. Program oversight will be provided by process and product metrics. Technical capabilities will be assessed through exercise of development prototypes. Development tools will allow SPO Engineers ongoing access into the development environment [CARDS93c]. Major emphasis will be placed on assurance that products produced under the development contract successfully integrate with the existing and future domain components.

Domain engineering and reuse need to be involved in all activities of application engineering, as opposed to waiting until the coding activity, and should be carried into the maintenance activity. The following helps explain this logic [BRACKEN92]:

- If requirements and design are done from scratch for a system and then engineers try to reuse code components from an existing system, many of the code components may not fit with the new requirements' specification and design. This may render a significant number of code components unusable or they may require a much greater tailoring effort.

- In going from the requirements analysis to design to coding, the volume of components increases by orders of magnitude. A sudden unfamiliarity with this sheer volume at a time when there are pressures to complete the system may deter people from reuse. If reuse is started earlier, the unfamiliarity can be dealt with a lot easier and in smaller chunks.

- There needs to be greater integration between domain analysis and development activities. Domain analysis occurs in parallel with developing systems (not after they are built) and under one organization.

The emerging emphasis on domain engineering and reuse encourages new life cycle definitions and new software development guidelines. New processes for acquiring software architectures and more closely addressing the oversight issues involved with domain management will be developed [SAUNDERS93].

The concept of a life cycle for use in software development is commonly associated with the development of the Waterfall Life Cycle Model in 1970 [ROYCE70]. This highly structured approach to software development stresses the early complete identification of system requirements. The Spiral Software Development Life Cycle Model [BOEHM86] is an iterative, prototype-based model stressing integrated risk assessment and mitigation. The Spiral Model also defines the requirements through the creation of successively more complete prototypes. The activities described below apply to any life cycle model, need not follow in strict sequential order, and can be repeated in an iterative process.

The government can benefit from adoption of an architectural model-based, domain-specific reuse library:

> *"From the government's perspective, a reuse library centered around a formal model of an application architecture (and the requirements such an architecture supports) can provide a significant resource for acquisition planning, request-for-proposal preparation, and contractor assessment (pre- and post-contract award):*
>
> - *A formal architecture model can capture the critical known requirements for a system, as well as future need;*
> - *An architecture-centric reuse library can support post-deployment maintenance by mitigating the risk of architecture "drift" and "erosion" as changes introduced by new requirements stress existing systems;*
> - *A model-based approach to reuse can provide a tool in support of DoD research and development in exploring, specifying and transmitting to procurement agencies the architectural requirements for a next-generation software system" [WALLNAU93].*

The sections below describe the differences in daily duties SPO Engineers can expect to work on within a program management office which emphasizes domain-specific reuse. These changes are described within the framework of the development cycle mandated by existing government regulations (e.g., DOD-STD-2167A, MIL-STD-499, and MIL-STD-1521B). DoD procurement and management standards are rapidly changing, with MIL-STD-498 (MIL-STD-SDD), incorporating an increased emphasis on reuse, currently under review as a replacement for DOD-STD-2167A. Although the activities defined by DOD-STD-2167A are used as a framework for describing changes in SPO activities, its use is not mandatory for this handbook.

5.3 Setting the Framework

Government procurement activities cover a wide range of systems. The specific duties of SPO Engineers vary depending on the type of procurement. For example, an RFP could be issued to develop domain-specific components to make an existing reuse library more complete and more suitable for use in developing systems. An RFP could also be issued to request the development of a specific system, using existing domain knowledge and components from an existing domain-specific library. The system requirements analysis, the nature of SPO Engineer oversight, and the specific new reuse-based activities that SPO Engineers will perform will vary.

The set of example procurements listed below illustrate some varying duties of SPO Engineers. The examples cover a range of typical current and expected future procurements and provide a framework for discussing the changes needed within a reuse-focused environment. These examples will be used throughout the remainder of this chapter to categorize the reuse-based activities that will occur.

5.3.1 Reusable Component Development

The term "Reusable Component Development" is used to describe a procurement issued to create components to remedy deficiencies found by the domain engineering staff during their domain analysis activities and library development activities for a specific domain. These domain engineers have prepared a detailed description of the requirements for the specific components they would like developed. The detailed requirements include interface specification to existing components within the domain, other existing domains and libraries that may be of use to the component developer, and a set of qualification criteria guidelines and rules to ensure the developed components can be integrated into the domain library.

5.3.2 System Development With Reuse

The term "System Development with Reuse" is used to describe a procurement issued for the development of a system for deployment. This system (for example, a command and control system) is to be built based upon domain knowledge in an existing domain-specific reuse library. The end users of the final system need new development to provide additional capabilities not found in any existing system. The end users have prepared a statement of need and have authorization to acquire a new system. As described in Chapter Four, the SPO Engineers have worked with both the end users and the domain analysts to develop a composite system requirement specification. The domain analysts, who control the domain-specific library for the domain of which this system is to become a part, have added their specific domain requirements to the system requirements. These domain analysts want to ensure the components of the new system will integrate with the existing domain components so the newly developed system becomes part of the domain. They also realized the system will be cheaper to develop and easier to maintain if it is constructed using existing components. The SOW requires the contractor to provide, as part of its proposal, a description of the tools and methods to be employed to support reuse, and has a requirement for the contractor to provide a description of the proposed interfaces with existing domain-specific libraries.

5.3.3 System Development for Reuse

The term "System Development for Reuse" is used to describe the development of a totally new system for deployment. The user, program managers, and the domain analysts working under the SPO have all agreed that the user's needs can best be satisfied by development of a totally new system. This new system shares little or no commonality with any previously developed and deployed systems. A domain manager has been appointed, under the SPO, to oversee any domain analysis effort. Domain analysis efforts might enhance the existing domain to incorporate the new system when completed, or to create a new domain incorporating the new system, or to decide that the new system is sufficiently specialized that its commonality with other systems is unlikely, and therefore no domain is required.

The end users have prepared a statement of need and have authorization to acquire a new system. The domain analysts have performed a preliminary domain analysis effort on the end user's needs. The SPO Engineers, working with the domain analysts, have included that preliminary analysis in the RFP. The contractor will be required to provide a detailed explanation of the tools and methods employed to support reuse, a description of the contractor's domain analysis capability, and a description of the process the contractor will employ to define the new domain's standards.

5.4 System Requirements Analysis/Design

The initial system requirements are a part of the RFP to which the contractor prepares the proposal and responds. These system requirements specify the nature of reuse to be involved with the system under development. They specify the reuse libraries to be accessed and will impose specific restrictions upon the system components developed to ensure they will become future reuse library components. The initial system requirements analysis and system design is performed before the RFP is issued and, as a minimum, a preliminary system specification accompanies the RFP. Chapter Four describes these activities.

After the contract has been awarded, a final system specification is produced by the contractor. This specification includes any items found missing during the proposal evaluation stage and includes the analysis results from the contractor. The contractor often finds conflicting or contradictory requirements requiring elaboration.

The SPO Engineer's major duties during this activity consist of working closely with the contractor to clarify misinterpretations of requirements and to monitor the contractor's establishment of their software development environment and program standards. In addition to a final system specification, the contractor usually delivers a revised SDP describing the tools, procedures, and reviews to be employed during the program.

Technology assisting this process for all procurement types includes the standards and component qualification manuals from the domain. The same tool suite used by the domain managers to perform domain analysis and modeling should be available to both the contractor and the SPO Engineers. Tools for requirements analysis, risk analysis, and program management allow the SPO Engineer to independently model the assumptions and solutions proposed by the contractor.

5.4.1 Activities under Reusable Component Development

The contractor's understanding of domain analysis in general, the existing domain, and its domain-specific reuse libraries are crucial to the success of a Reusable Component Development type of procurement. Therefore SPO Engineers must carefully review the contractor's plans for software reuse, usually provided as part of the SDP, to verify that understanding. Particular attention should be focused upon the contractor's training plans for the developers to provide a common understanding of the existing domain knowledge, the domain-specific library and the relationship of the newly developed software to existing components and domain standards. The SDP should describe far more frequent Technical Interchange Meetings (TIMs) with the SPO Engineers than the other example procurements due to the closely integrated nature of the final development product.

5.4.2 Activities under System Development with Reuse

The emphasis during this activity for System Development with Reuse is the establishment of a common, documented set of system level requirements. It is important to ensure the RFP's domain-specific requirements have been adequately captured in the systems specification to allow for their later test and verification. SPO Engineers should ensure the contractor's SDP describes the manner in which the contractor plans to access existing domain components and how they plan to ensure newly developed components can be incorporated in the existing domain knowledge base and into existing domain-specific libraries. The contractor's planned risk mitigation activities should be examined to ensure risks resulting from reusable components have been identified.

5.4.3 Activities under System Development for Reuse

SPO Engineers must ensure that System Development for Reuse establishes a documented set of system level requirements. They must also provide the communications interface between the government and the contractor domain analysts to ensure close communication during the initial development of the domain model for this new domain. If the contractor is responsible for the creation of the new domain, the SDP should describe, in detail, how the new domain, if any, will be established, how it will be controlled during the development cycle, and plans for transfer of ownership of the contractor domain components to the government. If the government domain analysts are establishing the domain, they must have complete access to the systems and software analysis results performed by the contractor.

5.5 Software Requirements Analysis

The software requirements analysis activity ensures applicable system requirements are traceable down to the developed software requirements to verify the requirements are included. It also establishes the foundation for later development of software test plans and procedures. Under all of the sample procurements, the SPO Engineers check that the software reuse specific requirements have been captured in the software requirement specification and are part of the contractor's on-line process database.

In some development efforts, the development may proceed using a Spiral Model for software development rather than the Waterfall Model. The requirements must still be defined for each iteration of the spiral, with an overall development plan defining the specific spiral wherein specified requirements will be satisfied.

For all life cycle models and development approaches, access to the same development tool suite as used by the contractor is critical. SPO Engineers need to understand the requirement traceability and accountability at a very detailed level to verify reuse specific requirements have been included. This detailed understanding is best achieved through access to the developer's on-line database.

5.5.1 Activities under Reusable Component Development

Reusable Component Development is unique in that no system is produced—only components intended to be integrated into an existing domain-specific reuse library. The requirements analysis must ensure certification requirements, coding and design standards, and process requirements are explicitly made part of the software requirement specification. Process requirements, and the audit processes to ensure their adherence, should also be included in the software requirement specification.

5.5.2 Activities under System Development with Reuse

System Development with Reuse requirements must include specific reuse requirements to ensure reusable domain components are used in the development of the system and the finished system components can be included as part of the existing domain. These reuse requirements and the system-specific requirements derived from the system specification must be testable and must be clear and unambiguous.

5.5.3 Activities under System Development for Reuse

In addition to requirements derived from the system specification, System Development for Reuse needs detailed requirements defining the creation of its domain, if any, and its subsequent control. Depending on the specific contractual arrangements, these will be performed by the contractor or by the government domain analysts. In either case, SPO Engineers provide the interface ensuring the domain analysis captures the important features of the new system. Domain standards need to be established, which will then apply to the specific development effort.

5.6 Preliminary and Detailed Design

The design activity not only creates the initial software design to be used for the system, but also prepares the test plans defining the acceptance test criteria and defines the internal test criteria to be used to validate the development effort. The results of reuse and domain-engineering activities by the contractor become more visible during this activity than were previously apparent.

SPO Engineers ensure that any specific qualification requirements imposed by the domain analysts or developed during software requirements analysis are fully traceable and accountable by the design. The user's functional requirements must be satisfied by the design, as always, but the specific qualification requirements imposed by the domain analysts are just as critical.

Test plans and test procedures should verify that reuse-specific requirements are satisfied, in addition to verifying the final performance.

Once the design activity has begun, the SPO Engineers' oversight will be to ensure the design fits within the overall domain architecture. This requires SPO Engineers to be familiar with the architectural model and the existing domain, or in the case of System Development for Reuse, the developing domain. It also requires SPO Engineers to learn the toolset in use to develop and analyze the domain to determine the adherence of the contractor's preliminary design to domain-specific qualification requirements. SPO Engineers require extensive training in domain management, domain library organization, and domain analysis to effectively understand and use these tools.

SPO Engineers must become familiar, prior to the start of this activity, with the particular contractor CASE toolset used to develop the design. Reuse issues require a closer examination of the design details than that provided by the existing templates for preliminary and detailed design documents. Those details of reuse-specific adherence to standards and interface design can be more easily understood through on-line examination of the design.

5.6.1 Activities under Reusable Component Development

Since no final system is being produced by Reusable Component Development, it is very important that any interim products, such as a preliminary or detailed design, are carefully reviewed. The sole reason for the existence of Reusable Component Development is to populate an existing domain with reusable components. Special care must be taken to ensure the newly developed components integrate with the existing components.

5.6.2 Activities under System Development with Reuse

SPO Engineers will focus on ensuring that the contractor uses, as much as possible, existing domain knowledge and components from any existing domain libraries in the development of their design. As in Reusable Component Development, they must also ensure any newly developed component designs conform to the standards used within the existing domain. Since a new system is being created, the SPO Engineers must verify that the architectural refinements are consistent between the existing domain and the new system. They provide the feedback to the domain analysts to determine the need to update the domain architecture to incorporate procurement-specific variations.

5.6.3 Activities under System Development for Reuse

Since this is a totally new development, a major effort is the establishment of the newly-created domain and its population with components from the analysis and design activities. In this effort SPO Engineers serve as the bridge between the government program office domain engineers and the corresponding contractor domain engineers. SPO Engineers assist the domain analysts in determining if the system should form the basis of a new domain and help in identifying the architectural basics of the new system.

5.7 Code and Unit Test

The coding and unit test activity implements the design in the actual code and then tests the resulting coded units for performance and adherence to requirements. Reuse libraries require strict adherence to coding standards for the development of reusable software components to provide a standardized, uniform interface to engineers accessing the coded unit at a later date. SPO Engineers must ensure the program coding standards are adequate and conform to the certification process for the reuse library. SPO Engineers must also ensure the coding standards are used, where required, and all applicable software units conform to the design.

Since integration into an existing domain is important, the unit tests applied to the coded units must not only test performance, but also compliance with certification requirements derived from the domain-specific certification process. For example, a domain may require software components to have a specified complexity level based upon some complexity measurement tool. SPO Engineers must verify coded units have been tested and satisfy this complexity as well as satisfying the performance tests.

SPO Engineers duties will be similar in all three procurement examples.

5.8 Software Integration and Testing

The software integration and testing activity is the final software activity. The software test plan, procedures, and cases are concluded with a software test report.

Software integration and testing should show major benefits of reusing existing components and domain knowledge. The time and resources needed to perform this activity should be greatly reduced, provided component interfacing has been properly designed and implemented. Another reuse benefit can result from showing that some requirements were previously tested, and passed, and therefore may not need retesting at this level.

As a result, SPO Engineers should be working with the Program Managers and the contractor to determine what tests do not have to be retested due to reuse. SPO Engineer duties will be similar in the three procurement examples although the actual tests performed will differ between the different types of procurements described. The details of the acceptance testing and criteria will have been already defined in the software integration test plans, procedures, and cases produced earlier during the development.

5.9 System Integration and Testing

The system integration and test activity is usually the final activity performed under a development contract. The activity concludes with a set of acceptance tests certifying the government acceptance of the delivered system. At this stage the final system deliverables are finished. The actual components created during the development process can be subjected to the domain certification process as a part of the final system test.

SPO Engineer duties will be similar in the three procurement examples although the actual tests performed will differ between the different types of procurements described. The details of the acceptance testing and criteria will have been already defined in the system test plans, procedures, and cases produced earlier during the development.

6 SUSTAINING ENGINEERING ACTIVITY

Sustaining engineering or maintenance efforts are typically performed in a different manner than original system development efforts. The SPO has the responsibility for sustaining engineering of fielded systems within their assigned specialty.

During this type of engineering effort, the SPO and contractor engineers work together as an integrated team on specific tasks as part of the engineering effort. The contract is usually established as a "task order" contract for an indefinite quantity of engineering services. There is a upper limit on the total dollar amount of these services. As contractor services are needed, task orders are issued to allocate resources and dollar amounts to specific short term tasks. Contractor personnel may work at the SPO facility with equipment provided by the SPO and function, on a day-to-day basis, as an auxiliary team of SPO Engineers.

Modifications, enhancements, and error-correction to fielded systems are treated, under these task orders, as mini-contracts. A task order has a statement of work, a schedule, an estimated effort/cost, and a list of requirements (changes). The SPO Engineer monitors the performance of each task order and is responsible for portions of the project work outside the specifications of the task order. Ongoing sustaining engineering efforts of fielded products require thorough supervision by SPO Engineers to ensure the modifications and enhancements made to the existing products fit within the overall domain management/enhancement plan. SPO Engineers ensure the components created under this sustaining engineering activity become part of the domain knowledge base and the sustaining engineering activity is used to "grow" the domain components.

The contractor is usually tasked to perform specific, short term duties using a work order tasking method under a blanket support contract. These individual work orders are too short in duration and too small in effort for SPO Engineers to provide the same oversight and control as with the development contract. SPO Engineers typically are working on their own portions of the sustaining engineering activity side-by-side with the contractors at the government facility. Contractual oversight is provided by the government program and individual task managers. The oversight varies from task to task.

SPO Engineer duties include domain analysis, domain management, and component qualification to ensure changes made to the fielded product are appropriately reflected in the domain components. SPO Engineers will be an integral part of the sustaining engineering team and will perform all tasks associated with sustaining engineering, including analysis, design, code, and test.

As members of the sustaining engineering team, SPO Engineer duties are similar to the combined duties of both the contractor and the SPO Engineer during a development effort. The activities performed are those described in Chapter Five and the description of duties within each activity still apply.

The SPO Engineer responsible for system maintenance will interact with domain engineers as a regular part of their duties. The interactions will take two forms:

1. Provide information to the domain engineers concerning changes made to the system being maintained.

2. Solicit and accept information concerning changes made to the managed component base and other systems in the domain.

6.1 Inputs To Domain Engineering

SPO Engineers are responsible for helping to "grow" and improve the domain component base. As changes are made to the fielded system, SPO Engineers should inform the domain engineers responsible for the domain's component base of the changes being made. If problems are being fixed or components in the system are improved, the component base should be updated. These updated components will then be made available to other users (e.g., other SPO Engineers). If major changes are being made such as component replacement or architecture changes, the domain engineers should be consulted so that the interests of the entire domain are taken into account.

6.2 Inputs From Domain Engineering

Traditionally, system modification, enhancements, and error-correction activities associated with sustaining engineering originate with users and customers of the system. With the advent of reuse from a component base comes the introduction of another driver in the direction that the sustaining activities take: component base managers. It is advantageous for SPO Engineers to exploit the knowledge and experiences of other systems within the domain to improve the systems that they are sustaining. The component management process will provide the conduit for the exchange of this information.

System failures and subsequent downtime are the rule and not the exception. If bugs discovered and fixed by users of other systems can be incorporated into a fielded system proactively, then downtime can be reduced. The sustaining SPO Engineer should solicit information from the domain's component base managers regarding component revisions and updates on a regular basis. The maintenance performed on other systems can be leveraged through the reuse process to reduce problems in systems and consequently reduce maintenance costs in those systems.

The SPO Engineer should also keep informed of changes in the domain's requirements, architectures, and status of qualified components. As new systems are developed, new technologies will replace old ones and the domain component base will evolve. As this evolution takes place, the SPO Engineer should perform analyses to determine how the fielded system may benefit. For example, if the domain's generic architecture changes to incorporate a new communications technology that provides substantial performance increases, SPO Engineers should consider modifying their system to take advantage of these advances. The decision to make such changes will undoubtedly involve economic factors, like comparing the upgrade costs to the savings achieved by extending the life of the system made possible by the upgrade.

6.3 Domain-Specific Reuse In Sustaining Engineering

Software maintainers must be aware that maintenance efforts may:

- Violate the basic system architecture and can erode or drift the architecture. This can lead to the architecture being brittle or inadaptable. This results in a lack of adherence and clarity of form, which in turn makes it easier to violate the architecture that has become more obscure.
- Identify flaws in the architecture, or possibly the requirements, necessitating a re-evaluation and modification(s) to the domain model. In this way, important information derived during sustained engineering activities is captured and the architecture will remain current and, most significantly, coherent to the implementation [HISS92].

APPENDIX A - REFERENCES AND RECOMMENDED READINGS

REFERENCES

The following is a list of references cited in this Handbook.

[BOEHM86] A Spiral Model of Software Development and Enhancement, ACM SIGSOFT Software Engineering Notes, Vol. 11, No. 4, B. W. Boehm, Aug 86.

[BRACKEN92] KAPTUR, Release 1.0 User's Guide (Draft), M. Bracken, 29 Apr 92.

[CARDS92a] Direction Level Handbook, Central Archive for Reusable Defense Software (CARDS), STARS-AC-04104/001/00, 20 Nov 92. Update (STARS-VC-B012/001/00) scheduled for release 25 Mar 94.

[CARDS93a] Acquisition Handbook, Central Archive for Reusable Defense Software (CARDS), STARS-AC-04105/001/00, 30 Apr 93. Update (STARS-VC-B011/001/00) scheduled for release 25 Mar 94.

[CARDS93b] Library Development Handbook, Central Archive for Reusable Defense Software (CARDS), STARS-VC-B005/002/00, 30 Oct 93.

[CARDS93c] Component Provider's & Tool Vendor's Handbook, Central Archive for Reusable Defense Software (CARDS), STARS-AC-0411/001/00, 15 Mar 93. Update (STARS-VC-B017/001/00) scheduled for release 25 Mar 94.

[CARDS93d] Proceedings: Software Legal Issues Workshop (22 - 24 March 1993), Central Archive for Reusable Defense Software (CARDS), STARS-AC-04117/001/00, 30 Apr 93.

[CARDS93e] Command Center Library Model Document, Release 3.1, Central Archive for Reusable Defense Software (CARDS), STARS-AC-0419/001/00, 30 Jul 93.

[CARDS93f] Engineer's Handbook, Central Archive for Reusable Defense Software (CARDS), STARS-VC-B002/001/00, 30 Sep 93.

[CARDS94a] Training Plan, Central Archive for Reusable Defense Software (CARDS), STARS-VC-B003/001/00, 28 Jan 94.

[CARDS94b] Technical Concepts Document, Central Archive for Reusable Defense Software (CARDS), STARS-VC-B009/001/00, 28 Feb 94.

[CARDS94c] Library Operations Policies and Procedures for the Central Archive for Reusable Defense Software (CARDS), Volume I, STARS-VC-B004/002/00, 27 Feb 94.

[CARDS94d] Library Operations Policies and Procedures for the Central Archive for Reusable Defense Software (CARDS), Volume II, STARS-VC-B004/002/01, 27 Feb 94.

[CARDS94e] Software Architecture Seminar Report, Central Archive for Reusable Defense Software (CARDS), STARS-VC-B008/001/00, 29 Jan 94.

[DISA92] DoD Domain Definition Report, DoD Center for Software Reuse Operations, The Defense Information Systems Agency/Corporate Information Management (DISA/CIM), May 92.

[DISA93] Domain Analysis and Design Process, Version 1, Software Reuse Program, DISA/CIM, Document No. 1222-04-210/30.1, CDRL 30.1, 30 Mar 93.

[DOD92] DoD Software Reuse Vision and Strategy Document, #1222-04-210/40, 15 Jul 92.

[FAR] Federal Acquisition Regulation (FAR), Subpart 15.6, 'Source Selection for Major Acquisitions'.

[HISS92] Targeting Domain-Specific Reuse for Post-Deployment Maintenance, Proceedings of the JSGCC Software Initiative Workshop, 1–4 Dec 92.

[HOROWITZ91] The Importance of Architecture in DoD Software, The MITRE Corporation (M91-35), Bedford, MA, B. M. Horowitz, July 91.

[KANG92] A Reuse-Based Software Development Methodology, Software Engineering Institute, K.C. Kang, S. Cohen, R. Holibaugh, J. Perry, A.S. Peterson, CMU/SEI-29-SR-4, Jan 92.

[MSTD881] Military Standard 881, Work Breakdown Structures for Defense Material Items.

[POORE93] SDP-2000: A Guide To Project Implementation Of Megaprogramming, Software Engineering Technology for IBM Corporation, CDRL 04050-001, J.H. Poore, Carmen J. Trammell, Charles Snyder, Laurence Van Dolsen, Mar 93.

[ROYCE70] Managing the Development of Large Software Systems: Concepts and Techniques, Proceedings, WESCON, W. W. Royce, 1970.

[SAUNDERS93] A New Process for Acquiring Software Architecture, The MITRE Corporation, Bedford, MA, T. Saunders, B. M. Horowitz, M. L. Mleziva, 1993.

[SIMOS93] Organization Domain Modeling (ODM), Volume I - Conceptual Foundations, Process, and Workproduct Descriptions, Version 1.0 (draft), M. A. Simos, 1993.

[STARS92] STARS Reuse Concepts, Volume 1, Conceptual Framework for Reuse Processes (CFRP), Version 2, STARS-UC-05159/001/00, 13 Nov 92.

[USAF93M003] USAF/AQ Policy Memo 93M-003, Policy on Use of Software Development Capability Evaluation in Source Selections, 4 June 93.

[USAF93M007] USAF/AQ Policy Memo 93M-007, Software Reuse Incentive Policy, 4 June 93.

[VITA92] Domain Analysis Guidelines, SofTech (DoD Software Reuse Initiative, The Defense Information Systems Agency/Corporate Information Management DISA/CIM), W.G. Vitaletti, R. Chhut, May 92.

[WALLNAU93] A Model-Based Reuse Approach in Support of the DoD Software Reuse Vision and Strategy, CARDS Technical Report, K. C. Wallnau, Feb 93.

[WARTIK92] Criteria for Comparing Reuse-Oriented Domain Analysis Approaches, The International Journal of Software Engineering & Knowledge Engineering, S. Wartik, R. Prieto-Diaz, Sep 92.

RECOMMENDED READINGS

The following is a list of recommended readings which can provide the reader with additional information on issues discussed in this Handbook. This recommended reading list is provided for SPO Engineers who may not be familiar with the acquisition process, and does not include information on reuse. For information on reuse, SPO Engineers should consult sources listed above in the References section of this appendix, and sources listed in section 1.4.1 of this document.

Acquiring Major Systems Contracts - Bidding Methods and Winning Strategies, Marshall H. Kaplan, 1983.

Air Force FAR Supplement (AFFARS).

Air Force Regulation (AFR) 70-15, Formal Source Selection for Major Acquisitions, 27 Apr 1988.

AFR 800-2, Acquisition Program Management.

Defense Federal Acquisition Regulations Supplement (DFARS).

Department of Defense Directive (DODD) 4105.62, Selection of Contractual Sources for Major Defense Systems.

DODD 5000.1, Major and Non-Major Defense Acquisition Programs.

DODD 5000.2, Proposal Evaluation and Source Selection.

Department of Defense Standard (DOD-STD) 2167A, Defense System Software Development, 1990.

Federal Acquisition Regulation (FAR), Subpart 15.6, Source Selection for Major Acquisitions.

APPENDIX B - DOMAIN ENGINEERING EVALUATION CRITERIA EXAMPLES

Appendix B has been prepared to provide readers with a list of possible Request for Proposal (RFP) domain engineering evaluation criteria. Since each RFP is different, the following is not intended to be a comprehensive list. Rather, the intent is to provide SPO Engineers with a starting point so other RFP domain engineering evaluation criteria can be thought of, considered, and applied. The following criteria are organized in terms of: general components, architecture, software and hardware components, protocol, system behavior, maintenance, prototyping, and future requirements:

- Components are clearly defined.
- Component dependency relationships are defined.
- Architectural attributes (i.e., characteristics to be incorporated into an architecture to establish quality and usefulness of the system) are easily identifiable and can be monitored throughout the program's life. Examples include:
  - Efficiency.
  - Reliability.
  - Maintainability.
  - Flexibility.
  - Reusability.
  - Cohesion.
  - Coupling.
  - Abstraction.
  - Encapsulation/information hiding.
- Major architectural components are easily identifiable.
- The following software components are completely described in the architecture, as needed:
  - Mission Specific Applications.
  - Support Applications.

    - Operation System Services consisting of kernel operations commands and utilities, system management, and security.

    - User Interface Services defining how users may interact with an application.

    - Programming Services such as programming languages and integrated software engineering environment. For example, Ada provides a foundation upon which to base reuse efforts.

    - Data Management Services allowing for the management of data independent of the process that created or used it.

    - Data Interchange Services supporting the interchange of data between applications on the same or different platforms.

    - Graphic Services supporting the capability to display element definition and management.

    - Network Services supporting applications requiring data access and applications, and interoperability in heterogeneous or homogeneous networked environments.

- The following hardware components are completely described in the architecture:

    - Computer System Platforms.

    - Communication Structures.

    - Application Specific Components.

- Software components are allocated to hardware components.

- Services available to an application are clearly defined.

- There is a clear distinction between mission specific applications (software components supporting specific mission functions) and support applications (software components that can be integrated in or shared by mission specific applications, e.g., word processor). This are also known as vertical and horizontal applications, respectively.

- Reusability with other programs or domains.

- Protocols are identified.

- There is a well defined standard way/style of communicating among components, e.g., the exchange of data and the rules governing this exchange.

- Clear information on system behavior (i.e., response of the system to its environment) is provided. This includes the following which could be reuse related:

  - Scheduling control, e.g., priority, real time constraints (interrupt latency time and context switching time), and controlled classifications (centralized and decentralized).

  - Concurrent Controls.

  - Graceful Degradation.

  - Security.

  - Start-up/Recovery.

  - Rules for Exception Handling.

  - States and Modes.

  - Deadlock Detection and Resolution.

  - Reuse Standards and Conventions.

- For maintenance: Rationale for choosing/eliminating particular architectural components and structures are clearly documented. This includes cost factors, schedule impact, and availability of components. Rationale examples include:

  - Selection criteria for COTS (Commercial Off The Shelf)/GOTS (Government Off The Shelf) software over developed software, i.e., buy vs. make.

  - Selection criteria for a particular reusable asset, COTS, GOTS, and/or contractor's package.

  - Selection criteria for hardware platforms.

  - Selection criteria for allocating processes or tasks among the platforms.

  - Criteria for partitioning and/or replicating data.

  - Selection criteria of network assets (software and hardware).

- Proposal and/or prototype demonstrates the desired system flexibility and extensibility to meet government's current and future needs.

- If there is a need for future (based on the vision document) requirements, they must be satisfied. This requires analysis by the domain engineers and others, e.g., SPO Engineers, systems engineers, and software engineers.

- Personnel skill levels match those required to implement reuse.

- Reuse component selection process meets the needs of this contract.

- Architecture evaluation occurs prior to selecting an acquisition strategy.

- Reuse is integrated throughout the life cycle.

- Reuse is systematically evaluated during the examination of alternative concepts.

- At specific critical points, reuse is re-evaluated and/or incorporated to attain the following goals: accelerated system development, reduce overall life-cycle costs, improve reliability, and provide well-structured components to serve as the basis for future system maintenance [DOD92].

- Government interests are protected if any reusable component company goes out of business.

- Suitable cost and business models are used for identifying proper business decisions to implement reuse.

- The architecture supports easy distribution across hardware elements.

- Components include documentation to enable modifications, e.g., Program Design Language (PDL).

- Endorsed coding guidelines are followed.

- There is documented evidence of successful, frequent reuse.

- Components are warranted by the company.

- A satisfactory process exists for matching a **new** set of system requirements to an **existing** set of system requirements.

- Reuse is performed systematically and formally.

- Bidder's process facilitates reuse.
- The needs to implement reuse on this contract have been identified.
- Reuse metrics include technical, management, evaluation, and predictive.

APPENDIX C - DOMAIN ENGINEERING PRE-AWARD SURVEY SUGGESTIONS

The following is a list of suggested pre-award survey questions SPO Engineers may want to ask bidders. (A follow-up on-site visit should also be performed to show actual implementation.) For each of these suggestions, the SPO Engineers must identify what will be needed to determine the best bidder and what will provide the best final product. Replies from bidders must not be revealed to other companies nor to non-members of the technical evaluation team.

- Has reuse been considered/implemented for any program your company (referred to as "you" throughout the rest of this pre-award survey) has worked on?

  If yes, at what point was reuse considered, e.g., up-front or only for maintenance and post-deployment software support? The earlier reuse is considered and used, the more experience a company will have gained.

- Provide your definition and scope of reuse.

- Provide your definition of a component.

- Was reuse pursued with:

  ___ components originally developed for future reuse?

  ___ components re-engineered to make them suitable for reuse?

  ___ other (explain)?

- If you have been involved in a program using reusable components, answer the following questions and provide needed descriptions:

  - What type of component was reused (check all that apply and provide your definition for the checked terms)?

    ___ domain model ___ architecture ___ designs

    ___ specifications ___ code ___ documentation

    ___ other (specify):

  - Were the components being used originally developed for future reuse (provide needed descriptions)?

  - Did you develop these components (provide needed descriptions)?

- Had the components undergone any qualification or certification process (if yes, describe)?

- Did you encounter significant problems with these components in (check all that apply and specify the nature of the difficulties and solutions):

  ___ using "as is" ___ modifying ___ re-engineering

  ___ integrating ___ other (specify):

- Is reuse incorporated in your training program? (If yes, describe the program and kinds of education and training. How many of the people who will be working this program have completed this training?)

- If you practice software reuse, what sort of technical support is used?

- What, if any, software reuse metrics are being collected (and by whom)?

- Are all of the software reuse metrics statistically validated (provide needed descriptions)?

- Provide specifics on how these reuse metric results have been used.

- Give specific examples of refinement suggestions these software reuse metrics have caused.

- How do you promote reuse?

- What reuse organizations do you actively belong to?

- What are the goals/history of your current reuse program?

- Is software reuse being pursued on an individual, team, or organizational level (explain how)?

- How did/do you go about identifying software reuse opportunities?

- Have you prototyped a system using reusable components?

- Identify/describe programs where you have implemented reuse and describe what/how reuse was implemented.

- Have you collected data on the up-front investment required to practice software reuse (provide needed descriptions)?

    - What did you find was required?
    - Are these data disseminated beyond your organization (if so, to whom)?
    - What existing reuse investments (e.g., hardware, software, and processes) already exist that you will use on this contract?
    - What is the up-front investment cost for this contract?
- What were your objectives in pursuing reuse in any program you have managed?
- Have you traded off requirements to reuse components (provide needed descriptions)?
- When requirements were being examined, was reuse a prime consideration (if yes, how)?
- How have you changed processes to evaluate performance to include reuse?
- How do you identify potential reuse components?
- What is the set of minimum criteria you use to evaluate potential reusable components?
- How do you evaluate components to determine their reuse applicability to your program?
- How do you determine the potential reusability of available components?
- Have you reused components as part of a test bed or pilot project?
- How do you evaluate COTS or GOTS for potential reuse components?
- If you obtain components from a reuse library(ies), which library(ies) was accessed? Identify/describe these library(ies) and the related interoperation process.
- Did you encounter any problems in using the reuse library(ies) (provide needed descriptions) and solutions?
- Did you provide usage reports back to the library(ies) (provide needed descriptions)?
- Which, if any, reuse library(ies) did you place components into?
    - Who manages this reuse library(ies)?

  - Which are in-house or external reuse libraries?

- How long has your reuse library been operational?
- What types of components are being stored in your reuse library?
- What classification/qualification schemes are being used in your reuse library?
- How was productivity enhanced through reuse?
- How successful was your reuse of existing components?
- Provide estimates/actuals of the time or money saved through software reuse?
- Were your reuse components developed against standards or guidelines regarding completeness, quality, and applicability (form/fit/function) (provide needed descriptions)?
- What reuse standards or guidelines (provide a copy) are used?
- Do you maintain the reuse components you develop or acquire (provide needed descriptions)?
- How are version control and documentation managed?
- What is your methodology to govern the evolution of components you reuse to ensure its reusability?
- What evidence do you have that your components are being reused? Which of these components where new, modified, and company owned?
- What positive and negative feedback have you received to help you gauge the effectiveness, efficiency, quality, validity, and reliability of your reusable components?
- What reuse technology do you use in developing reusable components?
- What procedure do you use to ensure other developers are aware of the components you create, and/or the reuse technology that facilitates development, and have access to them?
- What is the procedure for "black box" reuse, i.e., interfaces with other components are of concern, rather than the inner workings of the asset?

- What do you consider to be critical components of a software reuse infrastructure (a combination of policies, processes, technology and personnel required in an organization to incorporate reuse into the software development process) that would facilitate reuse of existing components or development of reusable components?
- What cost analysis or prediction models do you use to access risk, and/or analyze the impact, of software reuse on programs?
- What are the greatest reuse difficulties you have encountered in accomplishing your reuse missions? What did you do to overcome these difficulties?
- What criteria and procedures have you established to satisfy security, licensing, legal, and integrity requirements?
- What cataloguing scheme is being used for your reuse library? Does this conform to any established cataloguing standards?
- How is qualification and/or certification of components undertaken?
- How many levels of certification are possible and what are the criteria associated with each certification level?
- Are components qualified, certified, and validated against domain requirements?
- What library mechanism is being used for browsing, extraction, etc.?
- What types of support services does your reuse library provide?
- Does your reuse library have a hotline service? If yes, what services are provided and what are its hours of operation?
- What are your reuse library criteria for potential users to access the library(ies)?
- Does your reuse library interoperate with other reuse libraries (if yes, specify which libraries)?
- Are user extractions queried to determine whether the components were actually used in an application?
- What liabilities do you face in operating your reuse library?
- What do you consider to be critical, general, and specific characteristics of components you would be asked to reuse for this contract?

- What are the critical characteristics of the architecture pertaining to class of application for this contract?
- What are your reuse business policies, procedures, etc.?
- Describe your domain analysis process.
- Describe your component analysis process.
- Describe your policy on component ownership.
- What is your approach toward:
  - Domain-specific reuse?
  - Process-driven reuse?
  - Architecture-centric investment?
  - Interconnected reuse libraries?
- How does company policy reward personnel who implement reuse over those who do not implement reuse?
- How are the following reuse design goals satisfied:
  1. Generality?
  2. Completeness of requirements and design?
  3. Modularity?
  4. Application independence?
  5. Quality documentation?
  6. Extensibility/argumentability?
  7. Reliability?
  8. Performance/efficiency?
  9. Adaptability/flexibility?

10. Portability?

11. Robustness/fault tolerance?

12. Understandability/clarity?

13. Independence from machine/compiler/operating system?

14. Reusability?

15. Extensibility?

- How are component certification criteria accomplished for requirements, architecture, design, and code?

- If multi-level component certification is used, what is the implementation method? NOTE: Highest level could be: reviewed, approved, complies with standard, contains documentation and test materials, meets requirements, and has been cleared for security purposes.

- What tools are used to support domain analysis, component certification, and other aspects of reuse?

- What tools are used to determine constraints and the relationship among components?

- How does the bidder's reuse process support building secure applications?

- How does the bidder's reuse process identify requirements for ensuring security and integrity of reusable components?

- Provide a copy of an implemented Reuse Implementation Plan.

- Do test plans, text procedures, and test results exist for each reusable component? (Provide samples.)

- Is reuse evident in software engineering policies and procedures?

APPENDIX D - GLOSSARY

The following is a list of terms used in this document and includes other Central Archive for Reusable Defense Software (CARDS) Program reuse terms and definitions.

accessibility	1) A measure of openness of a system as determined by policy, network and library connectivity, communications support, and special hardware/software requirements. 2) A measure of extent to which a component facilitates selective use of its parts.
accountability	A measure of an operational reuse library's capability to collect, relate and utilize audit trails, usage statistics, behavior patterns and other metrics to support enforcement and continuous improvement of its operational policies and procedures.
acquisition	1) The process in which the government acquires goods and/or services through competitive negotiations. 2) The phase of library population in which potential software products are identified, screened and then evaluated for inclusion in the library. 3) The process of acquiring an item through purchase, lease, rent, etc.
ad-hoc reuse	Reuse is practiced ad-hoc when there are no defined methods for performing reuse.
adaptability	1) A measure of the ease with which a component can be altered to fit differing user images and system constraints. 2) A measure of a library mechanism's capability to represent multiple data models, user defined data models and other user defined tailoring, and the ability to support multi-organization's policies and to incorporate new technology.
adaptation	The phase of library population in which existing software products are modified or enhancements (i.e., wrappers) are designed, implemented, and tested as new software products.
application	A system providing a set of general services for solving a user problem.
application domain	The knowledge and concepts pertaining to a particular computer application.

application engineering	Similar to software engineering, the focus of application engineering is often on a single system within a domain. Uses products from domain engineering.
architectural constraint	A formalism of the relationships between architectural subsystems and any limitations placed upon them.
architectural requirement	See architectural constraint.
architecture	Often used as a synonym for a design. However, the term usually refers to a specification documenting the way in which pieces are integrated to form a whole, as in the components of a software system.
architecture-centric reuse	Reuse is architecture-centric when the component development process and application engineering processes are based on a generic architecture. The goal of an architecture-driven process is to achieve black-box reuse.
architecture-level integration	Combining architecture-level components to create a system architecture or domain architecture.
architecture model	A model representing the interrelationships between system elements, which sets a foundation for later requirements analysis and design steps.
architecture modeling	The process of creating a software architecture(s) that implements a solution to problems in the domain.
asset	See component.
auditability	A measure of a library's capability to support the capture and analysis of usage statistics, behavior patterns, and other metrics.
binding	Language specific interface to the services defined in a standard (a wrapper for components written in a different language).
black box	Electronic equipment/software that functions and is packaged as a unit and whose internal mechanism is hidden from the user.
black-box reuse	Black-box reuse is achieved when application engineers can compose systems by plugging together different reusable components based on an application's requirements.

cataloging Placing information about a reusable component into a reusable software library.

certification See component certification.

central design activity A central design activity (CDA) is generally a large DoD activity, with an average of 50+ personnel, associated with software design, development, re-engineering, maintenance, systems integration, and common support activities. Common support functions include workload control, systems development guidance and tools, data administration, software repositories, and application development process and assessment improvement programs.

classification A mapping of a collection of objects to a taxonomy; the process of determining such a mapping.

classification scheme The organization of reusable software components according to specific criteria.

command center A facility from which a commander and his/her representatives direct operations and control forces. It is organized to gather, process, analyze, display, and disseminate planning and operational data and to perform other related tasks.

commercial off-the-shelf (COTS) Commercially available software.

common criteria Attributes used to evaluate a component regardless of the domain. See component certification.

commonality Those features prevalent in the great majority of applications in a domain.

component A set of reusable resources related by virtue of being the inputs to various stages of the software life cycle, including requirements, design, code, test cases, documentation, etc. Components are the fundamental elements in a reusable software library. Any unit of captured knowledge that can potentially be reused. See reusable component.

component acquisition The process of obtaining components appropriate for reuse to be included in a library.

component-based library Component-based libraries are similar to book libraries. They can be thought of as software warehouses. The

	central focus of a component-based library is the component.
component certification	The process of determining that a component being considered for inclusion into a library meets the requirements of the library and passes all testing procedures. Evaluation takes place against a common set of criteria (reusability, portability, etc.).
component creation	The process of producing and evolving domain components such as: domain models, domain architectures, application generators, and software components.
component engineer	Responsible for evaluating components for the library, adapting the components if necessary, evaluating component criteria, analyzing the criteria, integrating components, and reporting the findings as appropriate.
component management	The process of acquiring, evaluating, and organizing components produced by the component creation process. Acts as a brokering mechanism between the component creators and component utilizers.
component qualification	The process of determining that a potential component is appropriate to a library and meets all quality requirements. Evaluation takes places against domain criteria.
component utilization	The process of using components from the component management process to identify, select, and tailor desired components and integrate them to create application systems within the target domain.
context	The circumstances, situation, or environment in which a particular system exists.
context analysis	The process of defining the extent (or bounds) of a domain for analysis.
context diagram	A top-level data flow diagram showing external interfaces to the process described.
context model	To model the scope of the domain as it exists within a larger domain denoting inputs and outputs.
data model	A logical representation of a collection of data elements and the association among those data elements.
domain	An area of activity or knowledge containing applications which share a set of common capabilities and data.

domain analysis
The process of identifying, collecting, organizing, analyzing, and representing the relevant information in a domain based on the study of existing systems and their development histories, knowledge captured from domain experts, underlying theory, and emerging technology within the domain (defining the 'problem space').

domain analyst
An individual skilled in domain analysis methodologies. The domain analyst is responsible for defining the language, tools, and techniques used in performing the domain analysis. This person also documents the domain model and may be responsible for defining any generic architectures associated with the domain.

domain architecture
High-level paradigms and constraints characterizing the commonality and variances of the interactions and relationships between applications within a domain.

domain constraint
Represents the mission-level requirements identified within the boundaries of a domain. They determine the functionality of the system expressed in terms and language dominant within a domain.

domain criteria
Specifications which a potential component must adhere to in order to obtain acceptability in the domain and inclusion in the library. A composite of three sets of constraints: component, architectural, and implementation.

domain design
The process concerned with the generation of a high level generic design solution that can be applied to multiple systems within a domain (proposing a 'solution space').

domain engineering
An encompassing process which includes domain analysis and the subsequent construction of components, methods, tools, and supporting documentation addressing the problems of system/subsystem development through the application of the knowledge in a domain model and software architecture. The focus is on multiple, related systems within a domain.

domain expert
An individual with extensive knowledge of a particular domain.

domain implementation
The process concerned with the acquisition (to include purchase, development, and re-engineering) of reusable components supporting the generic architecture(s) created

	during the domain design and which are consistent with the constraints inherent in these architectures (implementing the proposed 'solution space').
domain language	A collection of rules relating objects and functions and which can be explicit and be encapsulated in a formal language; further, it can be used to specify the construction of domain systems.
domain-level integration	The process of using and evolving domain and application components in the creation of requirements, architectures and implementations (domain and application).
domain management	Typically the management of a group of similar systems.
domain model	A definition of the functions, objects, data, and relationships in a domain, consisting of a concise representation of the commonalities and differences of the problems of the domain and their solutions.
domain modeling	The process of encoding knowledge about a domain into a formalism.
domain requirement	See domain constraint.
domain-specific library	A library whose components are bound by a specific domain.
domain-specific reuse	Reuse targeted for a specific domain (as opposed to reuse of general purpose work products). It typically involves reuse of larger work products (subsystems, architectures, etc.) than general purpose reuse.
domain-specific software architecture	An architecture (interactions and relationships between objects) used to develop software applications based on a specific domain.
encode	See library encoding.
entity	A particular and discrete unit; a named product, process object, or relationship.
expandability	The extent to which a library or component allows for adding new components or functions.
extensibility	1) A measure of the ability to modify and enhance an operational reuse library's data model and contents while minimizing interruption to subscribers. 2) The extent to

	which a component allows new capabilities to be added and existing capabilities to be easily tailored to user needs.
extraction	See retrieval.
feature	A prominent or distinctive user-detectable aspect, quality, or characteristic of a software system or systems.
flexibility	1) A measure of the operational reuse libraries' ability to accommodate changing subscriber requirements, such as handling of proprietary software, organization's policies and new technology. 2) The extent to which a component's missions, functions, or data satisfy other requirements.
franchise	An instance of a domain-specific infrastructure built utilizing the CARDS Concept of Operations/Franchise Plan.
franchisee	Group to whom a franchise is granted.
generic architecture	A collection of high-level paradigms and constraints characterizing the commonality and variances of the interactions and relationships between various components in a system.
generic command center architecture	The fundamental generic architecture underlying command center applications.
government off-the-shelf (GOTS)	Software developed for and owned by the government.
horizontal domain	The knowledge and concepts pertaining to a particular functionality of a set of software components that can be utilized across more than one application domain.
implementation constraint	Provides the hardware and software requirements to which the individual software modules must adhere.
implementation-level integration	Combining components to implement a system.
implementation requirement	See implementation constraint.
infrastructure	A basic, underlying framework or features.
integration	The process (in library population) of verifying that a software product meets the architectural constraints imposed by the generic architecture.

interoperability	The ability of two or more systems to exchange information and to mutually use the information that has been exchanged.
knowledge blueprint	A flexible plan to transition knowledge to the community.
knowledge engineer	An individual responsible for modeling domains. Knowledge engineers work closely with domain analysts and domain experts in encoding domain analysis products into a library model.
knowledge representation	Codification of domain knowledge. Captures requirements, design, code, and test information in machine processable form.
large scale reuse	Large scale reuse is the reapplication of high-level components (e.g., requirements, architectures, designs).
library-assisted reuse	Reuse is library-assisted when there exists a library to support the application domain. There may be more than one library and they may be interconnected.
library	A collection of components cataloged according to a common classification scheme and a set of applications providing a mechanism to browse and retrieve components.
library encoding	The process of encoding the products of the domain analysis into a library model.
library model	A model representing the domain components and the relationships between them.
library population	The process of acquiring/developing components in support of the library model.
library system	A set of one or more libraries.
life cycle	All the activities (e.g., design, code, test) a component is subjected to from its inception until it is no longer useful. A life cycle may be modeled in terms of phases, which are often characterizations of activities by their purpose or function, such as design, code, or test.
megaprogramming	Megaprogramming is achieved when systems and subsystems can be viewed as black-boxes that meet certain requirements. These systems can be reused in building other systems without the developer having detailed knowledge of the system's internal structures.

memorandum of understanding	An agreement stating terms of cooperation between two entities.
metrics	Quantitative and qualitative analysis values calculated and collected according to a precise definition and used to establish comparative aspects of development progress, quality assessment, or choice of options.
model	A representation of a real-world process, device, or concept.
model-based library	Model-based libraries are organized around the principle that what matters in a repository is the context in which reusable software components are used and the relationships among components. The focus of a model-based library is the model (requirements, architectures, design decisions and rationales) and the software which implements these models.
modeling	The process of creating a model.
operability	A measure of the ease of learning versus ease of use of a library mechanism's capability to support searches, retrievals, extractions and contributions.
opportunistic reuse	Reuse is practiced opportunistically when it is up to software developers to identify when reuse is possible, locate reusable components, and integrate them.
process-driven reuse	Software reuse is process-driven when it is an integral and transparent part of both the software engineering process and the broader acquisition process.
prototyping	The practice of building a first or original model (sometimes scaled down, but accurate) of a system to verify the operational process prior to building a final system.
qualification	See component qualification.
rapid prototyping	The process of using a library mechanism to quickly prototype a system.
re-engineering	The process of examining, altering, and re-implementing an existing computer system to reconstitute it in a new form. It uses practices such as restructuring, reverse engineering, and migration to identify and separate those systems worth maintaining from those that should be

	replaced; to extend the useful life of existing systems; and to perform maintenance more efficiently and correctly.
repository	The mechanism for defining, storing, and managing all information concerning an enterprise and its software systems - logical data and process models, physical definitions and code, and organization models and business rules.
retrieval	The process of obtaining a component from a library so that it may be used in the development process.
reusable component	Captured knowledge that is designed and implemented for the specific purpose of being reused in developing new systems. Reusable components include requirements, specifications, domain models, software architectures, product designs, and implementation components (source code, test plans, procedures and results, and system/software and process documentation).
reuse	The application of existing solutions to the problems of system development. Reuse involves transfer of expertise encoded in software-related work products. The simplest form of reuse from software work products is the use of subroutine/subprogram libraries for string manipulations or mathematical calculations.
reuse library	A library specifically designed, built, and maintained to house reusable components.
reverse engineering	The process of analyzing a computer system to identify its components and their interrelationships.
scaleability	A measure of number, diversity, and size of components that can be managed by a library mechanism.
semantic network	A graphical knowledge representation method composed of nodes linked to each other.
small scale reuse	Small scale reuse is the reapplication of code: subroutines, object libraries, or Ada packages.
software architecture	High-level paradigms and constraints characterizing the structure of operations and objects, their interfaces, and control to support the implementation of applications in a domain. Includes the description of each software component's functionality, name, parameters and their types, and a description of the component's interrelationships.

software engineering environment (SEE)	Computer hardware, operating system, tools, computer-hosted capabilities, rules, and techniques assisting in the development and production of software.
subsystem	Conceptual aggregate of complimentary functions within an architecture.
system architecture	A model representing the interrelationship between system elements which sets a foundation for later requirements analysis and design steps.
system composition	The automatic configuration of a prototype system based on hardware and software requirements.
system engineering	A process encompassing requirements gathering at the system level with a small amount of top-level design and analysis.
System Program Office (SPO) engineer	An engineer working at or below a program management level in a DoD organization. Common support functions include pre-Request for Proposal activities, proposal evaluation, monitoring of engineering activities after a contract is awarded, and monitoring of ongoing sustaining engineering efforts (or maintenance) of fielded products.
systematic reuse	Reuse is practiced systematically when there exist defined procedures for leveraging future software projects. Efforts are devoted up-front to creating a suitable process.
taxonomy	The theory, principles, and process of categorizing entities in established categories.
technical reference model	A conceptual description of the functionalities encompassed within the domain.
tools	Items contributing to the reuse infrastructure within an organization and can be applied to automated reuse processes, e.g., creating/maintaining/modifying and storing/retrieving reusable components.
vertical domain	The knowledge and concepts pertaining to a particular application domain.
white box	Electronic equipment/software that functions and is packaged as a unit and whose internal mechanism is known to the user.
wrapper	A component which allows passing of data between components.