Defense Information Systems Agency
Joint Interoperability & Engineering Organization
Center for Computer Systems Engineering

A Process for Transitioning to Object-Oriented Technology

POC: David Diskin
diskind@ncr.disa.mil
DISA/JIEO/CFCSE/JEXF

The work was conducted under contract DASW01-94-C-0054, Task T-S5-1266, for the Defense Information Systems Agency. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

PREFACE

This paper was prepared by the Institute for Defense Analyses (IDA) for the Defense Information Systems Agency (DISA) under the task entitled “Object-Oriented Technology Implementation in DoD.” This document fulfills a task objective to provide a technology insertion plan for an organization considering object-oriented technology in its software development process.

The following IDA research staff members were reviewers of this paper: Dr. Edward A. Feustel, Dr. Dennis W. Fife, Dr. Michael C. Frame, Dr. Richard J. Ivanetich, and Dr. Reginald N. Meeson.

The authors would like to acknowledge Mr. David Diskin for his constructive comments and suggestions on several versions of this paper.

Table of Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

The use of object-oriented technology (OOT) to develop software is becoming increasingly popular. Organizations unfamiliar with OOT may be ready to investigate whether the technology can provide them with significant improvements over present software development capabilities. While OOT is not the solution to every software development problem, it does offer the potential for improving productivity and quality.

The purpose of this document is to provide a suggested approach for an organization that wants to transition object-oriented techniques into its software development process. It describes a process for introducing and transitioning OOT to an organization. The process uses a pilot project as the primary mechanism for introducing OOT into the organization. The pilot project, or a small number of such projects, will provide sufficient OOT exposure to determine how suitable the technology is to the organization. The use of a pilot project captures important feedback for future potential projects, and the risks inherent in evaluating any new technology are isolated. The primary audience for this document is software development managers and developers within Department of Defense Central Design Activities, although most aspects of the transition process are generally applicable to any organization interested in transitioning to OOT. The OOT transition process consists of the following four steps:

a. Perform Planning. The purpose of this step is to develop a detailed understanding of the resources, activities, and commitment that will be required to conduct an OOT pilot project. The organization (through an individual called the “OOT champion”) explores the applicability of OOT to the organization and works to obtain management commitment to proceed. With this commitment, the executors of this transition, whether software development managers or OOT champion, conduct detailed planning for the initial pilot project.

b. Conduct Training. The purpose of this step is to establish a well-understood competency in OOT by all participants in the pilot project. The OOT champion,

managers, users, and developers receive appropriate training and education in OOT concepts, tools, and languages.

c. Conduct Pilot Project. The purpose of this step is to begin using OOT on one or more pilot projects. The organization tries OOT, the “new” technology, on a small scale. These early projects should be real projects, but not critical or high-risk ones. Although this document describes the conduct of a single pilot project, the use of multiple, concurrent pilot projects is not precluded.

d. Transition OOT to the Organization. The purpose of this step is to establish an OOT capability within the organization. Given successful experiences with the early pilot projects, the organization may decide to apply OOT to other software development projects.

A set of activities is described for each of the steps above. These activities provide more detailed guidance to an organization on various topics that relate to each step. A summary of the steps and activities follows.

Summary of the OOT Transition Process

- Perform Planning (A1)
  - Perform Preliminary Planning (A11)
  - Cultivate Management Commitment (A12)
  - Perform Detailed Planning (A13)
- Conduct Training (A2)
  - Train Early Adopters (A21)
  - Educate Management (A22)
  - Train Project Team (A23)
- Conduct Pilot Project (A3)
  - Acquire and Install Project Materials (A31)
  - Execute Pilot (A32)
  - Review and Assess Pilot (A33)
- Transition OOT to the Organization (A4)
  - Remove Impediments (A41)
  - Transition to Project-Based Use (A42)
  - Transition to Domain-Based Use (A43)
  - Transition to Enterprise-Based Use (A44)

1. INTRODUCTION

1.1 PURPOSE

The use of object-oriented technology (OOT) in software development has become popular in the past five years. OOT is being used in all phases of software development, as well as in other areas such as organizational modeling and database implementation. Effectively introducing OOT into an organization requires a concerted transition effort. Organizations can often point to examples of failed technology transition: automated tools became “shelfware” because of a lack of training, a software inspection process did not yield effective results due to an extended delay between training and implementation, or Ada design approaches were flawed because an experienced mentor was not available. There are many considerations involved in ensuring a successful transition to OOT.

The purpose of this document is to provide a suggested approach for an organization that wants to transition object-oriented techniques into its software development process. This document discusses a number of issues related to OOT transition, such as planning, training, pilot projects, and acquiring organizational commitment.

1.2 BACKGROUND

The Defense Information Systems Agency (DISA) recently sponsored a study of the potential use of OOT for development activities for Department of Defense (DoD) information systems [Jordan 1993]. One major conclusion of the study was that OOT could improve the maintainability and reusability of software, thus leading to a reduction in the costs for software development and maintenance. The study also noted that OOT is successfully employed by many organizations, both government and commercial. In addition, there now also exists a significant commercial infrastructure of OOT software development tools and methods for information system development. As a result, DISA considers OOT a technology that could provide substantial benefits to DoD’s software development organizations. In particular, the Central Design Activities (CDAs) responsible for in-house information system development and maintenance are prime candidates for transitioning to the use of OOT.

1.3 SCOPE

This document provides a generalized process by which an organization can consider, present, and introduce OOT. A pilot project is recommended as the primary approach for introducing OOT to an organization. Guidance is given on preparing for and conducting the pilot project, as well as transitioning OOT throughout the organization. An organization may choose to follow the process in a step-by-step manner or simply use those sections that pertain to its specific technology transition needs. The process provided here is based upon the transition experiences of other OOT adopters, and should be tailored according to the specific needs of the organization. This document does not address detailed technical aspects of OOT, such as specific design methods, language issues, and available commercial tools.

This document is intended for software development managers and developers, primarily within DoD CDAs. The people who are inclined to experiment with and adopt new technology (e.g., early adopters, technology champions) will find this document particularly useful, as it provides ideas and suggestions for introducing OOT to an organization.

1.4 NOTATION USED

Figure 1 depicts the IDEF0 modeling notation used to describe the OOT transition process presented in this document. The arrows in Figure 1 illustrate activities with inputs, controls, outputs, and mechanisms (ICOMs). The inputs are entities that will be either transformed or consumed by the activity. The controls represent entities that are used by the activity, but not transformed by it, such as a regulation or standard. The mechanisms are resources or enablers, such as a database or a person. The outputs are the products of the activity. ICOMs that are in parentheses are considered optional.

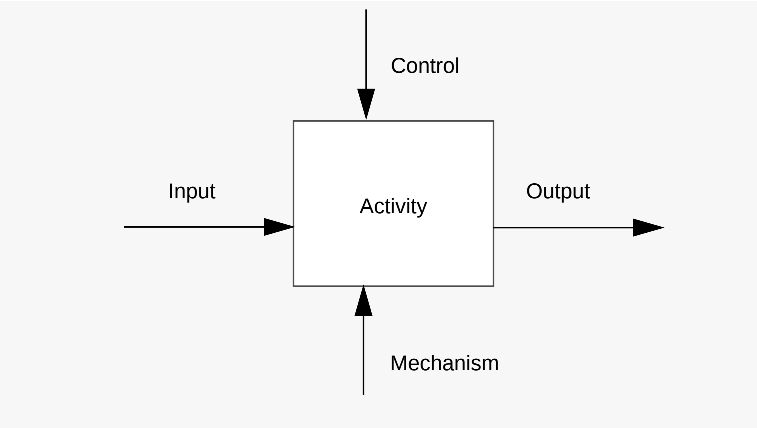


Figure 1. IDEF0 Notation

2. OVERVIEW OF THE TRANSITION TO OOT

OOT offers an improved approach for building software applications and databases [Capper 1994]. By using OOT, organizations are able to build more maintainable and reusable software and databases more capable of handling complex data. Until recently most software development approaches separated functions and data, structuring software around the functions to be performed and organizing data around information flows. Object-oriented approaches structure software specifications and code around objects, combining both data structure and behavior into a single entity. Object-oriented approaches to software analysis, design, and programming have become increasingly popular, and the object-oriented perspective has also been extended to other areas of information technology such as database design.

OOT currently covers a wide range of the information technology field. Today there are a variety of object-oriented programming languages (including Ada 95), requirements analysis methods, design techniques, and tools to support object-oriented development. OOT is increasingly embraced by industry for information system development and has a broad base of commercially available tools, training, and consulting. In addition, object standards are emerging: over 500 companies have formed a consortium called the Object Management Group to develop object specifications and standards [OMG 1994].

This section discusses the context of transitioning to OOT. This includes the effect upon the organization, technology transition approaches, assessing organizational readiness, obtaining support for OOT transition, and an overview of the OOT transition process.

2.1 ORGANIZATIONAL EFFECTS

Transitioning to OOT involves new ways of problem solving, new techniques and tools to learn, specific OOT training and mentoring, and a period of transition for the development team. There may be new roles and responsibilities for developers, such as class librarians, domain experts, and object modelers. The use of OOT often results in a change in the software development process, one that encourages a more iterative and incremental process as opposed to a traditional, waterfall process. There are some things an organization should anticipate as the result of transitioning to OOT.

First, OOT requires a “change in mindset” for problem solving. Most information systems have been developed using functional or process-based approaches. That is, the software was structured around the functions to be performed as opposed to the data to be managed. The result of these functionally oriented systems was that data was global to the entire program and a change in one place resulted in a “ripple” of changes throughout the software. An object-oriented approach structures the software around the objects in the system, reducing the likelihood of such “ripples.” However, implementing this approach requires one to “unlearn” functional or process-based techniques and replace them with the object-oriented approach to problem-solving.

Second, an organization will experience a learning curve in acquiring object concepts, languages, techniques, and tools. Exploiting object technology for its benefits of high reuse and maintainability has the best chance of happening if developers are well trained. So an organization must be prepared for both the cost and time required for OOT training. Estimates for an organization to transition range from one to five years. Successful transition will depend upon the existing expertise and capabilities of the developers.

Third, an organization will experience a change in the software development life cycle as object orientation becomes prevalent. Although the fundamental life cycle activities are likely to remain the same (i.e., requirements, design, code, test), the object-oriented life cycle is centered around objects. These activities are focused on creating, modifying, and applying objects. This is in contrast to other life cycle approaches, where products from one phase may not be well understood (or properly used) in the next phase.

2.2 OOT TRANSITION APPROACHES

A successful technology transition effort does not occur overnight. There are different stages an organization must go through before a technology is fully integrated into its operations. Following is a discussion of different approaches some organizations have used in introducing OOT. Note that in all of the successful transitions that were observed, the initial use of OOT was small in scale and exploratory. The progression from this initial use to a mainstream technology did not happen immediately for these organizations. Understanding this is valuable since it is easy to set unrealistic expectations regarding any technology transition.

The following three technology transition approaches were described in [Jordan 1993]:

a. System/Project-Based Approach. The system/project-based approach is driven by the demands of a specific project. This may be the result of imposing a design or implementation requirement (such as a specific programming language) that is better suited to an overall object-oriented approach. Or its use may be the result of the specific motivation of the development team interested in using OOT. This use of OOT is more "bottom-up." These initial applications of OOT are often called pilot projects, in which the organization is interested in learning whether the new approach is practical. There is no requirement for all domains or enterprise-wide software development to be object oriented. The degree of reuse may vary according to the specific reuse goals of the organization.

b. Domain-Based Approach. A domain-based approach is one in which the use of OOT is not mandated across the entire enterprise, but has been adopted for a limited number of application domains. In this approach, there is planning for reuse since multiple systems will be developed within the individual domains.

c. Enterprise-Based Approach. In the enterprise-based approach, the implementation of OOT is adopted throughout the software development enterprise. In this sense, OOT is more mandated rather than just available. The use of OOT may be seen as a competitive edge, such as the ability to get products to market quicker or to respond to customer's needs faster. Long-term use of OOT is expected, and there will be an effort to build reusable libraries and architectures.

An organization having no OOT experience should consider using these three approaches in sequence. For example, after several pilot projects (i.e., applying the system/ project-based approach), the organization may identify domains of interest that offer increased reusability from object-oriented software. Using this domain-based approach to adopting OOT, additional effort may be expended in developing this software to increase the likelihood of future reuse. At some later date, OOT may permeate the organization and the use of OOT becomes an integral part of the organization's software development culture (i.e., the enterprise-based approach).

2.3 ORGANIZATIONAL READINESS

One of the issues that arises in making a change of any sort is whether the organization is ready for transitioning a new technology. This decision is unique to each organization, and an assessment of the organization can determine the degree, pace, and manner in which OOT is introduced. Whatever transition approach is used, it must consider the capabilities, resources, and willingness of the organization. For that reason, there is no single set approach for all organizations. But certain elements are required for an organization to achieve an established OOT capability:

a. Organizational commitment. Organizational commitment is required within all levels, from the development team to senior management. It will be necessary to communicate to all levels the business context and rationale for such a change.

b. Resources. Resources are essential since this transition will require new tools and training. Providing resources can be a concern especially when there are short-term pressures to reduce costs. Management needs to take a long-term perspective regarding a transition.

c. Time. Sufficient time must be allowed for an organization to absorb the changes that the new technology brings and to show benefits from its use. Here the long-term perspective and sufficient resources will sustain the period of this transition. An organization should expect a one- to five-year conversion time, depending upon the capabilities of the development team and the level of support it receives.

Not all organizations will be ready for the transition to OOT. If the organization is lacking fundamental process capabilities (e.g., a Level 1 project as described by the Software Engineering Institute's Capability Maturity Model [Paulk 1993]), then those capabilities should be addressed before attempting to transition OOT. Alternatively, the organization could acknowledge their process immaturity and still resolve to introduce OOT in a limited form, focusing on education, training, and using a low-risk pilot project. If the organization is undergoing frequent restructuring, the time may not be right for establishing a coherent OOT pilot project. If the organization has had a history of failed technology transition efforts, there may be underlying problems that should be resolved before attempting to insert OOT.

2.4 SUPPORT FOR OOT TRANSITION

When an organization decides to introduce OOT, it is often helpful to acquire external support for the transition. External support may consist of OOT training and education in development approaches and object-oriented languages and tools, as well as on-site mentoring and consultation for specific projects. It may also be helpful to get basic software engineering support if the development teams are unfamiliar with issues such as software life cycle management, configuration management, and quality control.

There are several sources of support for technology transition. The DISA Center for Software provides support for technology transition readiness assessments, OOT guidance, software process assessment, domain engineering, and reengineering guidance. The DISA OOT Program also has a list of OOT consultants. The number of commercial sources of OOT support (e.g., OOT products, services, consulting, training, mentoring, books, videos, and CD-ROMs) is increasing rapidly [Gaumer 1995]. Sources of information can be found in OOT magazines and journals such as the *Journal of Object-Oriented Programming (JOOP)* and *Object Magazine*. *Succeeding with Objects* by Adele Goldberg and Kenneth Rubin [Goldberg 1995] is one of the first books solely dedicated to organizational OOT transitions. DISA also has a number of publications that are available to support both OOT and general technology transition. These publications include the following:

- *Software Technology Transfer Today*: Quarterly newsletter that provides news about various software-related DISA programs.
- *Object-Oriented Technology Training Survey Report*, *Version 1.0*, January 6, 1995: Catalog of publicly available training courses in OOT and a recommended OOT training curriculum for technical, functional, and management personnel.
- *Domain Engineering Process, Version 2*, April 1995: Description of the basic activities of domain engineering with the intent to construct domain-specific architectures and identify opportunities for software reuse.
- *Strategies for the Use of Object-Oriented Technology*, July 1995: Guidance on using OOT for analysis, design, and programming of DoD information systems, covering the situations of new development, reengineering, and legacy system wrapping.

- *Extended Object-Oriented Technology Example*, December 15, 1994: Example of the Booch method of object-oriented development applied to an information system development.

To acquire these publications or obtain support for technology transition, contact the following DISA points of contact:

- Lloyd Anderson, Chief, Technology Transition Division, Software Environments Department, Center for Software, 703-681-2348, email address: andersonl@cc.ims.disa.mil.
- David Diskin, OOT Program Manager, Software Environments Department, Center for Software, 703-681-2310, email address: diskind@cc.ims.disa.mil.

2.5 CONTEXT OF OOT TRANSITION

Figure 2 presents an IDEF0 diagram depicting the general context for the transition of OOT into an organization. Chapter 3 of this document describes in further detail the OOT Transition Process noted by A0. The objective of this process is to take an existing software development capability and transform it into an established OOT capability. This transition will be affected by a number of factors, including any existing policies, standards, technical

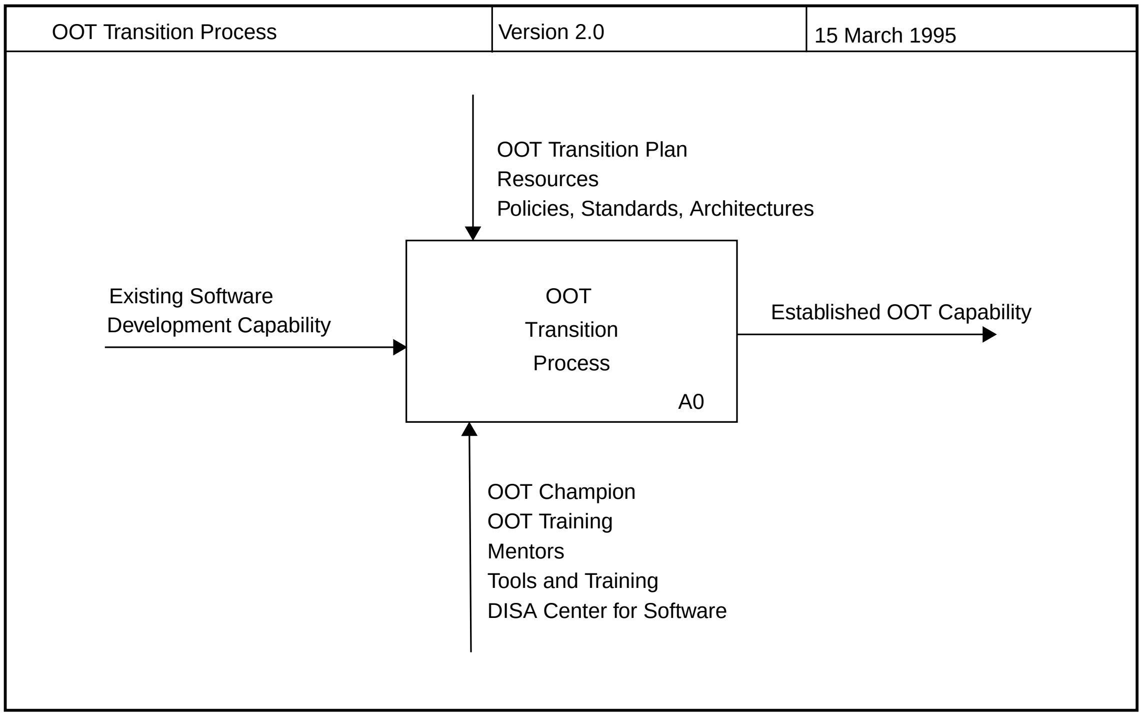


Figure 2. Context Diagram of OOT Transition Process

or system architectures, and by any resource limitations. An OOT transition plan should consider these factors in managing the organization’s transition.

The primary mechanisms for making OOT transition occur are the organization’s developers and managers who have an interest in OOT. Often there is an “OOT champion,” a single, highly motivated individual, who pursues and manages this transition. An OOT champion is not required for successful transition of OOT, but such a person is extremely useful in guiding the transition process. Of course, the transition will also require tools, training, and specialized individuals called “mentors” who work on an individual basis with developers to teach OOT concepts. As discussed in Section 2.4, DISA can provide a range of assistance in the transition to OOT.

Other technology transition processes have been developed that are similar to the one presented in this document. For example, the Software Productivity Consortium has a generalized technology transition process that has the following steps [SPC 1993]:

a. Understand Context.

b. Analyze Risks and Select Strategy.

c. Plan Technology Implementation.

d. Implement Technology.

e. Review and Update Transfer Plan.

The process embodied within Figure 2 and described in this document can be viewed as an instantiation of a generalized process, in that it is specifically focused on transition of OOT. Additional guidance on OOT transition can be found in Yourdon’s recent book [1994], which devotes several chapters to getting started with OOT. Yourdon discusses the issues associated with revolutionary vs. evolutionary transition to OOT, the political pros and cons of transitioning to OOT, the importance of training in a successful OOT transition, and how to develop an overall organizational “battle plan” for OOT transition.

3. OOT TRANSITION PROCESS

The overall process for transitioning OOT to an organization is depicted in Figure 3, and consists of four major steps:

a. Perform Planning. The OOT champion explores the applicability of OOT to the organization and works to obtain management commitment to proceed. With this commitment, the executors of this transition, whether software development managers or OOT champions, conduct detailed planning for the initial pilot project.

b. Conduct Training. OOT champions, managers, users, developers and support personnel receive appropriate training and education in OOT concepts, tools, and languages.

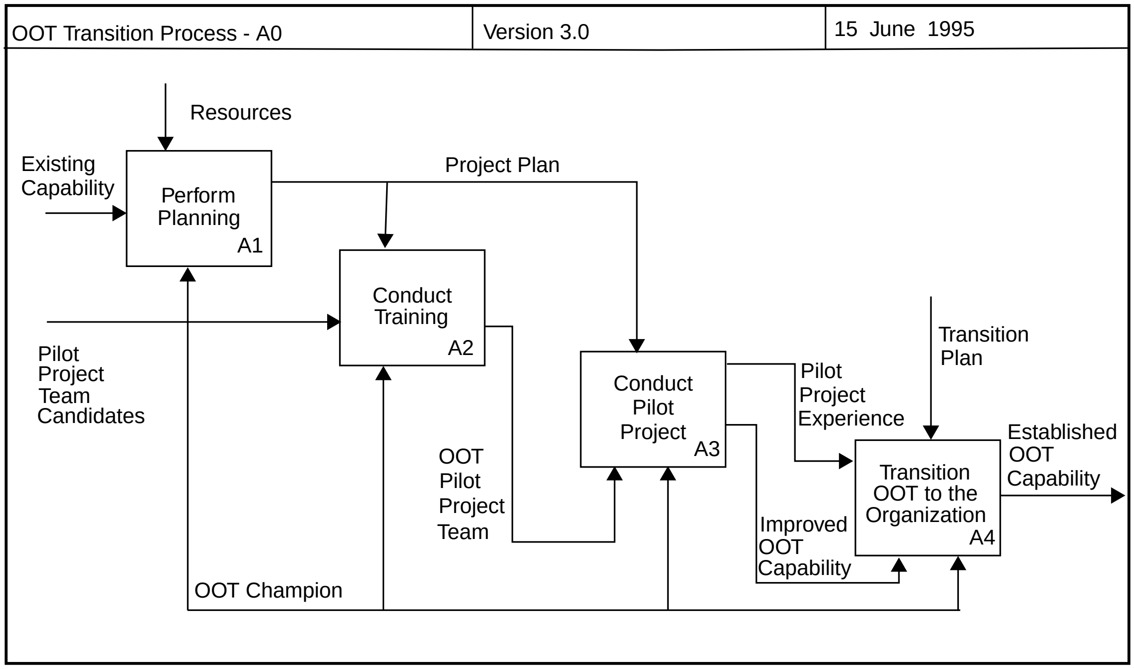


Figure 3. OOT Transition Process

c. Conduct Pilot Project. The organization tries OOT on a small scale. These early projects should be real projects, but not critical or high-risk ones. Although this document describes the conduct of a single pilot project, the organization should not feel precluded from conducting multiple, concurrent pilot projects.

d. Transition OOT to the Organization. Given successful experiences resulting from the early pilot projects, the organization may decide to transition additional software development to using OOT.

Although Figure 3 implies that all four steps in the transition process are separate and distinct, they can be combined or mixed as needed. For example, although it does not make much sense to conduct a pilot project before obtaining training, these activities can (and should) be closely coupled. Some organizations may find the suggested activities within the steps more useful than the actual formalized process depicted in Figure 3.

As mentioned in Chapter 2, an OOT champion is not absolutely required for successful OOT transition. Nor does the OOT champion necessarily embody a single individual. For example, management may perform a surrogate role as OOT champion. The key idea is that organizational motivation for using OOT exists, and that sound technical guidance is available to the pilot project team members.

3.1 PERFORM PLANNING (A1)

Purpose The purpose of this step is to develop a detailed understanding of the resources, activities, and commitment that will be required to conduct an OOT pilot project.

Activities This step consists of the following three activities (see Figure 4):

a. Perform Preliminary Planning. Develop an overall understanding of OOT, determine whether there is initial management interest in performing a pilot project, and prepare a high-level description of a OOT pilot project within the organization.

b. Cultivate Management Commitment. Provide management with preliminary plans for a pilot project, identifying necessary commitments of time, money, and personnel.

c. Perform Detailed Planning. Obtain management commitment and ensure that the initial OOT pilot project is a success.

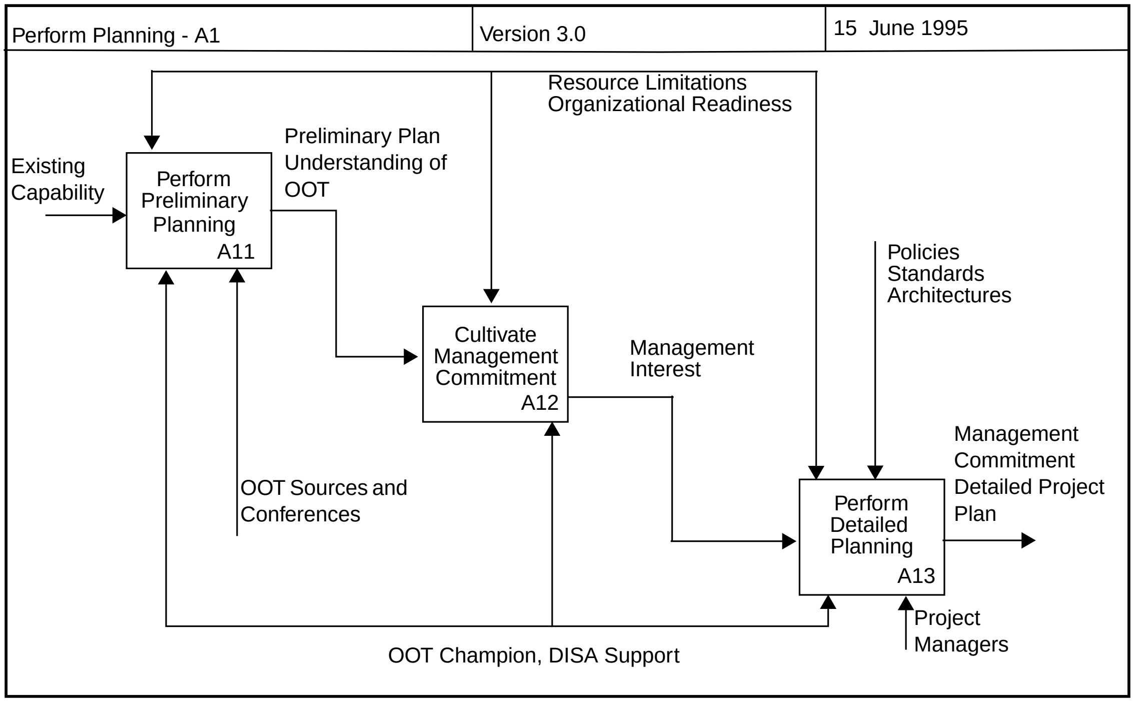


Figure 4. Perform Planning Step

3.1.1 Perform Preliminary Planning (A11)

Description The first activity of the technology transition process involves preliminary planning to develop an overall understanding of OOT, to determine initial management interest in performing a pilot project, and to prepare a high-level description of the OOT pilot project within the organization. The desired result is approval for the pilot project in the next activity, Cultivate Management Commitment (A12).

Inputs	• Existing Capability
Controls	• Resource Limitations
	• Organizational Readiness
Outputs	• Preliminary Plan
	• Understanding of OOT
Mechanisms	• OOT Champion
	• DISA Support
	• OOT Sources and Conferences

Considerations The actions in this activity are primarily performed by someone within the organization who will be the OOT champion. The champion is the focal point for the organization's OOT knowledge during the pilot project and is responsible for most of the pilot project planning that relates to OOT. Selecting an appropriate champion is an important factor in the success of the technology transition process. The OOT champion should possess effective people skills to enable successful interaction with management and project personnel. The champion may not necessarily possess extensive OOT knowledge at the outset of the technology transition process; an organization may have to “grow” such a champion. The champion will often be found within an organization's software engineering process group or may be a technology advocate working on a specific project. For the purposes of this document, it is assumed that the champion is a single person. However, it is entirely possible that there is a small group of technology advocates who champion the introduction of OOT.

The preliminary planning activity may require up to several months to complete, depending on the level of OOT understanding possessed by the champion and the proportion of the champion's time allocated towards the activity. If the champion lacks technical depth in OOT, gaining that knowledge will involve obtaining training in OOT, reading the literature on OOT, talking with people who have experience in OOT, discussing with people the opportunities for and interest in applying OOT within the organization, and developing OOT software. Understanding the many different object-oriented methods and their terminology will take time, as well. For example, Appendix A provides a range of terminology used by contemporary object-oriented development methodologies. Developing a suitable understanding of these object-oriented terms and concepts will be important in understanding the many different object-oriented methods that exist.

3.1.2 Cultivate Management Commitment (A12)

Description In the early stages of the transition process, it is important to cultivate management's interest in, commitment to, and sponsorship of OOT. Even if the idea for an OOT pilot project originated with management, the OOT champion must ensure that management's expectations of OOT are not set too high, that it understands organizational resources will be needed to conduct the pilot project, and that investment in education and training will be required. Management interest can be cultivated informally at this stage of the process; a more formal treatment occurs in the next activity, Perform Detailed Planning (A13).

Inputs	• Preliminary Plan
	• Understanding of OOT
Controls	• Resource Limitations
	• Organizational Readiness
Outputs	• Management Interest
Mechanisms	• OOT Champion
	• DISA Support

Considerations The OOT champion should identify a manager, or group of managers, within the organization who would have an influential role in an OOT pilot project. Example managers might be potential project managers who will be heading up a forthcoming software development project, a group-level manager responsible for a number of projects, and a senior manager interested in improving the software process. The level of management targeted by the OOT champion will depend on the particular situation. The general rule of thumb is to target one level above the manager who has direct authority over the potential project. The OOT champion should informally discuss with these managers the potential benefits of OOT, gauge interest in performing the OOT pilot project, and seek to gain approval to develop a high-level project plan and briefing.

The OOT champion should ensure that management's initial expectations of OOT are appropriate. Care should be taken to avoid overselling the benefits of OOT. Figure 5 provides a notional description of the learning curve an organization experiences in transitioning OOT [Kerth 1992]. The key point of Figure 5 is that there is a maturation period before the benefits (in this case, productivity improvements) resulting from OOT are realized. Suggestions for properly characterizing the benefits of OOT are found in Section 3.1.3.

The OOT champion should discuss with management how the OOT pilot project might be performed, what resources (e.g., training budget, people, schedule) will be required, and that the first (and perhaps subsequent) pilot projects may actually experience lower productivity than past projects due to the additional learning that will be necessary. The OOT champion should try to gain approval to develop a high-level preliminary project plan that provides an overview of how the pilot project may be performed. The OOT champion should also schedule a briefing to management summarizing the preliminary pilot project plan.

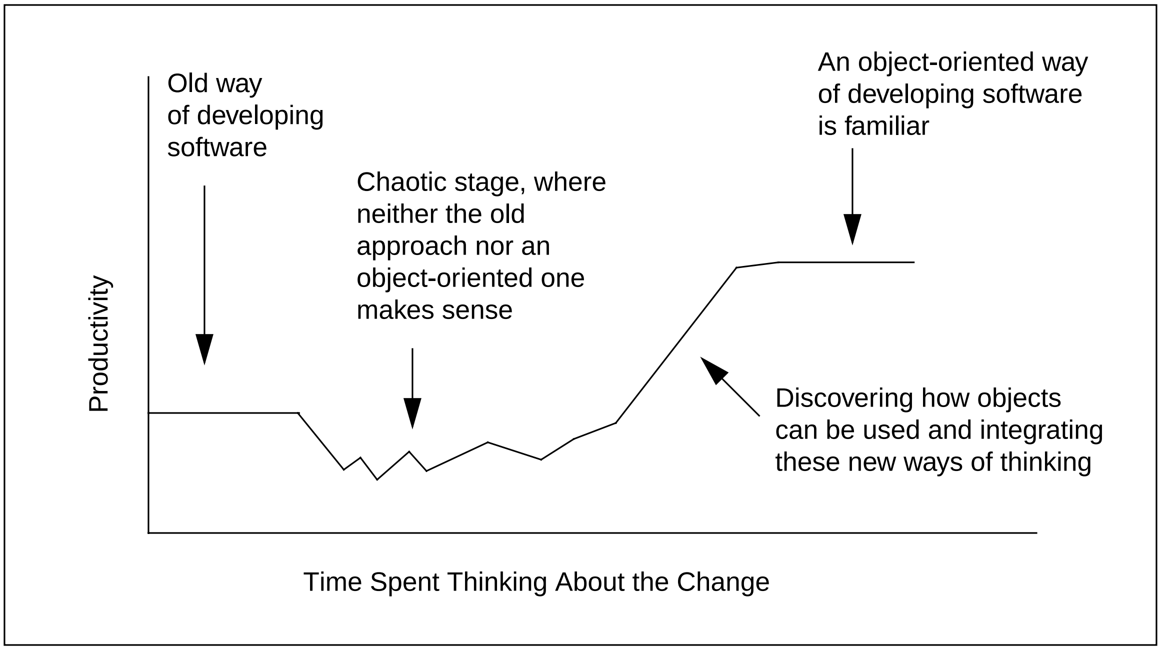


Figure 5. OOT Learning Curve

3.1.3 Perform Detailed Planning (A13)

Description The purpose of this activity is to develop a pilot project plan that can be briefed to management in order to obtain their approval and commitment to the project. The pilot project plan is prepared by the OOT champion and should discuss a number of topics: 1) justification for performing an OOT pilot project, 2) candidate or selected projects for the pilot, 3) overview of expected resources needed for the pilot, and 4) review of known pilot project pitfalls. An outline of an example pilot project plan is found in Appendix B.

Inputs

- Management Interest

Controls

- Policies, Standards, Architectures
- Resource Limitations
- Organizational Readiness

Outputs

- Management Commitment
- Detailed Project Plan

Mechanisms

- OOT Champion
- DISA Support
- Project Managers

Considerations **Benefits.** There are various reasons why an organization should consider performing an OOT pilot project, and most of them stem from a desire to increase software development productivity. Taylor [1990] provides a brief high-level description of the potential benefits to an organization from adopting OOT. His list of benefits includes the following:

a. Faster development due to support for constructing systems from standard objects, reusing existing models of corporate processes, and replacing conventional development phases with rapid prototyping.

b. Higher quality by assembling programs from existing, proven components rather than writing programs from scratch.

c. Easier maintenance stemming from higher quality software and better organization of data and functions.

d. Reduced cost due to assembling programs from components rather than writing them from scratch.

e. Increased scalability due to improved modularization and polymorphism.

f. Better information structures for representing complex information.

g. Increased adaptability due to inherent support for making local changes without rebuilding entire systems.

Risks. The use of OOT is not a panacea for solving all software development problems. Taylor also provides the following partial list of potential concerns, or risks, to be considered:

a. There is a shortage of qualified personnel with object-oriented development experience.

b. The costs of converting an organization to OOT are significant, in terms of new languages, databases, tools, and training.

c. Support for large-grained modularity (e.g., composite objects) is not supported as well as it is at the fine-grained (e.g., simple objects) level.

Success Criteria. The OOT pilot project should have an associated list of success criteria to ensure that all interested parties understand how the project will be evaluated. This list will help to communicate a common understanding of the project's expectations and prevent misunderstandings about the project's desired outcomes. The list should be developed by the OOT champion and coordinated with the appropriate managers (both within and above the pilot project) and development personnel. In addition to the overall project success criteria, the OOT champion should develop criteria specific to the OOT transition effort.

The list of success criteria should be documented in the project plan, and should clearly state the desired outcome and how the outcome will be measured. An example success criterion is the following:

> At least 15 software engineers will obtain training and experience in developing object-oriented systems. These engineers will be proficient in developing object classes in Ada 95. This expertise will be used to determine whether further organizational OOT transition efforts are warranted.

For the first pilot project, success criteria related to cost, schedule, and resources should be avoided due to the learning curve associated with introducing a new technology. However, this does not mean that the pilot project should not measure the effects of OOT on cost, schedule, and resources.

The actual change in software development productivity resulting from OOT is extremely difficult to measure. Few organizations understand their current software development productivity (e.g., effort in phases, levels of rework) in sufficient detail such that they can accurately assess the effect of introducing a new technology (e.g., OOT). Too often, reported increases in productivity attributable to OOT are based more on intuition or gross measures. In addition, the first few pilot projects that use OOT may experience a decrease in productivity due to the associated technology learning curve. Nonetheless, one should expect at least a moderate increase in software development and maintenance productivity once the organization has used OOT on several projects.

The organization should give careful consideration to various measures in order to adequately characterize the effect of OOT. Parkhill [1992], for example, suggests tracking the following process indicators during the initial OOT pilot projects:

a. Design time (calendar and staff months).

b. Implementation time (calendar and staff months).

c. Resulting code size (function points or lines of code).

d. Number of defects by severity by phase.

e. Time to repair defects.

f. Reuse of components from existing libraries.

g. Code available for reuse on future projects.

h. Reuse cost savings.

i. Training cost (time and course fees).

j. Mentor costs.

A more recent source of metrics tailored to object-oriented software is found in a recent book by Lorenz and Kidd [1994]. This book covers a range of project and design metrics, and provides the authors' recommended set of metrics to collect during an object-oriented development project.

Reuse and Software Engineering. The potential for increasing the amount of reuse during development is another reason for transitioning to OOT. Anecdotal evidence suggests that the effort spent developing suitable objects within a particular domain leads to higher reuse potential rather than attempts at reuse using traditional structured development methods.

Another reason for transitioning OOT into an organization is to introduce development staff to modern software engineering principles such as information hiding, abstraction, and inheritance. Some software developers may not have been exposed to nor have applied these principles in practice. In their exposure to OOT, these developers will gain training in these principles. Thus, OOT becomes a mechanism for improving the general skill level of an organization. Yet another reason for performing an OOT pilot project is simply to gain an understanding and appreciation of the technology in order to determine how it might fit into the organization. Organizations typically do not experiment with new technology for the sake of doing so. This experimentation occurs to identify better ways of developing software. OOT is clearly one of the more highly publicized new technologies that have surfaced in recent times.

Selecting a Pilot Project. Once the justification for performing an OOT pilot project has been developed, a list of candidate projects for the pilot should be identified. Depending on the organization, there may be many or few candidate projects available to perform an OOT pilot. The following guidelines offer suggestions on the selection of the pilot project:

a. Do not select very high risk, highly visible, or schedule-driven projects. Such projects may not provide a good environment for a pilot project because the product development emphasis discourages absorbing the OOT learning curve.

b. Avoid selecting a maintenance effort for the pilot project unless a subset of new development within that effort can be isolated. The OOT pilot project may need to perform full life cycle activities (e.g., requirements, design, code); thus, incremental maintenance would not be an effective phase in which to use OOT. The pilot project should be either a new development, a segregated portion of new development within an existing maintenance project, or a reengineering effort.

c. If possible, the OOT champion should limit the number of new technologies being demonstrated on the pilot project. A new technology is defined as one that requires formal training by project personnel. Having too many new technologies on a pilot project increases the risk of project failure and dilutes the ability to determine which technology lead to various results. Ideally, at most one other new technology (e.g., client/server, Ada) should be allowed. This does not include refresher training in areas that should already be practiced (e.g., fundamental software engineering skills).

d. The pilot project should not be a “toy project” involving only a couple of people for several months. Nor should it be a large project involving dozens of people for several years. Part of the purpose of the pilot project is to expose a small number of people to a realistic OOT development effort. A pilot project size of 5 to 20 people over 6 to 12 months is recommended.

Once a list of potential pilot projects has been assembled, the OOT champion should begin to characterize the resources likely to be required. These resources involve the number and type of people involved in the pilot project, the duration of training each person will receive, the actual costs associated with providing the training and mentoring, and the possible costs involved in procuring automated tools. Clearly, the OOT champion cannot precisely predict the actual costs. However, rough estimates should be made in order to provide management with a better understanding of the scope of resources required for the pilot project.

Review Potential Pitfalls. Another activity in preparing a preliminary pilot project plan is to review known pitfalls associated with pilot projects to reduce their likelihood of occurrence. Potential pitfalls associated with introducing OOT include the following:

a. Insufficient training. One of the most important aspects of any technology transition effort is providing appropriate training at the right time to the participants. If adequate training is not provided, the risk of pilot project failure is greatly increased. Similarly, the training should occur just before the skills are applied. The OOT champion should investigate the various types of training that are available, their duration and cost, and their approach in training participants. See Section 3.2 for many training considerations.

b. Inappropriate management expectations. The OOT champion must ensure that management has an appropriate set of expectations regarding both the pilot project and OOT in general.

c. Organization not receptive to process improvement/experimentation. Some organizations do not provide a receptive climate for improving the software development process. Most forms of technology transition or process improvement are futile for these organizations. Perhaps the best strategy would be to start small by initiating a pilot project within a group that the OOT champion has control (or strong influence) over.

Management Briefing. A management briefing should be prepared by the OOT champion summarizing the information developed in previous activities. The audience for this briefing is the managers who have approval authority over the resources that are likely to be needed for the pilot project. If possible, senior management should be included to provide additional support and commitment for the pilot.

The briefing should provide an overall understanding of OOT, an assessment of the potential benefits from adopting OOT, and a presentation of the topics developed in the preliminary pilot project plan. In particular, the OOT champion should ensure that management's expectations are properly set (e.g., do not lead them to expect radical productivity improvements from OOT). Other aspects of the preliminary pilot project plan, such as the cost and type of training, potential pitfalls, should also be discussed.

If more than one project is available for the first OOT pilot, the specific advantages and disadvantages of using OOT should be discussed for each individual project. Multiple projects may be selected if there is interest in obtaining feedback more quickly.

In concluding the briefing, the OOT champion should identify the resources needed for the pilot project, ask the appropriate managers to commit to providing these resources, and obtain active sponsorship of the OOT pilot program. The needed resources should have been discussed in the preliminary pilot project plan, and include people, funding, time, the pilot project itself, preparation of detailed pilot project plan, training, and risk mitigation.

3.2 CONDUCT TRAINING (A2)

Purpose The purpose of this step is to establish a well-understood competency in object technology by all participants in the development process. These participants include not only the early adopters and project teams, but managers as well. Training is a critical part of any technology transition effort. Without adequate training, the risk of failure in introducing a new technology can be very high. OOT training is particularly important because the participants often need to “unlearn” software development processes that they may have practiced for many years. The transition to OOT often involves a shift in how software development is approached, and this shift is best accomplished by hands-on training.

DISA has developed a set of abstracts that describe a curriculum of OOT courses [DISA 1995]. Recommended courses include an executive overview, OOT overview, OOT project management, and various technical courses related to object-oriented analysis, design, and programming.

Activities This step consists of the following three activities (see Figure 6):

a. Train Early Adopters. Establish a core group of people with object technology knowledge and expertise.

b. Educate Management. Establish a proper understanding of the technology and a realistic set of expectations.

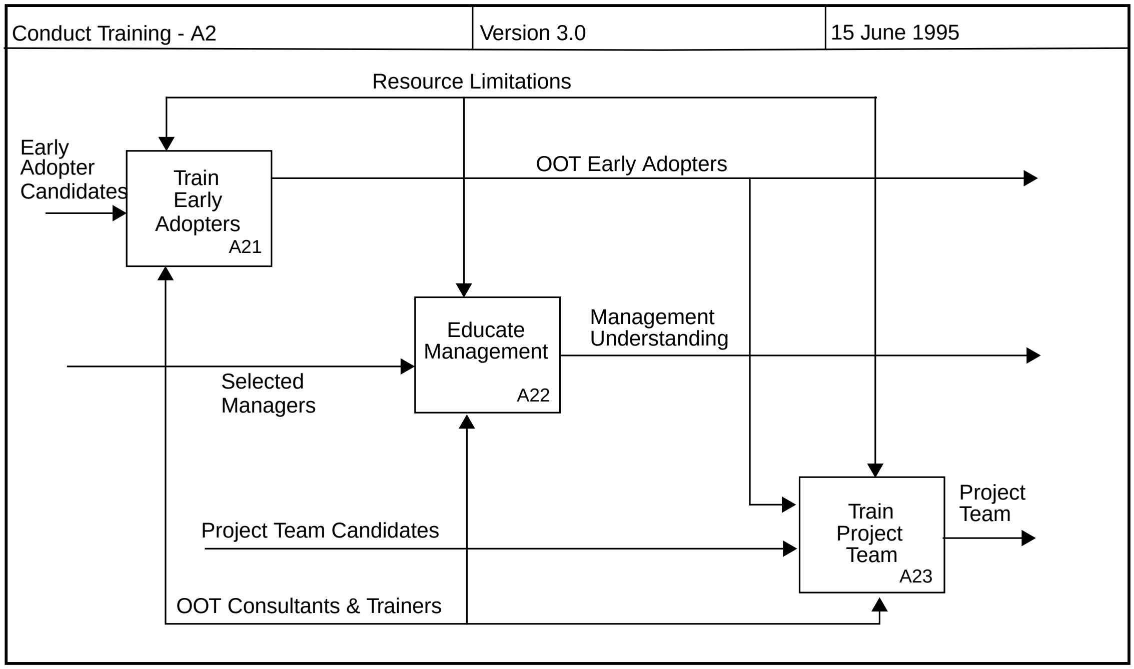
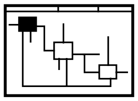


Figure 6. Conduct Training Step

c. Train Project Team. Establish a capability for conducting the pilot project.

3.2.1 Train Early Adopters (A21)



Description This optional activity consists of training the OOT champion and early adopters in the organization. Before the pilot project's managers and development team are trained in object-oriented methods, the early adopters may need to gain additional proficiency in OOT.

Inputs	• Early Adopter Candidates
Controls	• Resource Limitations
Outputs	• OOT Early Adopters
Mechanisms	• OOT Consultants and Trainers

Considerations Early adopters are people within an organization who are receptive to learning and applying a new technology. For this activity, early adopters can provide advice on the initial phases of the OOT transition effort, before the actual start of the pilot project. Thus, the early adopters augment the role of the OOT champion. In fact, the OOT champion is an early adopter. The early adopters may be drawn from the members of the pilot project team.

The training of early adopters is considered optional, since the organization may not have the resources available to train two separate groups of individuals (i.e., the early adopters and the pilot project team members). If this activity is not performed, the pilot project team members should be considered the early adopters.

The scope of the training for the early adopters does not have to be as comprehensive as that provided to the pilot project team. One- or two-day OOT tutorials are often available at professional conferences. OOT-related conferences can provide tutorials on the fundamentals of object orientation, different object-oriented methodologies and tools, and techniques for applying object-oriented concepts in contemporary programming languages. A list of over 250 providers of OOT training and mentoring can be found in JOOP [1995]. Care should be taken in selecting a source of training. Consideration should be given to the trainer's experience in object-oriented analysis, development, and management.

Several companies offer OOT training at various major cities on a lecture circuit across the country. These companies often hire an OOT expert who can provide an excellent training in the technology that is being considered for the pilot project.

3.2.2 Educate Management (A22)

Description The managers directly involved in the OOT pilot project should receive an education in the fundamental aspects of OOT and how the technology will be applied. This education can take the form of a briefing or seminar course. The purpose of the briefing or course is to provide managers with a working knowledge of OOT so that they understand how the technology may affect the development process.

Inputs	• Selected Managers
Controls	• Resource Limitations
Outputs	• Management Understanding
Mechanisms	• OOT Consultants and Trainers

Considerations This education should be targeted to the organization's managers, emphasizing the value of OOT to the organization overall rather than the technical aspects. Care should be taken to avoid overselling OOT to management; initial pilot project results may be less productive due to training costs and the technology learning curve.

If possible, the educational effort should be obtained from an outside source, such as consultants and trainers. The DISA Object-Oriented Technology Training Survey Report provides an overview of several different courses that provide management with an understanding of OOT [DISA 1995]. For example, the report describes the following three courses:

a. OOT Executive Overview. This course should introduce the fundamentals of OOT to upper-level management. The course should discuss the benefits of the technology along with risks and associated mitigation strategies. The material presented should enable the student to easily translate the concepts to business terms that will facilitate decision-making and transition strategy development.

b. OOT Overview. This course should serve as the foundation for all subsequent OOT training. It should introduce object-oriented concepts, terms, and principles along with their benefits and drawbacks. The course material should relate the topics to a traditional development methodology to reinforce the concepts. It should discuss the complete object-oriented development life cycle and introduce the types of languages, tools, and methodologies that support the technology.

c. OOT Project Management. This course should present the effects that OOT has on the way that software projects are managed. Topics such as new organization roles, development life cycle considerations, and methodology/language selection should be covered. This course should recommend ways to maximize the potential benefits of OOT and to identify and mitigate the risks that are inherent in the transition to this technology.

3.2.3 Train Project Team (A23)

Description All members of the pilot project team should receive training in the object-oriented concepts, tools, and processes that will be used during the pilot project.

Inputs	• Project Team Candidates
	• OOT Early Adopters
Controls	• Resource Limitations
Outputs	• Project Team
Mechanisms	• OOT Consultants and Trainers

Considerations The point in time at which training is provided is an important consideration. Ideally, the training should be provided as close as possible to the start of the pilot project. The participant's OOT knowledge will be effectively reinforced if applied immediately after the training course (i.e., "just-in-time" training). If the training is provided too early, participants may not retain their OOT knowledge by the time they are scheduled to begin the pilot project.

Another training consideration is how strongly the programming language is mapped to the object-oriented concepts being taught. For example, some trainers prefer to teach object-oriented concepts using "classical" object-oriented programming languages such as Smalltalk and Eiffel. These languages facilitate learning of object-oriented techniques, which the trainees then adapt to their production environment. Other training courses focus on how to map object-oriented concepts into a particular programming language (e.g., C++, Ada 95). A combination of these two training philosophies may be best, whereby trainees initially learn object-oriented concepts using classical object-oriented programming languages, and then learn how to adapt these techniques into the programming language and environment to be used in the pilot program. Given the steep learning curve associated with OOT, project team members should be provided the equivalent of a three-hour semester college course.

The OOT champion should consider receiving additional OOT training in order to improve his or her ability to serve as an onsite "mentor" for the pilot project team. This additional training may include attendance at OOT conferences to increase exposure to the state of OOT, detailed training in particular object-oriented development methods, or additional training in how to map object-oriented concepts into particular programming languages. The OOT champion is strongly encouraged to obtain the services of an experienced OOT consultant to aid in the mentoring role during the first pilot project.

3.3 CONDUCT PILOT PROJECT (A3)

Purpose The purpose of this step is to begin using OOT on the pilot project. The pilot project will enable the organization to improve its understanding of OOT and help to identify refinements that can be made in training, tools, the overall transition process, development process, and the software management process.

Activities This step consists of the following three activities (see Figure 7):

a. Acquire and Install Project Materials. Acquire resources such as tools, libraries, and mentoring/consulting services.

b. Execute Pilot. Exercise and test OOT and the team's capability.

c. Review and Assess Pilot. Suggest any refinements to the project structure and process and in the overall transition process.

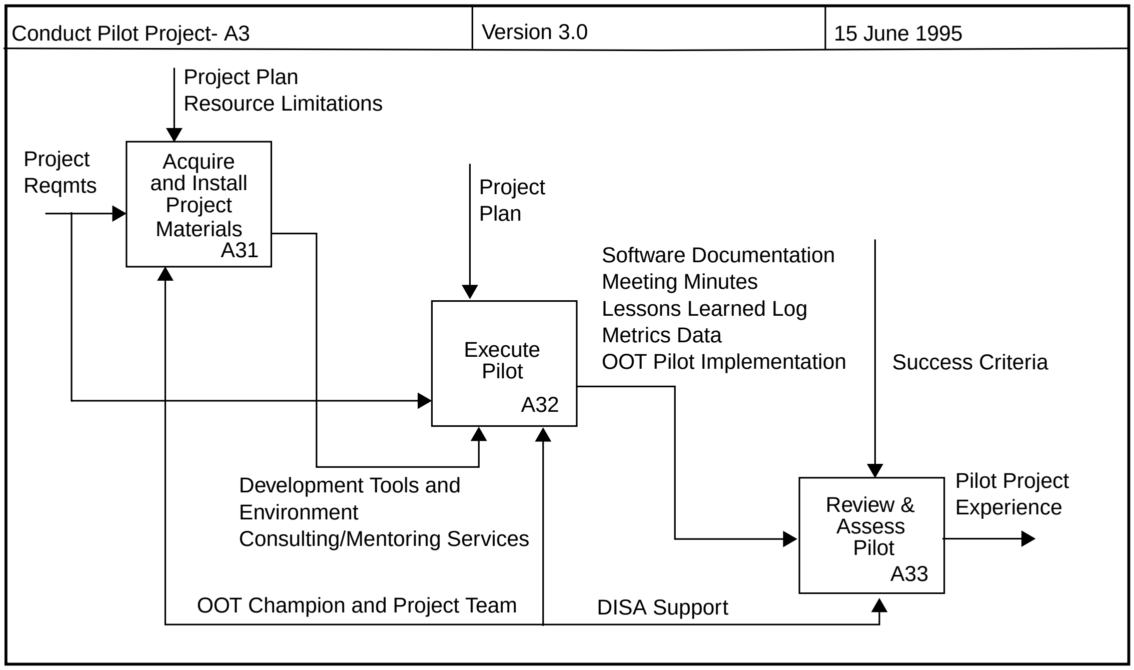


Figure 7. Conduct Pilot Project Step

3.3.1 Acquire and Install Project Materials (A31)

Description This activity consists of identifying, evaluating, and acquiring the OOT-related project materials and resources that will be needed during the pilot program.

Inputs	• Project Requirements
Controls	• Project Plan
	• Resource Limitations
Outputs	• Development Tools and Environment
	• Consulting/Mentoring Services
Mechanisms	• OOT Champion and Project Team
	• DISA Support

Considerations A key consideration for the pilot project is the evaluation, selection, procurement, and installation of any object-oriented tools and libraries to be used during the pilot project. The OOT project team needs to perform a review of available technology that may be useful for the pilot. Tools are often dependent on a number of factors, such as design methodology, development language, host platform, database system, and network. The review process should occur concurrently with the consideration of these factors, as well as the type of training to be provided to pilot project team members. The time and effort involved in installing the tools and libraries should not be overlooked. An organization should expect at least two months for tool setup and installation.

Additional development platforms may need to be acquired or current ones upgraded to support the object-oriented development tools and libraries. For example, if the organization's current platforms are Sun workstations running Solaris and the target platform is a PC running Windows, then the necessary tools and libraries may require PC platforms with Windows. The new tools may also require additional memory (both RAM and hard disk) and upgraded monitors.

If a consultant is to be used to assist in the OOT pilot project, appropriate planning should be performed to ensure services are available when needed. The OOT consultants and trainers considered in the Train Early Adopters (A21) activity should have provided the information needed to select the consultant. The OOT champion should ensure that the administrative process for obtaining the consultant is complete or nearly complete.

3.3.2 Execute Pilot (A32)

Description Once the pilot project planning has completed, the pilot project is ready to begin.

Inputs	• Project Requirements
Controls	• Project Plan
Outputs	• Software Documentation
	• Meeting Minutes
	• Lessons Learned Log
	• Metrics Data
	• OOT Pilot Implementation
Mechanisms	• OOT Champion and Project Team
	• Development Tools and Environment
	• DISA Support
	• Consulting/Mentoring Services

Considerations The primary responsibility of the OOT champion during this phase is to provide support to the pilot project team members. The team will be faced with many issues associated with object-oriented development, and having access to a more knowledgable or experienced person will greatly help. This person may not be the OOT champion; support may be provided from outside the organization such as independent consultants, DISA, or others providing OOT training to the team.

As part of a lessons-learned logging activity, the OOT champion should hold periodic in-process reviews of the pilot project in order to discuss how well the OOT activities are proceeding, problems they have run into, key issues that the team has addressed, and metrics being collected. Such reviews provide an excellent opportunity to capture valuable suggestions for the next phase, Review and Assess Pilot (A33).

3.3.3 Review and Assess Pilot (A33)

Description A review of lessons learned should be performed upon completion of the pilot project. Care must be taken to avoid over-generalizing the pilot project's results. Although the first pilot project will provide valuable insight into the use of OOT, the true effect on an organization may be difficult to determine until several pilots have been completed.

Inputs	• Software Documentation • Meeting Minutes • Lessons Learned Log • Metrics Data • OOT Pilot Implementation
Controls	• Success Criteria
Outputs	• Pilot Project Experience
Mechanisms	• OOT Champion and Project Team • DISA Support

Considerations As was discussed earlier, the expectations for the pilot project should not have been set too high, since the learning curve associated with OOT may initially result in lower productivity for the initial pilots. However, the pilot project participants should be able to characterize their reactions to the use of OOT. The OOT champion should identify OOT effects in order to determine whether to continue with transition of OOT within the organization (e.g., performing another pilot project). Other sources of information can be used to determine the effects of the pilot project's use of OOT:

a. Defect data resulting from the development effort may be useful in comparing with previous defect rates.

b. Class libraries created and their likelihood for reuse should be examined.

c. The lessons learned log should provide a record of the experiences of the pilot project team.

d. The success criteria established in the pilot project plan should be reviewed.

The OOT champion should review the pilot project's results with management to determine whether or not to continue with the introduction of OOT into the organization. If further OOT transition is desired, the

OOT champion should publicize the results of the first pilot project within the organization and begin planning for another pilot project.

A key barrier to software technology transition is the human tendency to reject new methods in favor of previously used methods, even if the older methods are not particularly successful. Publicizing the results of the pilot project helps to lower this barrier, in that OOT becomes more familiar to the staff. The OOT champion, or a pilot project member, should provide a brief description of OOT and how it was applied on the pilot project. Results of the pilot project, both positive and negative, should be presented. The pilot project results can be disseminated in a organization newsletter, a memorandum, or a presentation to the staff.

3.4 TRANSITION OOT TO THE ORGANIZATION (A4)

Purpose The purpose of this step is to establish an OOT capability within the organization that is either project, domain, or enterprise based. This step assumes that the organization has found OOT to be beneficial and is interested in furthering OOT adoption. This may not be the case for all organizations (e.g., the pilot may not have been successful and further efforts may be viewed as being too risky, or pervasive OOT adoption has been determined not to be a goal).

Activities This step consists of the following four activities (see Figure 8):

- Remove Impediments. This is with regard to any remaining technical or organizational issues that may inhibit OOT use.
- Transition to Project-Based Use. Each project within the organization decides whether or not to use OOT. Reuse benefits may be limited.
- Transition to Domain-Based Use. OOT is used for all applications within a particular domain, emphasizing reuse across applications.
- Transition to Enterprise-Based Use. This is the most widespread use of OOT. An organization uses OOT across many domains and con-

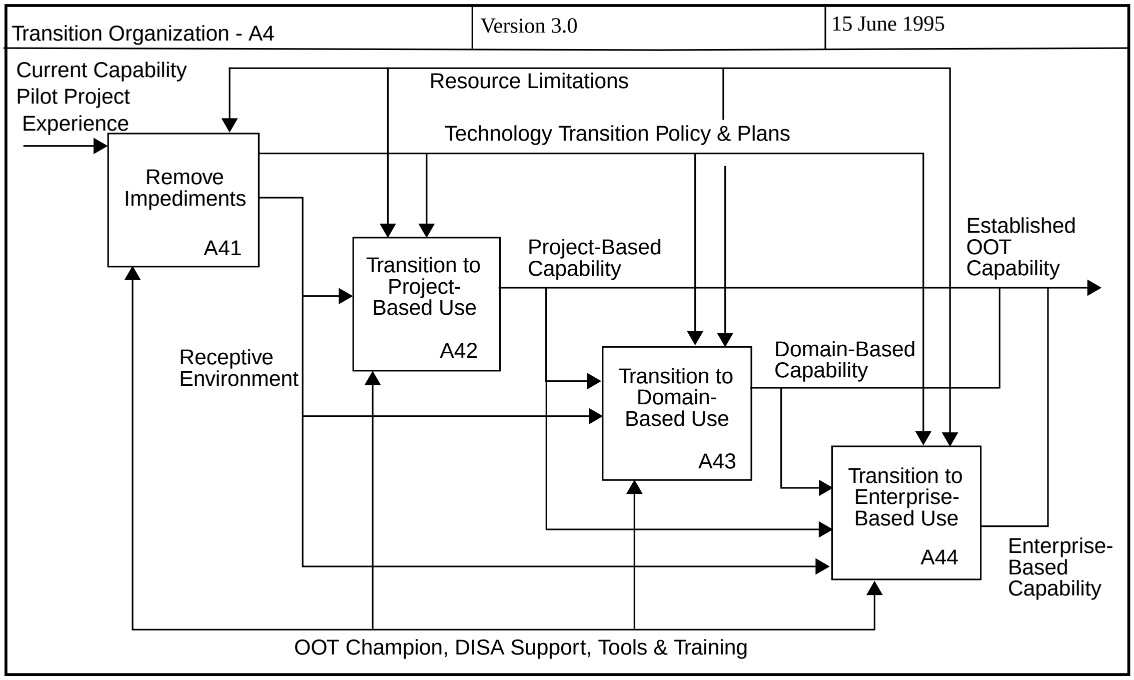


Figure 8. Transition OOT to the Organization Step

siders an object-oriented approach the default for application development.

Less guidance is presented in this section for the latter two activities because few organizations have experience (or have described their experiences) in transitioning to domain or enterprise-wide OOT usage. The reader primarily interested in initiating an OOT pilot project should only briefly review this section.

3.4.1 Remove Impediments (A41)

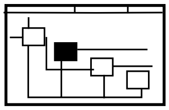
Description This activity consists of removing any remaining impediments to using OOT. These impediments are things such as organizational policy, resources, process, methodology, or documentation requirements.

Inputs	• Current Capability
	• Pilot Project Experience
Controls	• Resource Limitations
Outputs	• Receptive Environment
	• Technology Transition Policy and Plans
Mechanisms	• OOT Champion
	• DISA Support
	• Tools and Training

Considerations Although an organization may have been successful in conducting pilot projects, there may be impediments to full-scale organizational use. Before attempting a transition to the organization, it is necessary to assess whether there are any remaining conditions within the organization that may impede the introduction and transition of OOT. Impediments can consist of existing policies regarding software development, such as the mandated use of non-OO languages or techniques. Impediments may be cultural, for example, managers may not want to adopt new techniques or languages. Dealing with any remaining impediments may also be influenced by the experience of the pilot projects. A thorough review of lessons learned will help to identify possible problems for a transition.

If an organization does decide to adopt OOT, then the OOT champion should review the experiences and knowledge gained from the pilot project(s) (e.g., benefits, costs, lessons learned) and should consider establishing an informal group within the organization to periodically discuss issues relating to OOT. Such a group may discuss experiences in using a particular tool or methodology, objects that may be reusable by other projects, and particular object-oriented concepts in more detail. People within this group may also provide a mentoring capability to future pilot project team members. All of these issues can be reflected in the Technology Transition Policy and Plans. The development of a policy and plan may help avoid repeating any problems encountered during the pilot projects. Such a policy or plan could specify the type of necessary training, tools, time required for project setup and tool installation, the size of development teams, the type of development process to be used, and any other resources or elements required to move to some form of organization OOT use. This policy could also define how OOT is to be integrated with the existing development processes and procedures and how they may need to be refined to incorporate OOT use.

3.4.2 Transition to Project-Based Use (A42)



Description This activity consists of establishing a project-based OOT capability within the organization. In this situation, OOT is used on a case-by-case basis for each application or system. Expectations for reuse on future projects may not be a primary consideration.

Inputs	• Receptive Environment
Controls	• Resource Limitations
	• Technology Transition Policy and Plans
Outputs	• Project-Based Capability
Mechanisms	• OOT Champion
	• DISA Support
	• Tools and Training

Considerations This transition is relatively easy since the demands for tools and training are driven by the needs of each specific project. Software, hardware, and training are required for only the project team members. The use of OOT may be more limited in scope to the specific system. For example, the degree of object-oriented analysis will be constrained to a system and should not necessarily have to include an entire domain. As a result, however, high expectations for reuse may not be achieved.

Additional Pilot Projects

One approach to moving to a project-based capability is the use of additional pilot projects, incrementally increasing the size of the pilots and their scope. A key consideration for additional pilots is the selection of staff, and there are three basic strategies. First, the initial pilot project team can remain intact and work on a second project. The advantages with this strategy are that training costs may be minimized and the team members can further their OOT expertise through continued practice. The disadvantage to this strategy is that it does not promote wider OOT practice within the organization. A second strategy is to select a completely different set of staff to form the additional pilot project. This strategy has essentially the reverse set of advantages and disadvantages as the previous strategy. The third strategy is to select a mix of staff, some from the original pilot and some who are not. The advantage of this approach is that it increases the organization's OOT experience base and allows those with OOT experience to serve as mentors on the project. In actual practice, other organizational factors (e.g., project selection) will strongly influence which staffing strategy is selected.

The pilot project considerations previously presented in Section 3.1 (e.g., development of a project plan that addresses OOT issues) should be reviewed and applied where applicable.

3.4.3 Transition to Domain-Based Use (A43)

Description This activity consists of establishing a domain-based OOT capability within the organization. In this situation, the use of OOT has expanded beyond project specific and is used across applications within the same domain.

Inputs	• Receptive Environment • Project-Based Capability
Controls	• Resource Limitations • Technology Transition Policy and Plans
Outputs	• Domain-Based Capability
Mechanisms	• OOT Champion • DISA Support • Tools and Training

Considerations For this type of transition, longer-term planning will be necessary since domain-based use presumes reuse across applications within the same domain. Note that this type of transition will be easier if the organization has already established expertise in object-oriented techniques and tools.

The approach to object-oriented analysis may change since the scope of modeling will exceed that of a single system or application. Previous modeling in the same domain may have to be considered when creating new object models. There may be an element of negotiation required between differing depictions of a domain, and naming and object specification conventions may be required. Early in the process it may be necessary to establish the characteristics of the domain and consider the development of a comprehensive domain model. As object models are built and software components are developed, this domain model can guide the definition of additional object components and serve as the basis for new system and object model definitions.

Although reuse will be apparent for software (code) objects and models, an organization may reuse specifications, architectures, and standards as well. An organization will need to determine what type of reuse activity is possible given its existing capability and process maturity. Goldberg and Rubin [Goldberg 1995] provide an extensive discussion on reuse, categorizing reuse into five models of ad hoc, supply and demand, expert services, product center, and COTS.

New roles and responsibilities will also appear such as class librarian or domain modeler. As with any use of OOT, domain experts are essential for understanding the domain and identifying opportunities for reuse. Goldberg and Rubin [Goldberg 1995] describe a variety of new roles such as analysis prototyper, design prototyper, object coach, object technology expert, framework designer, reuse administrator, reuse engineer, and reuse manager.

3.4.4 Transition to Enterprise-Based Use (A44)

Description This activity consists of establishing an enterprise-based OOT capability within the organization. In this situation, an organization has made OOT an integral part of the software development process. OOT is used, or at least considered, for applications in different domains.

Inputs

- Receptive Environment
- (Project-Based Capability)
- (Domain-Based Capability)

Controls

- Resource Limitations
- Technology Transition Policy and Plans

Outputs

- Enterprise-Based Capability

Mechanisms

- OOT Champion
- DISA Support
- Tools and Training

Considerations This type of transition is, by far, the most extensive and will require significant effort and commitment throughout all levels of the organization. It would be unusual to see an organization attempt this transition without having earlier made a transition to a project- or domain-based capability. Of course, this type of transition has the highest potential for exploiting OOT in terms of reuse and maintainability. The organization itself should be reasonably mature in its software development capability, having internal processes for development, documentation, configuration management, and quality assurance.

The transition time for an enterprise-based capability will probably be between three and five years, depending upon the capabilities of the developers, the level of training, education, and tools provided, and the overall organizational commitment to the transition. Organizations with less mature development teams and processes may find that OOT is not the only change they are introducing, but that they are also addressing existing deficiencies in these areas.

For an enterprise-based transition, it may be worthwhile to set up an “object technology center,” which acts as a centralized place within an organization to focus OOT expertise [Kristek 1994]. The primary mission of the object technology center is to introduce OOT into the corporate software development culture. This is accomplished by providing OOT training, mentoring, pilot project support, and tool support. The object technology center can be a place where object-oriented tools and technologies can be explored before introducing them to the organization at large, thus reducing individual project risk in tool selection. The center is useful in providing a consolidated base of OOT experience that develops over a series of projects. Such an experience base can help OOT newcomers avoid many of the technology transition pitfalls. The technology center can also facilitate management buy-in for new OOT projects, serving as a permanent, in-house group of OOT champions.

APPENDIX A. DIVERSITY OF OBJECT-ORIENTED TERMINOLOGY

One of the difficulties in understanding object-oriented concepts is the wide range of terminology and graphical notations used by contemporary object-oriented design and development methodologies. Identifying the differences and similarities between these methodologies can be confusing. The OOT champion should be aware of the diversity of terminology, especially if several methodologies are being examined for pilot project use.

Table A-1 provides a general cross-reference that Singer [1993] developed to illustrate the variety of object-oriented terms used by several popular methodologies [Booch 1994, Jacobson 1992, Rumbaugh 1990, Shlaer 1992, Wirfs-Brock 1990].

Table A-1. Examples of Object-Oriented Terminology

Booch	Jacobson	Rumbaugh	Shlaer/Mellor	Wirfs-Brock
Class Diagram	Domain Object Model, Design Model	Object Diagram	Information Model, Communications Model	Collaboration Graph
Graph Template	—	Template	—	Protocol
Inheritance	Inheritance	Generalization	Inheritance	Inheritance
Action	Activity	Activity	Action	Responsibility
Relationship	Acquaintance, Communication	Association	Relationship	Collaboration
Uses	Communication	Uses	—	Contract, Responsibility
Cardinality	Cardinality	Multiplicity	Multiplicity	—
Message	Stimulus	Event	Event	Message
Operation	Operation	Operation	Process	Method
Aggregation	Partition Aggregate	Aggregation	—	Composite

Many books have been written on the subject of object-oriented concepts. These books emphasize application of object orientation to topics such as modeling and analysis, design, and programming. Two of the more popular books are by Rumbaugh [1990] and Booch

[1994]. Both of these books examine the application of object-oriented concepts throughout the software life cycle and introduce graphical design notations.

APPENDIX B. EXAMPLE PILOT PROJECT PLAN OUTLINE

This appendix provides an example outline for an OOT pilot project plan. The purpose of this outline is to provide suggested topics to present to management in order to gain approval for initiating the OOT pilot project. This outline should be augmented by the OOT champion to include site-specific information and circumstances. The plan does not need to be overly detailed. An initial plan of three to four pages in length should be satisfactory. The plan should be updated as additional information is developed.

Pilot Project Plan Outline

1.0 Introduction

- What is OOT? Provide a brief description of OOT. Concentrate on those aspects of OOT that you plan to implement on the pilot project (e.g., object-oriented design, object-oriented programming).
- Why should our organization be interested in OOT? Describe potential benefits from learning and applying OOT.
- What is the overall proposed approach for inserting OOT? Briefly describe the pilot project approach.
- What commitments are needed from management for the pilot project and what are appropriate expectations? List any management commitments you need (e.g., resources, top-level involvement). Identify anticipated benefits resulting from the initial OOT pilot program.

2.0 Pilot Project Plan

- Identify the project selected for the OOT pilot.
- Describe how the project will apply OOT. What functionality will be implemented using OOT? What are the preliminary schedules? What aspects of OOT will be implemented?

- List resources needed. Describe training requirements (e.g., who gets trained, how long, what kind of training, who will do the training) and tools needed.
- Identify any project risks from applying OOT and how they will be monitored.
- Describe how OOT results will be measured. What are the measures of success for this project?

3.0 Potential Follow-on Activities

- Describe the preliminary approach for continuing OOT transition activities based after completion of initial pilot.

LIST OF REFERENCES

[Booch 1994] Booch, Grady. *Object-Oriented Analysis and Design with Applications*. Second Edition, Benjamin/Cummings, Redwood City, CA, 1994.

[Capper 1994] Capper, N.P. et al. The Impact of Object-Oriented Technology on Software Quality: Three Case Histories. *IBM Systems Journal*, Vol. 33, No. 1, 1994, pp. 131-157.

[DISA 1995] Defense Information Systems Agency, Joint Interoperability & Engineering Organization. *Object-Oriented Technology Training Survey Report, Version 1.0*. January 6, 1995.

[Gaumer 1995] Gaumer, Dale. Directory of Object Technology, DG Innovations, Fort Wayne, IN, 1995.

[Goldberg 1995] Goldberg, Adele and Kenneth Rubin. *Succeeding with Objects*, Addison-Wesley, 1995.

[Jacobson 1992] Jacobson, I. et al. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, Wokingham, UK, 1992.

[JOOP 1995] The JOOP Guide to Project Mentoring. *Journal of Object-Oriented Programming*, Vol. 8, No. 1, March/April 1995, pp. 60-74.

[Jordan 1993] Jordan, Kathleen A., Richard P. Morton, and Robert P. Furick. *An Assessment of the Potential Implementation of Object-Oriented Technology in the Department of Defense*. Institute for Defense Analyses, IDA Paper P-2904, Alexandria, VA, October 1993.

[Kerth 1992] Kerth, Norman L. and Eileen Andreason. Managing the Objects: Management's Role in a Successful Transition to Object Orientation. *American Programmer*, October 1992, pp. 28-35.

[Kristek 1994] Kristek, Tom and Vijay Vaishnavi. Role of a Corporate Object Technology Center, *OOPS Messenger*, Vol. 5, No. 4, Association for Computing Machinery, October 1994.

[Lorenz 1994] Lorenz, Mark and Jeff Kidd. *Object-Oriented Software Metrics*. Prentice-Hall, Englewood Cliffs, NJ, 1994.

[OMG 1994] Object Management Group. *First Class*. Volume IV, Issue VI, December 1994.

[Parkhill 1992] Parkhill, David. Object-Oriented Technology Transfer: Techniques and Guidelines for a Smooth Transition. *Object Magazine*, May/June 1992, pp. 57-59.

[Paulk 1993] Mark C. Paulk et al. *Capability Maturity Model for Software, Version 1.1*. Software Engineering Institute, CMU/SEI-93-TR-24, Pittsburgh, PA, February 1993.

[Rumbaugh 1990] Rumbaugh, James et al. *Object-Oriented Modeling and Design*. Prentice-Hall, Englewood Cliffs, NJ, 1990.

[Shlaer 1992] Shlaer, S. and S. Mellor. *Object Lifecycles: Modeling the World in States*. Yourdon Press/Prentice-Hall, Englewood Cliffs, NJ, 1990.

[Singer 1993] Singer, Gilbert. An Eclectic Approach to Developing an O-O Methodology. *Object Magazine*, Vol. 3, No. 4, November-December 1993, pp. 37-41.

[SPC 1993] Software Productivity Consortium. *Using New Technologies: A Technology Transfer Guidebook.* SPC-92046-CMC, Version 02.00.08, Reston, VA, December 1993.

[Taylor 1990] Taylor, David A. *Object-Oriented Technology: A Manager's Guide*. Addison-Wesley, Reading, MA, 1990.

[Wirfs-Brock 1990] Wirfs-Brock, R., B. Wilkerson, and L. Wiener. *Designing Object-Oriented Software*. Prentice-Hall, Englewood Cliffs, NJ, 1990.

[Yourdon 1994] Yourdon, Edward. *Object-Oriented Systems Design*. Yourdon Press, Englewood Cliffs, NJ, 1994.

GLOSSARY

Abstraction Focusing on the essential, inherent properties of an entity and ignoring its accidental properties.

Attribute Characteristic or property of an object or class; represented by data which maintains some value (state information), for that object or class.

Class An object template. This template defines the methods and variables for a particular type of object. Also a set of similar objects.

Encapsulation Grouping both the data and operations that affect that data into a single object.

Framework A collection of class libraries, generics, design, scenario models, documentation, etc., that serves as a platform to build applications.

Information Hiding Making the internal data and methods inaccessible by separating the external aspects of an object from the internal (hidden) implementation details of the object.

Inheritance The creation of new class by extending an old class by adding new attributes and/or methods.

Message A request by one object for the services of another object.

Object A combination of state and a set of methods that explicitly embodies an abstraction characterized by the behavior of relevant requests. An object is an instance of an implementation and an interface. An object models a real-world entity (such as a person, place, thing, or concept), and it is implemented as a computational entity that encapsulates state and operations (internally implemented as data and methods) and responds to requestor services.

Object Request Broker A program that provides a location and implementation independent mechanism for passing a message from one object to another.

Operation/Method A specific behavior that an object exhibits; implemented as a procedure contained within the object.

Polymorphism Objects in different classes may understand the same message, yet respond differently.

LIST OF ACRONYMS

CDA	Dentral Design Activity
DISA	Defense Information Systems Agency
DoD	Department of Defense
ICOM	Input, Control, Output, Mechanism
IDA	Institute for Defense Analyses
IDEF	Integrated Computer Aided Manufacturing (ICAM) Definition Language
JOOP	Journal of Object-Oriented Programming
OOT	Object-Oriented Technology