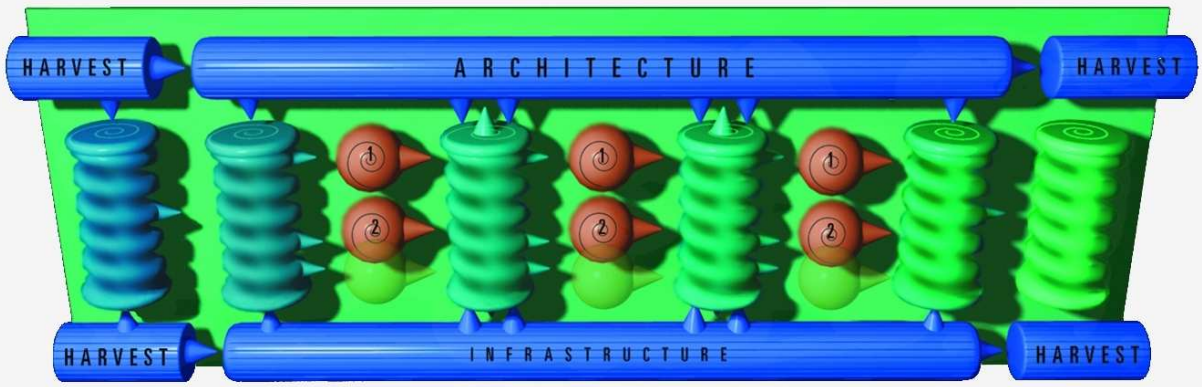
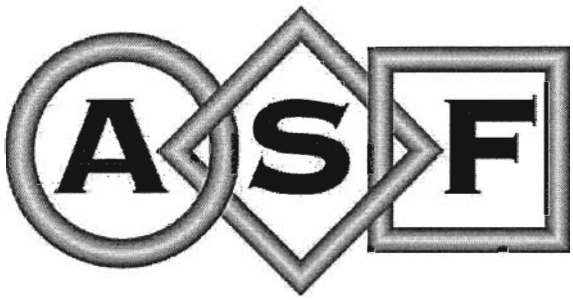
Process Management for Object-Oriented Development



ASF
AUSTIN SOFTWARE FOUNDRY



AUSTIN SOFTWARE FOUNDRY

Process Management for Object-Oriented Development

November 1997

Printed: November, 1997

AUSTIN SOFTWARE FOUNDRY
CAPITAL OF TEXAS HIGHWAY, NORTH
BUILDING 8, SUITE 250
AUSTIN, TEXAS 78746
(512) 329 - 6697 VOICE
(512) 329 - 6698 FAX
WWW.FOUNDRY.COM

Table of Contents

Unit 1: Introduction

Unit 2: Cultural and Conceptual Shifts

Unit 3: Moving to a Client/Server Process Model

Unit 4: Roles, Responsibilities, and Skill Requirements

Unit 5: Project Start-Up

Unit 6: Business & Requirements Analysis

Unit 7: Business Object Modeling

Unit 8: Clustering

Unit 9: Entering Timebox Development

Unit 10: Managing Design and Construction

Unit 11: Managing the Validation Process

Unit 12: Managing the Generalization Process

Unit 13: Integration and Review

Unit 14: Sucessful Client/Server Development

Appendix A: Glossary of Terms

Appendix B: Capability Maturity Model

Appendix C: Iterative Development Project Management Checklist

Appendix D: Example Object-Oriented Deliverables

Appendix E: Vendor Listing

Appendix F: Object-Oriented Metrics

Appendix G: A Prioritized Top-Ten List of Software Risk Items

Appendix H: Bibliography

Unit 1: Introduction

Objectives

Upon completion of this unit you should be able to:

- Explain what this course is about
- Understand how the course is organized
- Discuss the driving force behind the transition to object-oriented development
- Discuss differences and key challenges in managing an object-oriented development project

Topics

- Introduction
- Course Objectives
- Transition to Object-oriented

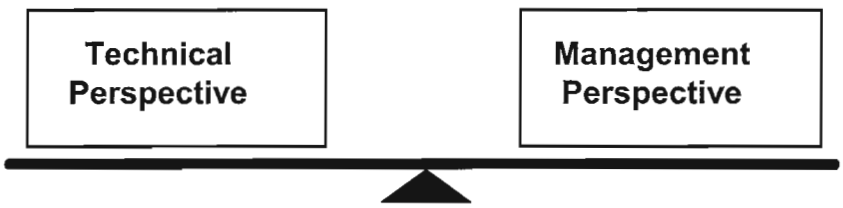
Introduction

It has been said that "the only constant in life is change." This statement is very appropriate for those professionals whose careers are tied to the development, management, and use of information technologies. We are in the midst of radical changes in both business and technology environments. The migration to client/server technologies has caused IS organizations and professionals to rethink their architectures, management practices, and most importantly, their people's skills. The purpose of this class is to explore these changes as they relate to managing object-oriented development.

Definition

Project management is the process of leading a team of people in planning and managing a number of related tasks that must be accomplished by a specific date. All projects have the following two major aspects:

- Technical perspective - how to develop a system. The technical perspective involves the underlying technologies, methodologies, process, and tools used to implement a system.
- Management perspective - how to control, steer, and follow up the project. The management perspective involves organizational culture and maturity, project teams, roles and responsibilities, and communication.



The technical and management aspects of a project must fit together in order for a system to be successfully implemented. Successful project management is about knowing how to balance the technology and management issues related to the project.

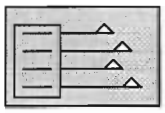
Course Objectives

Upon successful completion of the *Process Management for Object-Oriented Development* course, you should be able to:

- Understand the technical and management issues relevant to object-oriented development in order to effectively plan, organize, coach, communicate, and monitor object-oriented development projects.
- Effectively manage the iterative development process necessary for object-oriented development.
- Understand the processes and deliverables associated with the iterative development lifecycle.
- Understand the roles, responsibilities, and skill requirements necessary for object-oriented development.
- Define and discuss object-oriented terminology and interpret object-oriented models.

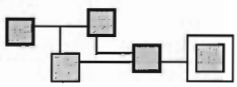
What You Already Know

Successful project management demands that the project manager and team members invest sufficient time up front in identifying all the tasks that need to be done, the sequence and schedule for the tasks, and the resources needed to see them through to completion. The manager and team members must continually juggle quality, schedule, and cost.



Gantt Charts

Project Plans



Task Networks

Work Breakdown Structures

Traditionally, during the systems development process, managers:

- **Develop Gantt charts** to graphically track the tasks to be accomplished, the sequence and time schedule for the tasks, and the resources assigned. Gantt charts showed horizontal bars representing the duration of work to be completed.
- **Develop project plans** and periodic status reports informing users, management, and team members about project progress.
- **Create task network structures** that display the sequence of project tasks on nodes and lines showing predecessor and successor tasks. Task network structures show a collection of paths beginning at a start nodes and converging at the end of the project.
- **Create work breakdown structures** showing hierarchies that displayed the levels of project work.

There is an abundance of literature about how to manage software development projects. This is, however, not the subject of this class. Instead, this class will concentrate on how to combine common project management practices with system engineering for object-oriented development. It will also address the implications object-oriented development has on project management and what project managers should consider when working with this technology.

What You Will Learn

From a **technical** perspective you will learn:

- A new process model that is better suited for object-oriented development. In learning a new process model, you will learn new tasks and deliverables that are associated with it. You will also learn about some of the major issues that must be addressed when developing object-oriented applications.
- How to manage an iterative development process. You will learn some techniques for controlling and defining multiple iterations for a project and how delivering applications in smaller increments can help you be successful.
- Guidelines for managing object-oriented applications including how to control the prototyping process and avoid some common prototyping pitfalls, how to develop a risk management plan, and how to evaluate purchased application frameworks.

From a **management** perspective you will learn:

- That the move to object-oriented development results in new roles, responsibilities, and skills. Analyzing project team members current skills portfolio and obtaining the appropriate mix of roles and skill levels are critical for successful object-oriented development.
- That the move to object-oriented development results in many cultural and conceptual changes that affect end-users, management, and project team members. You will learn what you need to do to help manage the adaptation to changes in the new environment.
- Organizational requirements necessary to develop components. Effective reuse of system components requires different organizational responsibilities than traditional development as well as organizational changes from a project to component culture.

Software Engineering Challenges

Project managers face major challenges in developing systems in today's environment. Systems are being developed in an environment which is changing dramatically:

- Business requirements are changing faster than ever; users are demanding more and wanting systems delivered quicker
- Information technology is undergoing the most radical and fundamental change in its brief history

In addition to the changing business climate, information systems are increasingly becoming strategic weapons to achieve a competitive advantage instead of tactical weapons used to achieve efficiencies. This puts systems on the "front line" of business activity and makes them subject to the rapidly changing business environment.

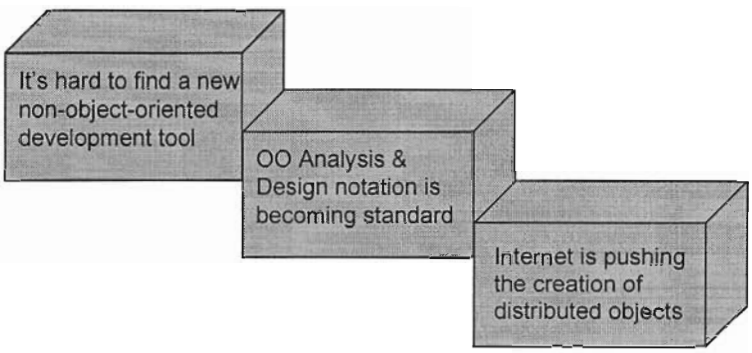
- Competitors respond to strategic systems initiatives with technology applications of their own

Object-Oriented Development Is Becoming Mainstream

Object technology is rapidly becoming a predominant development paradigm. Object-oriented programming is everywhere. Today, it is difficult to find a new tool that is not object-oriented. Object-oriented analysis and design is becoming a mainstream methodology. This is partly due to the standardization of the modeling notation (Unified Modeling Language developed by Rational Software Corp). In addition, the internet is creating a powerful surge of new applications that are in need of distributed objects.

What Are the Driving Forces for Object-Oriented Development?

As it certainly appears that object-oriented technology is here to stay, what are the major forces that are driving the transition to object-oriented development? It is certainly not the cost. Some studies have shown that current object-oriented technology is equal to or in some cases more expensive than traditional development. One of the biggest pay backs for object-oriented development is the way it enables better approaches to solving business problems, and thus supports "business-driven application development".



- **Business-driven application development** means working closely with users and customers to solve a business problem. Object-oriented technology fosters business-driven application development and facilitates a company's efforts to deploy business process reengineering.

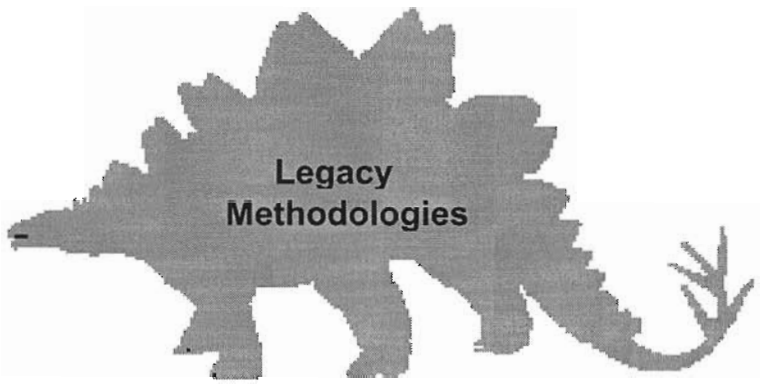
- Object-oriented development tools and techniques such as GUI tools, rapid application prototyping, and concurrent engineering have **accelerated development** time and allowed business solutions to be deployed much quicker than in a traditional environment.

- Organizations are also finding that object-oriented development environments offer **increased flexibility** in resulting applications. Object-oriented applications can be easily distributed. In addition, changes to parts of an application, such as swapping out a windows interface with a web-enabled interface are accomplished much cheaper and quicker than with traditional development.

The vision for object-oriented development is to provide enterprise-wide components that will allow applications to be built quickly by assembling components throughout the enterprise. As distributed architecture matures, object-oriented development will continue to drive the creation of components and component-based development. By implementing object-oriented development, businesses hope to eventually cut total costs, leverage their current investments, deliver valuable services to end-users and improve overall productivity.

Business-Driven Development

For more than 25 years, the mainframe has dictated the way central IS organizations have been run. Corporations have invested millions of dollars and hours developing mission-critical business applications spanning millions of lines of code. With investments as deep as these, few would expect to see a dramatic or quick change in the ways things are done in an IS organization.



But business-driven development requires a new way of developing applications. Legacy methodologies using 3GL development must be replaced with an event-driven, prototype intensive, iterative development environment using high-level modeling and objects to capture and enable the essence of a business transaction. These changes will require developers to embrace new development processes, methodologies, tools and standards.

Modern Object-Oriented Development Process

A modern object-oriented development process recognizes the essential differences between object-oriented and traditional development, as well as the diversities in skill sets needed for developers. A modern object-oriented development process should involve the following:

- **Rapid Application Development (RAD) techniques** - intensive prototyping sessions with end-users to gather system and data requirements, business rules, and user-interface requirements.
- **Iterative development and concurrent engineering techniques** - development of multiple subsystems simultaneously with multiple iterations
- **Integrated object models** - development of not only the entities and their attributes, but their methods and how they interact with one another.
- **Reusability from analysis through testing** - reuse of components and frameworks achieved through proper application partitioning and layering.
- **Sessions to address specific client/server database issues** - determination of where data should be accessed and stored across the network and the use of data replication, partitioning, and caching.
- **Customizable methodology support using process management tools** - Although a standards organization or quality-control group should control methodology customizations, the rapid rate at which products and technology changes occur is demanding a customizable methodology.

What's Different from Traditional Development?

Some may question why project management should be any different for object-oriented development. It is true at the highest level of thinking, a project is a project, and the development approach should be irrelevant. In actual practice a different management approach is needed to successfully manage object-oriented development for the following reasons:

- **Tasks and skills.** Tasks associated with identifying, characterizing, and documenting objects are new. Although the basic skills required of the software engineer are the same regardless of the development approach used, the application of these skills and the techniques employed must change with the change in development approach. This change is often referred to as a "paradigm shift."

- **Roles and responsibilities.** There are new roles such as object-oriented analysis and design specialists and reuse administrators. Other roles have changed, such as the role of the traditional programmer. The need for the traditional "backroom" coder has been replaced with a multifaceted object-designer, facilitator, and prototyper who must work closely with the end-users to capture their vision.

- **Lifecycles.** The approach to creating applications has changed. It is no longer a sequential approach using 3GL tools. Technology has changed to an event-driven, prototype intensive, iterative development environment using high-level modeling and objects to capture and enable the essence of a business transaction.

- **Deliverables.** Some deliverables associated with traditional methodologies are no longer necessary. Deliverables such as data flow diagrams and structure charts have been replaced with more modern deliverables such as object models and interactive prototypes.

- **New issues.** There are many new issues that must be addressed for object-oriented development. Some of the key issues are application design and distribution in a client/server environment, implementing change control using reusable components, and controlling iterative development.

Many of the skills that you have to effectively manage projects today are still valid, but they need to be adjusted to effectively manage projects in a object-oriented environment.

Object-Oriented Project Management Challenges

If systems development projects were easily implemented, there would be little need for detailed project planning and control. Anyone could be a successful project manager. However, anyone who has worked on a major development project knows that projects can become complex and difficult to plan and manage for a variety of reasons, including the following:

- Defining and refining the scope of a project is often an ongoing process that continues until the project is complete.
- Newly formed project teams, committees, and task forces are often made up of many people who have never worked together before.
- The needs of many clients or users must be coordinated simultaneously. Managing user expectations is critical to project success.
- Dealing with conformance to organizational policies, procedures, and standards may add hurdles or red tape.
- Organizational boundaries may add operational and political problems.

The challenges above exist for any development environment. They do not disappear with object-oriented development. However, object-oriented development adds new challenges for project managers to consider such as:

- Managing the impact of cultural changes which affect management, users, and IS staff.
- Controlling the iterative development cycle.
- Using new technologies that may have unforeseen results.
- Managing end-users' expectations as they become more proficient with the use of PC's and PC tools.

These are just a few of the many new challenges that a project manager must face in managing object-oriented development. Throughout this course we will discuss techniques and guidelines to help you manage these challenges.

Summary Questions

1. What is the biggest driving force for the transition to object-oriented development?

2. What kinds of things should a modern object-oriented methodology involve?

3. Why should project management be any different for object-oriented development than for traditional development?