



University of Southern California
Center for Software Engineering

The MBASE* Life Cycle Architecture Package -and Relations to the Draft DoD 5000 Series

**Barry Boehm, USC
GSAW 2000 Keynote
February 23, 2000**

**boehm@sunset.usc.edu
<http://sunset.usc.edu/MBASE>**

***Model Based (System) Architecting and Software Engineering**



Outline

- **Draft DoD 5000 Series Highlights**
- **Evolutionary Acquisition Critical Success Factors**
- **Relation to MBASE Life Cycle Architecture Package**
- **Summary; Relation of MBASE LCA Package to Draft DoD 5000 Series**



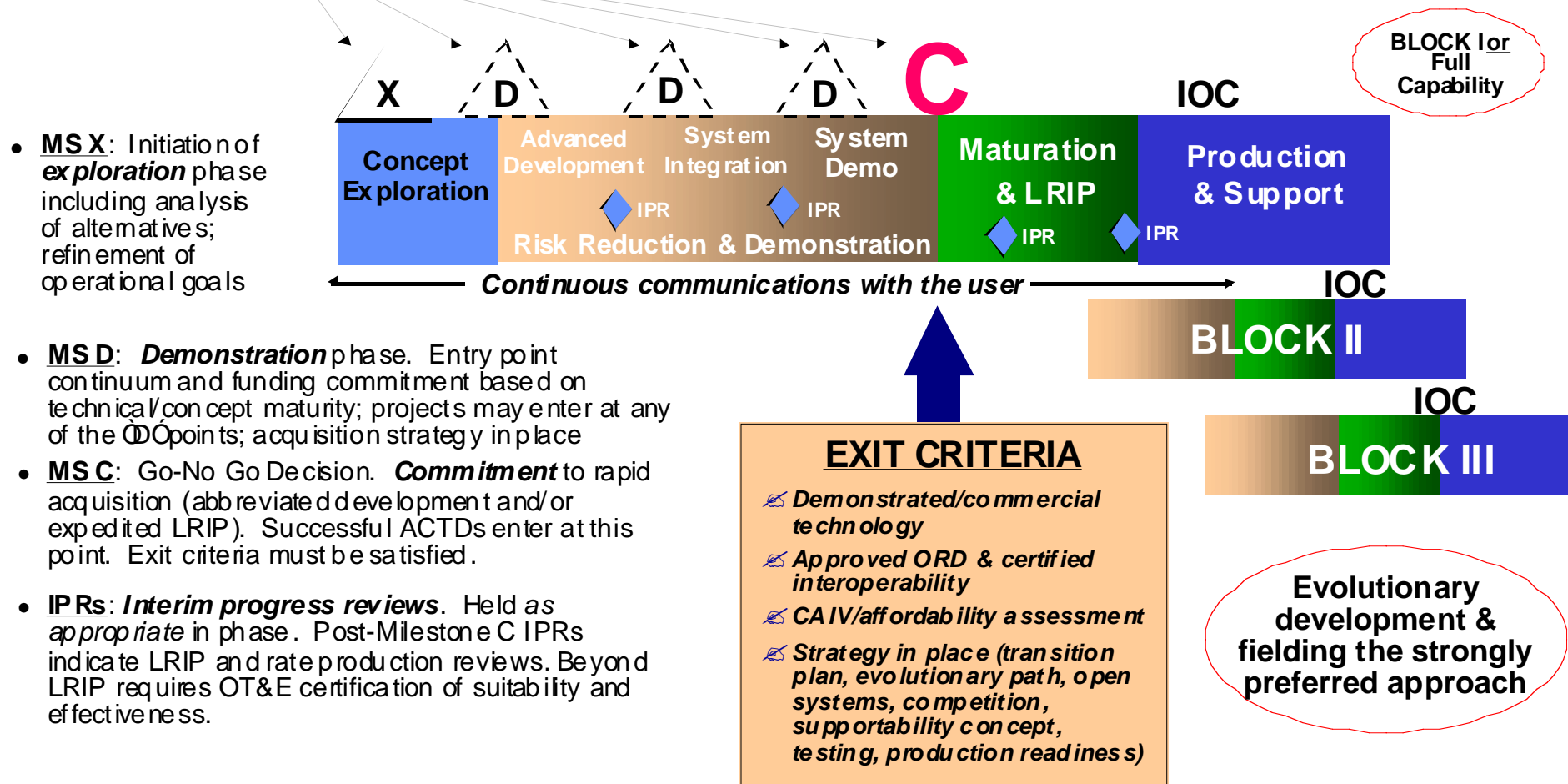
Draft DoD 5000 Series Highlights

- **Rapid transition from technology to products**
 - Time- phased requirements
 - Use of commercial products and technologies
- **Rapid transition from acquisition to fielding**
 - Evolutionary acquisition (EA)
 - Integrated test and evaluation
 - Interoperability
 - Competition
 - Commitment Criteria
 - Mature technology, well-understood requirements, strong EA plan covering factors above
- **Integrated operational support**
- **Guidelines in DoDI 5000.2, DoD 5000.2-R**
 - New decision milestones

A MODIFIED ACQUISITION PROCESS

Technology Opportunities & User Challenges

- **New milestone names** to emphasize new approach
- **Multiple entry points** possible depending on technical/concept maturity
- **Three basic options at each decision point:** Proceed into next phase; do additional work; terminate effort





Some EA Critical Success Factors

- **Assurance that requirements are achievable**
 - Via architecture-based analysis and demonstration
- **Assurance that architecture will scale to life-cycle**
 - Need to specify evolution requirements
- **Assurance that cost, schedule targets can be met**
 - CAIV, SAIV, and architecture implications
- **Replacements for waterfall contract milestones**
 - Clear milestone pass/fail criteria



Schedule As Independent Variable (SAIV)

- **Build a core capability early**
- **Drop additional features as necessary to meet schedule**
 - **Need prioritized features (“requirements”).**
 - **Need an architecture enabling easily-dropped features.**
 - **Need good estimates, planning and control.**



Outline

- **Draft DoD 5000 Series Highlights**
- **Evolutionary Acquisition Critical Success Factors**
- ➔ • **Relation to MBASE Life Cycle Architecture Package**
- **Summary; Relation of MBASE LCA Package to Draft DoD 5000 Series**



Life Cycle Anchor Points

- **Common System/Software stakeholder commitment points**
 - Defined in concert with Government, industry affiliates
 - Coordinated with Rational's Unified Software Development Process
- **Life Cycle Objectives (LCO)**
 - Stakeholders' commitment to support system architecting
 - Like getting engaged
- **Life Cycle Architecture (LCA)**
 - Stakeholders' commitment to support full life cycle
 - Like getting married
- **Initial Operational Capability (IOC)**
 - Stakeholders' commitment to support operations
 - Like having your first child



Win Win Spiral Anchor Points

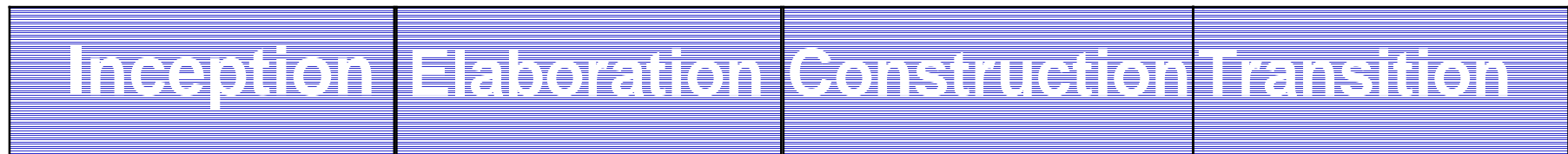
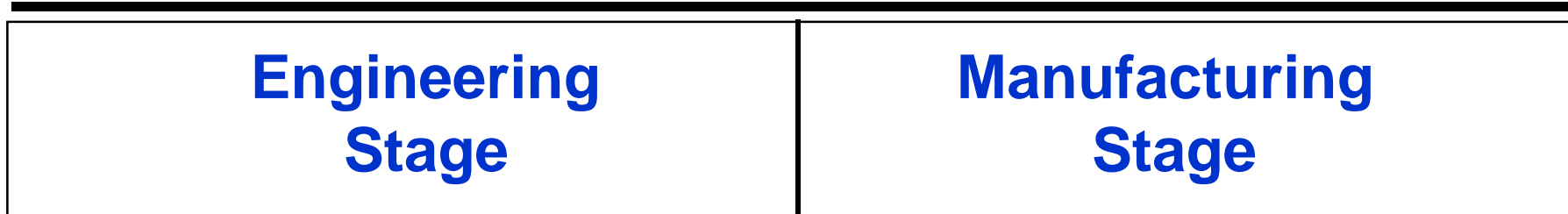
(Risk-driven level of detail for each element)

Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
Definition of Operational Concept	<ul style="list-style-type: none"> • Top-level system objectives and scope <ul style="list-style-type: none"> - System boundary - Environment parameters and assumptions - Evolution parameters • Operational concept <ul style="list-style-type: none"> - Operations and maintenance scenarios and parameters - Organizational life-cycle responsibilities (stakeholders) 	<ul style="list-style-type: none"> • Elaboration of system objectives and scope of increment • Elaboration of operational concept by increment
System Prototype(s)	<ul style="list-style-type: none"> • Exercise key usage scenarios • Resolve critical risks 	<ul style="list-style-type: none"> • Exercise range of usage scenarios • Resolve major outstanding risks
Definition of System Requirements	<ul style="list-style-type: none"> • Top-level functions, interfaces, quality attribute levels, including: <ul style="list-style-type: none"> - Growth vectors and priorities - Prototypes • Stakeholders' concurrence on essentials 	<ul style="list-style-type: none"> • Elaboration of functions, interfaces, quality attributes, and prototypes by increment <ul style="list-style-type: none"> - Identification of TBD's(to-be-determined items) • Stakeholders' concurrence on their priority concerns
Definition of System and Software Architecture	<ul style="list-style-type: none"> • Top-level definition of at least one feasible architecture <ul style="list-style-type: none"> - Physical and logical elements and relationships - Choices of COTS and reusable software elements • Identification of infeasible architecture options 	<ul style="list-style-type: none"> • Choice of architecture and elaboration by increment <ul style="list-style-type: none"> - Physical and logical components, connectors, configurations, constraints - COTS, reuse choices - Domain-architecture and architectural style choices • Architecture evolution parameters
Definition of Life-Cycle Plan	<ul style="list-style-type: none"> • Identification of life-cycle stakeholders <ul style="list-style-type: none"> - Users, customers, developers, maintainers, interoperators, general public, others • Identification of life-cycle process model <ul style="list-style-type: none"> - Top-level stages, increments • Top-level WWWWWHH* by stage 	<ul style="list-style-type: none"> • Elaboration of WWWWWHH* for Initial Operational Capability (IOC) <ul style="list-style-type: none"> - Partial elaboration, identification of key TBD's for later increments
Feasibility Rationale	<ul style="list-style-type: none"> • Assurance of consistency among elements above <ul style="list-style-type: none"> - via analysis, measurement, prototyping, simulation, etc. - Business case analysis for requirements, feasible architectures 	<ul style="list-style-type: none"> • Assurance of consistency among elements above • All major risks resolved or covered by risk management plan

*WWWWWHH: Why, What, When, Who, Where, How, How Much



Anchor Points and Rational Process Phases



LCO

LCA

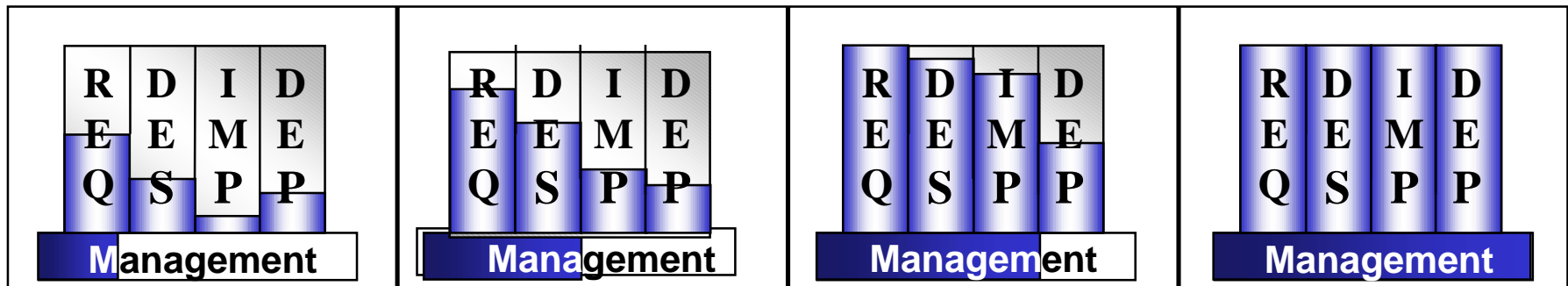
IOC

Feasibility Iterations

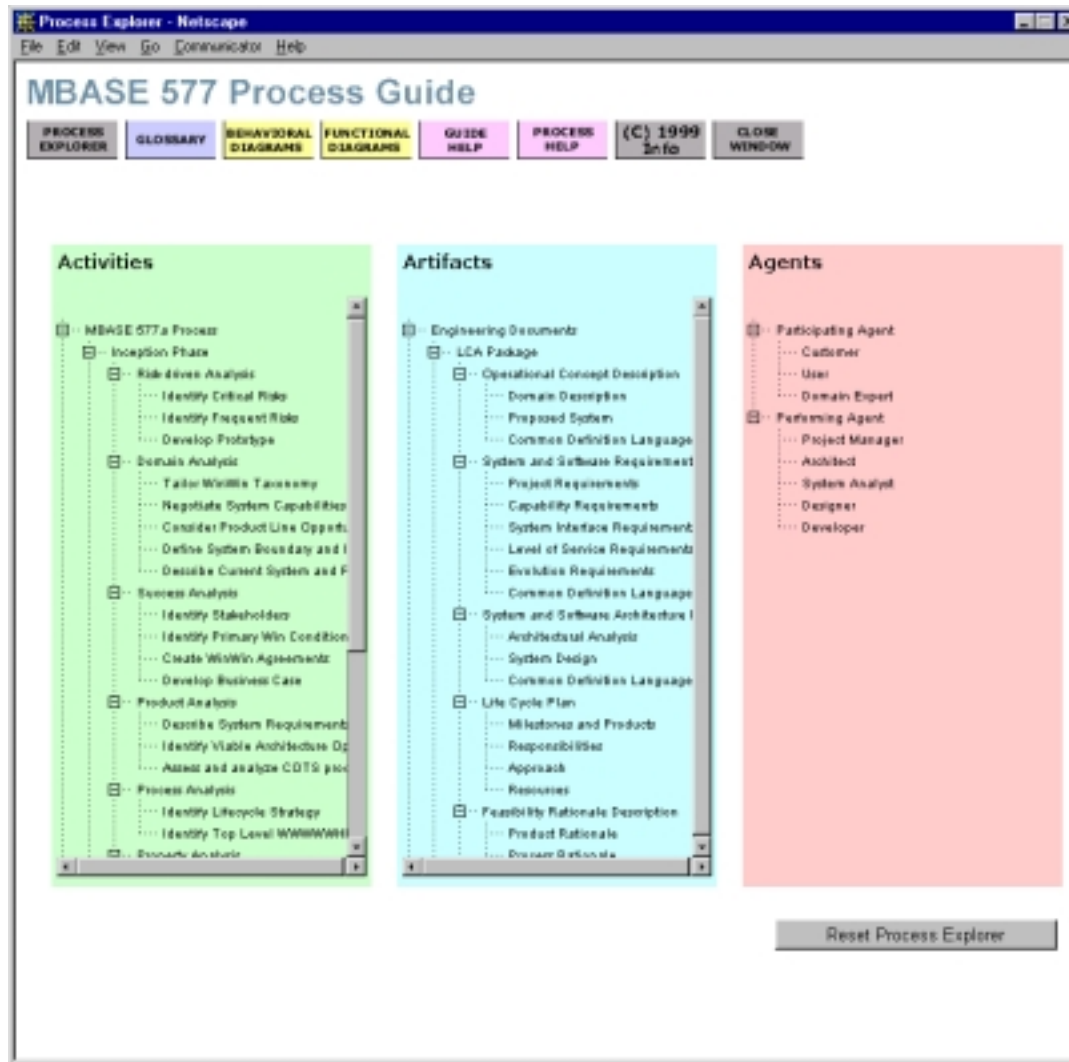
Architecture Iterations

Usable Iterations

Product Releases



MBASE Electronic Process Guide (1)



- Developed in collaboration with CMU-SEI
- Using SEI EPG tool



MBASE Electronic Process Guide (2)

Activities:

- MBASE 577a Process
 - Inception Phase
 - Elaboration Phase
 - Record Project Effort
 - Methods to MBASE

Activity: MBASE 577a Process

- Overview
- Purpose
- Decomposition
- Description
- Tools and Techniques
- Pitfalls
- Participation
- Participants
- Interacts
- Behavior
- Effort Guidelines
- Recommended Prerequisites
- Applicable Policies and Standards
- Concepts and References

MBASE 577a Process

Overview

The process followed by students of CS 577a for Digital Library projects

Purpose

The objectives for the MBASE 577a process are:

- To define a life cycle process for the students of CS 577a for use in Digital Library and similar projects
- To provide guidance to project evaluators about the inter-dependence of MBASE process elements

Decomposition

The activity MBASE 577a Process is decomposed into the following:

- Inception Phase
- Elaboration Phase
- Record Project Effort

Description

Model-Based (System) Architecting and Software Engineering (MBASE) is an approach for developing software intensive systems. The MBASE approach integrates the four common development models (success, product, process and property) around the creation and use of a software architecture package. Using MBASE, model changes can be recognized and reconciled as a matter of fact rather than after the fact.

The MBASE 577 process is a variant of the MBASE framework used by students in the course CS 577 offered at USC. The process consists of four distinct phases: Inception, Elaboration, Construction and Transition. Each phase is completed with a commitment from the stakeholders during a review.

The tasks to be performed during the MBASE 577 process are:

- Identifying and resolving the critical risks
- Analyzing the problem domain
- Identifying the feasible architecture
- Developing a prototype to test the validity of the architecture and synthesizes of stakeholders' concerns
- Understanding the requirements of the life cycle and obtaining concurrence from the system stakeholders

Tools and Techniques

Electronic Process Guide for MBASE

Guidelines for Model-Based Architecting and Software Engineering (MBASE) deliverables: Inception and Elaboration

Pitfalls

Artifacts:

- Using Documents
- Package
 - Operational Concept Description
 - Domain Description
 - Proposed System
 - Common Definition Language for System and Software Requirements Description and Software Architecture Description
 - Life Cycle Plan
 - Feasibility/Rationale Description
 - Prototype
 - WinWin Negotiation Report
 - System and Software Architecture Description
 - System and Software Requirements Description
 - Work Effort Report
 - Use Patterns

Artifact: Operational Concept Description

- Overview
- Purpose
- Owner
- Decomposition
- Description

Operational Concept Description

Overview

Provides the overall context of the proposed system and its operational concept

Purpose

- Describe the overall context of the system to be developed, why it's being built, what exists now, and where the project is starting from
- Describe to the stakeholders of the system to be developed ('developed' is meant to include such terms as 'enhanced', 'updated', 're-engineered', 'automated'), how the system will work in practice once it is deployed
- Enable the operational stakeholders to evolve knowledgeably from their current operational concept to the new operational concept, and to collaboratively adapt the operational concept as developments arise, to make clear the value of developing the new system

Owner

The artifact Operational Concept Description is owned by the agent System Analyst.

Decomposition

The artifact Operational Concept Description is decomposed into the following:

- Domain Description
- Proposed System
- Common Definition Language for Domain

Description

OCD Audience and Participants

- Audience
- Customer for Domain Description Domain
- Domain Expert for System Analysis
- Use language and define CDL appropriately for intended audience
- Participants
- Same stakeholders as WinWin negotiation
- Establish concept of operation agreed on by all stakeholders

OCD High-Level Dependencies

- WinWin Negotiations Give
- System Responsibilities
- Changes Considered But Not Included
- Domain Description Terms
- Project Goals, Quality Goals
- OCD YH&S
- Project System and Quality Req for SS&D
- Domain Description and Initial Analysis for SS&D
- Stakeholder and Organizational Responsibilities



MBASE LCO and LCA Pass/Fail Criteria

- **LCO and LCA Feasibility Rationale:**
 - **If we build a system to the given architecture, it will**
 - **Satisfy the requirements (including evolution)**
 - **Be faithful to the prototype**
 - **Support the operational concept**
 - **Be buildable within the budget and schedule in the plan**
 - **Satisfy a viable business case for the investment**
- **LCO: Satisfies criteria for at least one choice of architecture and COTS**
- **LCA: Satisfies criteria for specific detailed choice of architecture and COTS**
 - **All major risks resolved or covered by risk management plan**



Summary: Relation of MBASE LCA Package to Draft DoD 5000 Series

- **MBASE LCO content and pass/fail criteria fit new DoD-5000 milestone D**
- **MBASE LCA content and pass/fail criteria fit new DoD-5000 milestone C**
- **LCO and LCA milestones satisfy EA critical success factors**
 - **Assurance that requirements are achievable**
 - **Assurance that architecture will scale to life-cycle needs**
 - **Assurance that cost, schedule targets can be met**
 - **Clear milestone pass/fail criteria**