BAE SYSTEMS

Software Architecture in an Integrated Engineering Methodology

J.D. Baker
Systems Engineer, BAE Systems and
Member, OMG Architecture Board

BAE SYSTEMS

Abstract and Bio

J.D. Baker is a member of the Object Management Group Architecture Board, where he represents BAE Systems. Within the OMG, he has participated in the development of UML, OMG SysML, and the UML Profile for DoDAF and MODAF. At BAE Systems he is the lead Software System Engineer/Architect for the Integrated Engineering Methodology, a model-based methodology for the design and construction of complex, software-intensive systems. J.D. holds many industry certifications, including OMG Certified UML Professional, Sun Certified Java Programmer, and he holds certificates as an SEI Software Architecture Professional and ATAM Evaluator.

Fitting software architecture into the engineering process becomes a challenge when you are developing complex systems. What are the inputs, where do they come from, how do I know that what the other disciplines are creating will meet my needs, how do I know I'm creating useful work products and they are being produced at the right time? Recognizing this complexity, BAE Systems has developed the Integrated Engineering Methodology (IEM), a model-based, end-to-end methodology that seeks to ensure that only the products that are needed are developed and that development occurs at the right time. How do you do all that and maintain the organization at CMMI Level 5? This paper describes the IEM, highlights the software architecture and describes its relationship to the other elements of the methodology.

BAE SYSTEMS

Background

The Approach and Motivation for Pursuing an Integrated Engineering Methodology

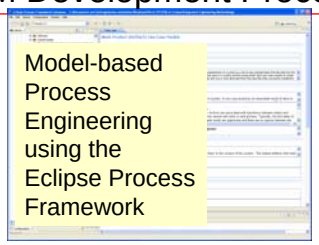
BAE SYSTEMS

IEM Development Approach

- A Model-based (UML and M&S) methodology
  - Supports INCOSE MBSE and OMG MDA
- Practical
  - Formalization of existing best practices from successful projects, not the invention of something new
  - Inputs from multiple business units
- Highly integrated
  - inputs and outputs span all of the disciplines
- Flexible and scalable
  - Ability to publish multiple configurations to support process agility
    - e.g. R&D process has been incorporated
  - Supports current Process Selection Tool
- Standards-based
  - Meets customer desires for development using open standards

BAE SYSTEMS

IEM Development Process

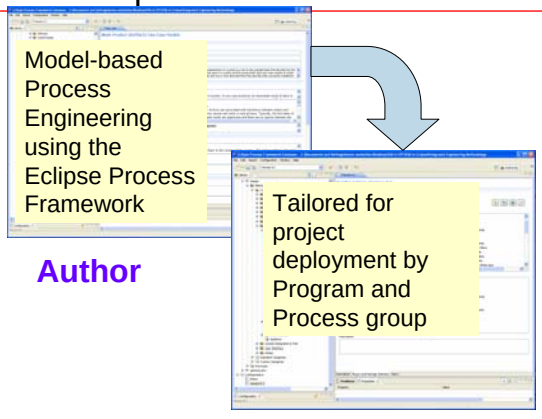


Author

- EPF is an open-source tool
  - IBM Rational Method Composer (RMC) without the RUP content and without the license cost
  - Implements the Software and Systems Process Metamodeling (SPEM) 2.0 standard
  - Authors fill in standard templates with content
  - Authors and process modelers describe the integration that result in links and references in the published web site

BAE SYSTEMS

IEM Development Process



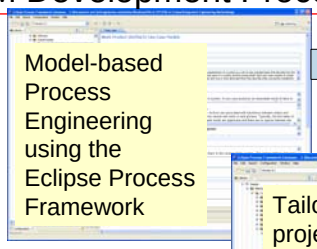
Author

Configure

- EPF can maintain multiple configuration views
  - Standard views
  - Tailored views for projects
  - Content is consistent for task and work product descriptions for all users
  - Consistent work products can be counted and measured meaningfully

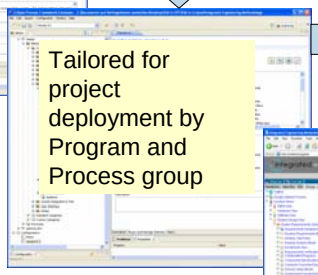
BAE SYSTEMS

IEM Development Process



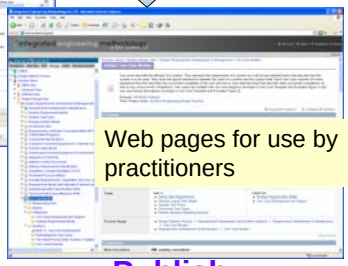
Model-based Process Engineering using the Eclipse Process Framework

Author



Tailored for project deployment by Program and Process group

Configure



- Users just need a browser to access the IEM content

Web pages for use by practitioners

Publish

BAE SYSTEMS

Notes

Key elements in the modeling of an engineering methodology

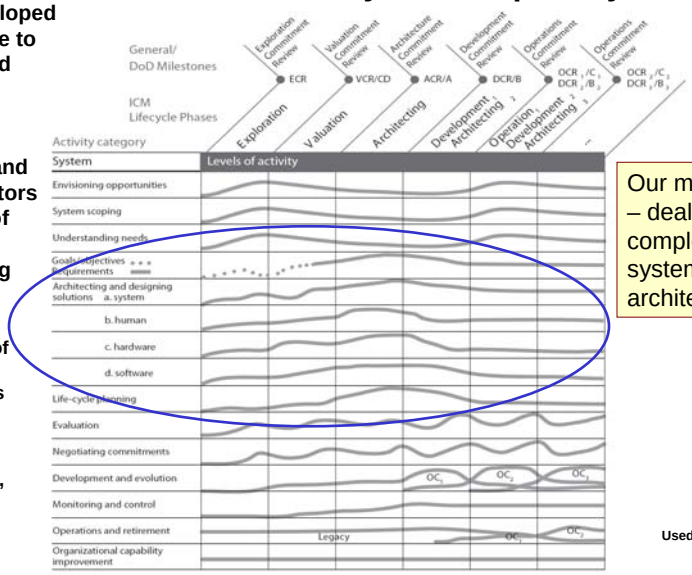
1. Standards-based notation/modeling language highly desirable
    1. The Eclipse Process Framework is based on the Software and System Engineering Meta-model v2.0
2. Commonly used tool so content can be reused
    1. EPF is being used to model the ICM
    2. EPF is used by Telelogic to model the Harmony SE and SW processes
    3. EPF is used by John McGregor (Clemson and SEI) to model Software Product Line related processes
    4. EPF is used by ICONIX Software to model ICONIX process
    5. EPF is used to model an agile enterprise architecture process - http://www.agileea.com/
3. Tailorable publication
    1. Projects tailor the IEM to their needs.
    2. Work products developed are consistent across variable projects to support systems and software estimating
4. Easy to use
    1. EPF publishes to HTMl
    2. Publication to a wiki coming soon

ICM HSI Levels of Activity for Complex Systems

ICM - Developed in response to DoD-related issues
Integrates hardware, software, and human factors elements of systems engineering

Concurrent exploration of needs and opportunities

Concurrent engineering of hardware, software, human aspects



Our motivation – dealing with complex system architecture

Used with permission

Notes

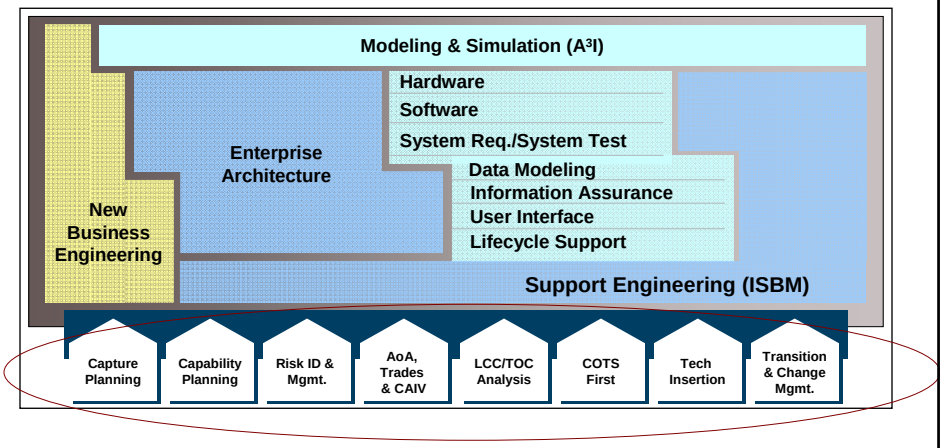
ICM HSI Levels of Activity for Complex Systems

As mentioned earlier, with the ICM, a number of system aspects are being concurrently engineered at an increasing level of understanding, definition, and development. The most significant of these aspects are shown in this slide, an extension of a similar view of concurrently engineered software projects developed as part of the RUP (shown in a backup slide).

As with the RUP version, it should be emphasized that the magnitude and shape of the levels of effort will be risk-driven and likely to vary from project to project. In particular, they are likely to have mini risk/opportunity-driven peaks and valleys, rather than the smooth curves shown for simplicity in this slide. The main intent of this view is to emphasize the necessary concurrency of the primary success-critical activities shown as rows. Thus, in interpreting the Exploration column, although system scoping is the primary objective of the Exploration phase, doing it well involves a considerable amount of activity in understanding needs, envisioning opportunities, identifying and reconciling stakeholder goals and objectives, architecting solutions, life cycle planning, evaluation of alternatives, and negotiation of stakeholder commitments.

Used with permission

IEM Structure

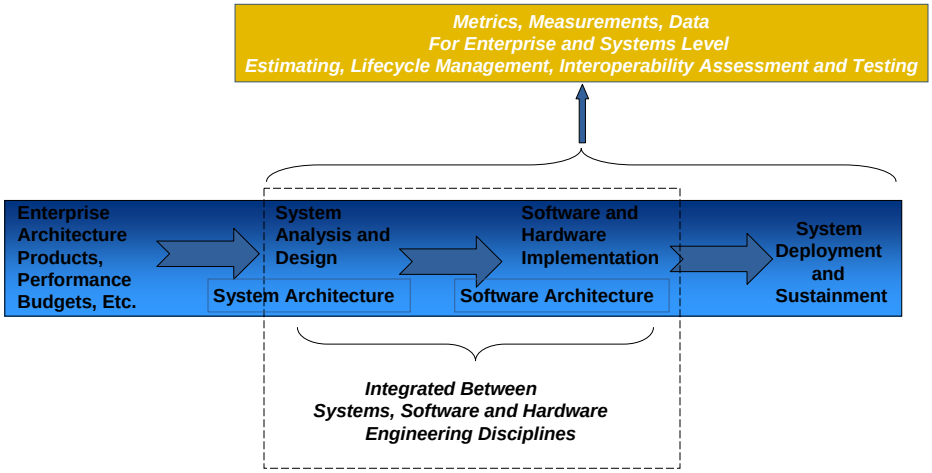


Supporting functions are applied as necessary throughout the system lifecycle.

Integrated Workflows

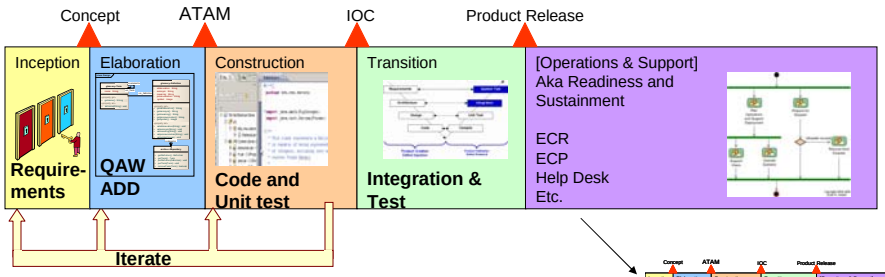
BAE SYSTEMS

Integration from the Systems Perspective

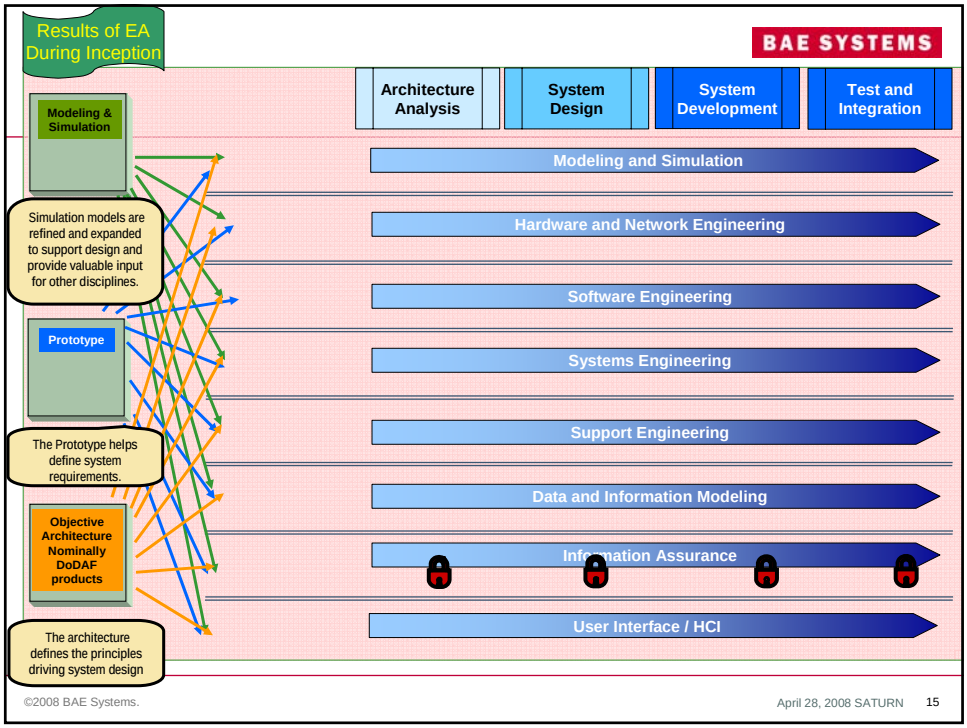


BAE SYSTEMS

Software Lifecycle



The SW workflow is repeated multiple times for long-lived systems. This workflow needs to fit into the system development workflow.

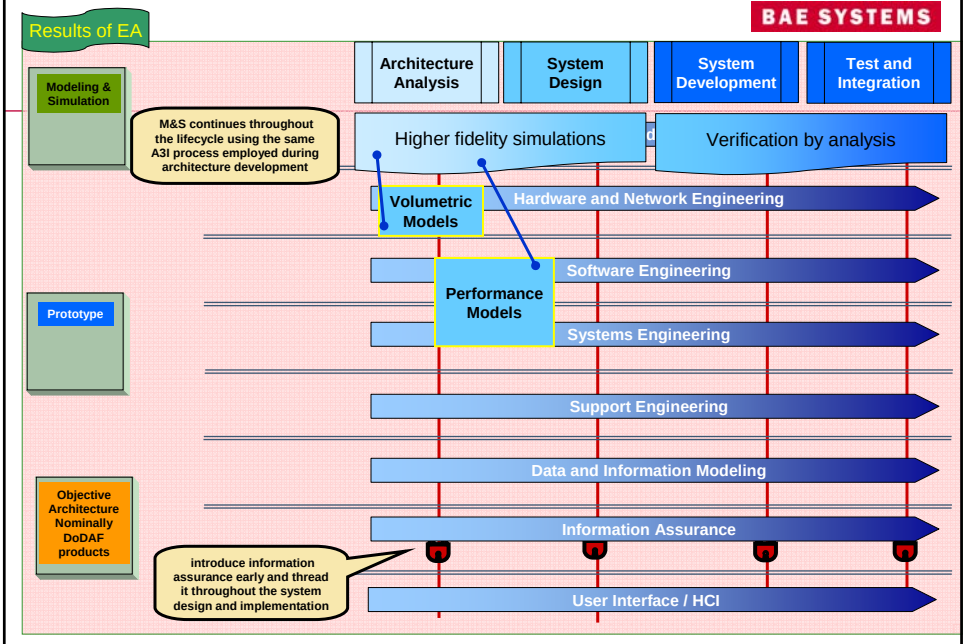


BAE SYSTEMS

Notes

Keys to developing workflows and lifecycle descriptions:

1. All development is iterative in the small
2. All development is linear in the large
3. All of that linearity and iterative development can not be reasonably represented in a single graphic

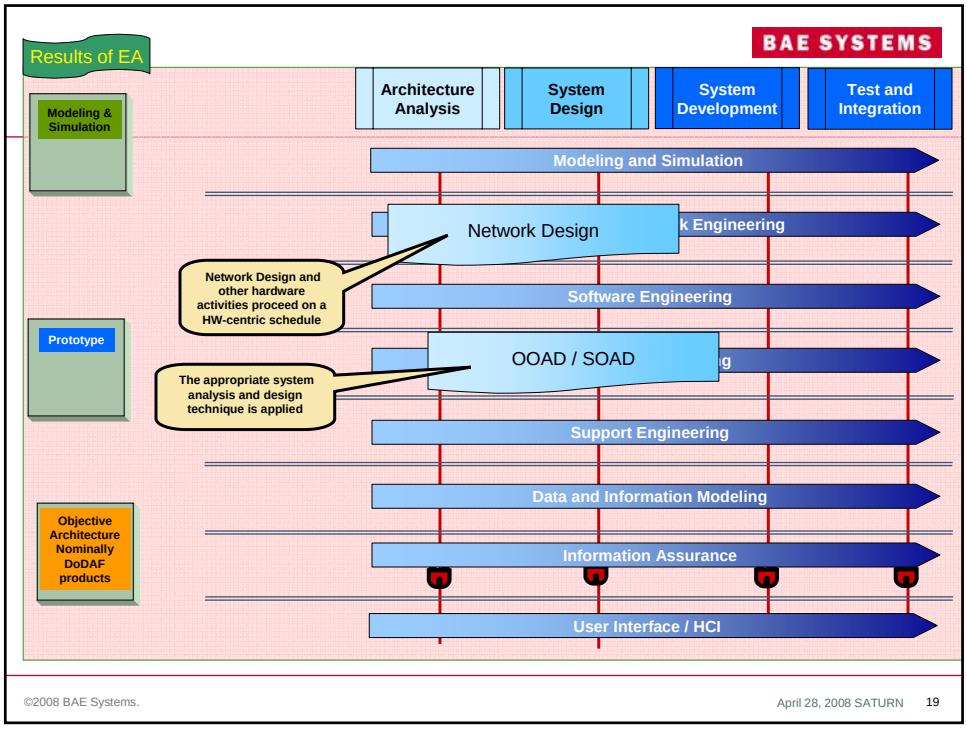


BAE SYSTEMS

Notes

Keys to integrating Modeling and Simulation:

1. In the capabilities and objectives definition stage of develop, enterprise architects and systems engineers develop scenario-based products (e.g. use cases, QAW scenarios, SV-10C)
2. M&S uses that information to understand the behaviors they need to simulate
3. M&S is data-driven. Software and Systems architects have to be conscious of the fact that they need to provide that data, in addition to validating the simulations.

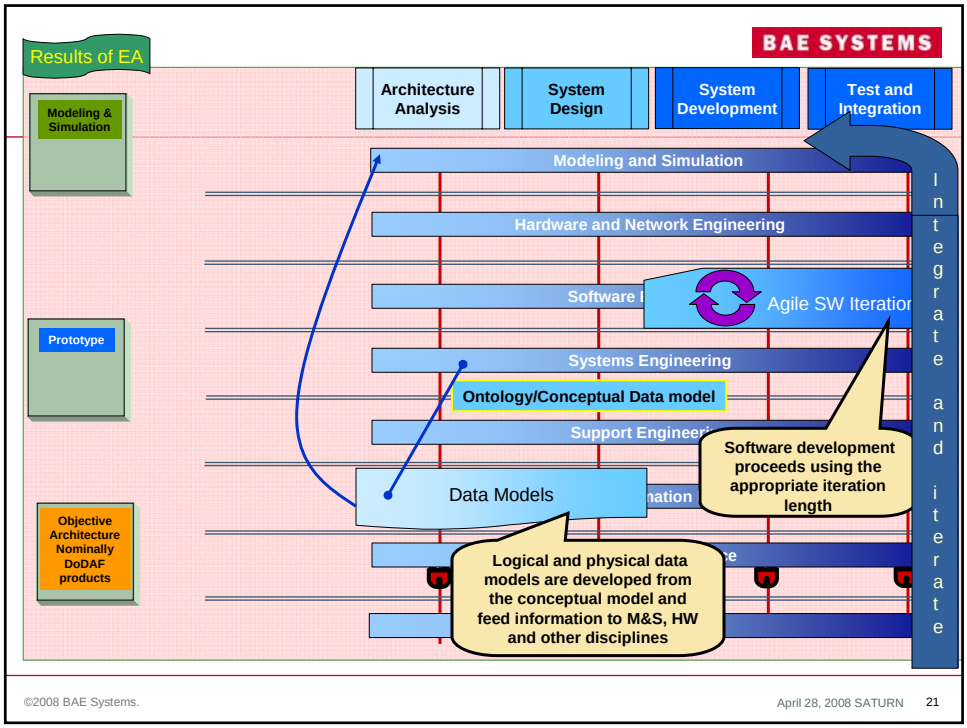


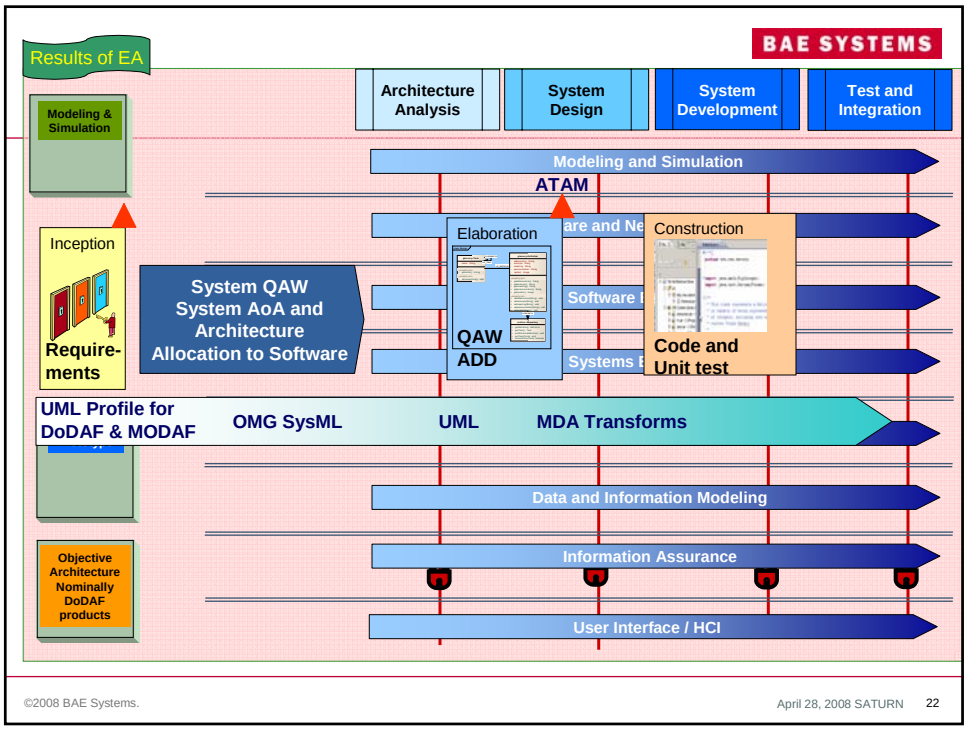
BAE SYSTEMS

Notes

Keys to aligning software and hardware lifecycles

1. Hardware often has long lead times for critical items
2. Software must carefully select alternate resources to support early analysis and prototyping


BAE SYSTEMS
Results of EA
Architecture Analysis
System Design
System Development
Test and Integration
Modeling & Simulation
Modeling and Simulation
Hardware and Network Engineering
Software
Agile SW Iteration
Prototype
Systems Engineering
Ontology/Conceptual Data model
Support Engineering
Data Models
Integrate and iterate
Software development proceeds using the appropriate iteration length
Objective Architecture Nominally DoDAF products
Logical and physical data models are developed from the conceptual model and feed information to M&S, HW and other disciplines
©2008 BAE Systems.
April 28, 2008 SATURN
21


BAE SYSTEMS
Results of EA
Architecture Analysis
System Design
System Development
Test and Integration
Modeling & Simulation
Modeling and Simulation
ATAM
Inception
Elaboration
Construction
Require-ments
System QAW
System AoA and Architecture Allocation to Software
QAW
ADD
Code and Unit test
UML Profile for DoDAF & MODAF
OMG SysML
UML
MDA Transforms
Data and Information Modeling
Information Assurance
User Interface / HCI
Objective Architecture Nominally DoDAF products
©2008 BAE Systems.
April 28, 2008 SATURN
22

BAE SYSTEMS

Notes

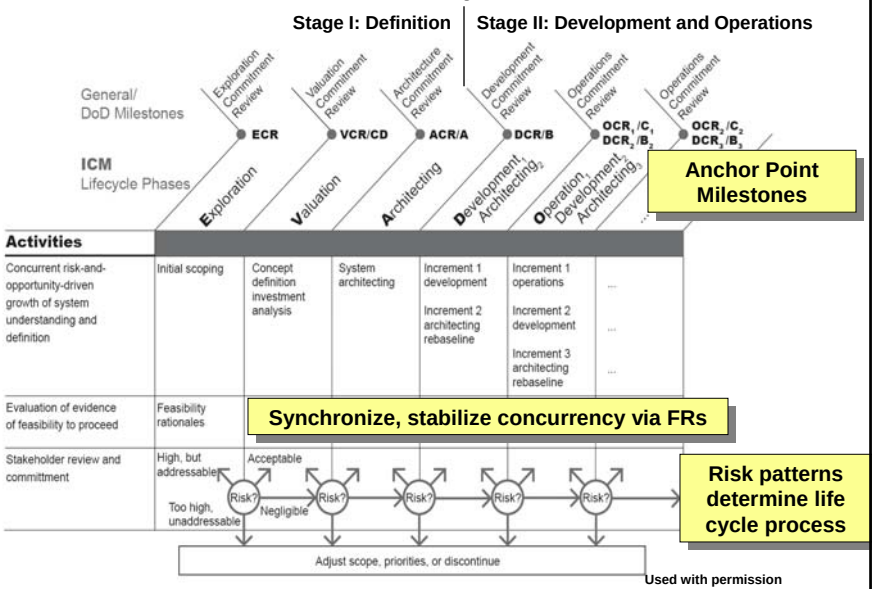
Keys to developing systems that appropriately (not blindly) implement the architecture

1. Understand the work products being developed to describe the architectural views well enough that everything has a purpose
2. Ensure those work products relate to one another
3. Work products should be in a UML-based model to the greatest extent possible.
4. For complex systems consider two QAWs as a risk reduction technique. One for the system architecture and one for the software architecture.

BAE SYSTEMS

Back-up Slides

The Incremental Commitment Life Cycle Process: Overview



Used with permission

Notes

The Incremental Commitment Life Cycle Process: Overview

This slide shows how the ICM spans the life cycle process from concept exploration to operations. Each phase culminates with an anchor point milestone review. At each anchor point, there are 4 options, based on the assessed risk of the proposed system. Some options involve go-backs. These options result in many possible process paths. The life cycle is divided into two stages: Stage I of the ICM (Definition) has 3 decision nodes with 4 options/node, culminating with incremental development in Stage II (Development and Operations). Stage II has an additional 2 decision nodes, again with 4 options/node.

One can use ICM risk patterns to generate frequently-used processes with confidence that they fit the situation. Initial risk patterns can generally be determined in the Exploration phase. One then proceeds with development as a proposed plan with risk-based evidence at VCR milestone, adjusting in later phases as necessary.

Risks associated with the system drive the life cycle process. Information about the risk(s) (feasibility assessments) supports the decision to proceed, adjust scope or priorities, or cancel the program.

Used with permission