The Engineering of Complex Enterprise Systems

Hal Sorenson

Professor and Faculty Director

Architecture-based Enterprise Systems Engineering (AESE) Leadership Program

Jacobs School of Engineering

Rady School of Management

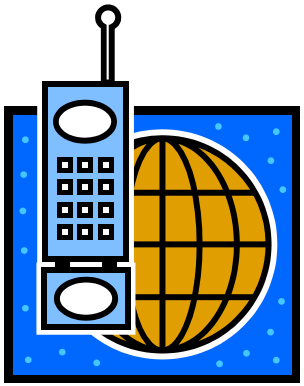
May 18, 2011

How Are Complex Enterprise Systems Engineered?

- Experiences that shaped the proposed methodology
- The enterprise systems engineering problem
- The basic architectural approach
- Top-down considerations
- The Architecture-based Enterprise Systems Engineering (AESE) Leadership Program

The Global Infosphere

- **The WWW has made the world *"flat"***
- **How can *enterprises* exploit the *"flatness"* to their greatest *advantage*?**



The General Context

- Business & National Security Organizations Involve
  - Many diverse stakeholders with differing cultures and responsibilities

This is an environment that abounds with complexity

  - Large number of autonomous or stove-piped systems
  - Inconsistent data/information models and data bases
  - …..
- Business & National Security Organizations Need
  - Cross-domain interoperation
  - Ability to respond to unexpected events in timely and effective manner

These are complex adaptive systems

  - Affordable "IT renovations" that provide improved and new capabilities in the short-term

Be innovative and adaptive

SE

Necessity is the mother of invention..

System of Systems*

- A system will be called a System of Systems (SOS) when:
  - The component systems achieve well-substantiated purposes in their own right even if detached from the overall system;
  - The component systems are managed in large part for their own purposes rather than the purposes of the whole;
  - It exhibits behavior, including emergent behavior, not achievable by the component systems acting independently
  - Functions, behaviors and component systems may be added or removed during its use
- The System of Systems concept is really at the heart of enterprise architecting and engineering

* After Maier, Sage, Levis…

Avoid destructive emergent behaviors

I'm sure this guy still wonders why he got fired that day

Components of Complexity

- Three components
  - Variety
  - Connectedness
  - Disorder
- Something is more complex if
  - There is greater variety among its inputs and systems
  - The number of connections is greater than less
  - Variety and connections are mixed and tangled-up, not orderly
- Number, as contrasted to variety, is not an essential characteristic of complexity
  - 10 paving stones is not more complex than 20 paving stones

Axiom: Complexity can be managed but not reduced

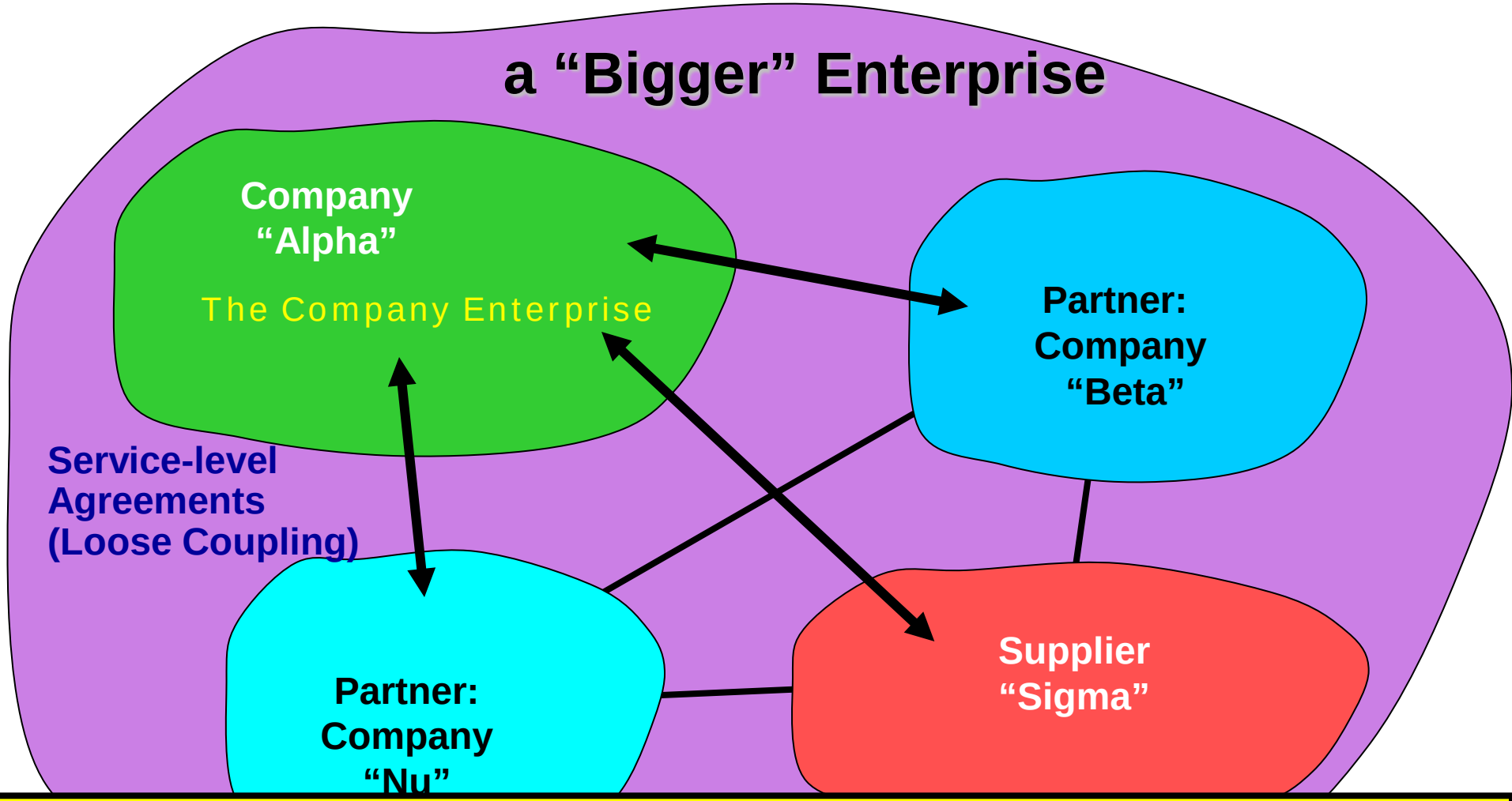
What do we mean by “Enterprise”?

- An organization … or a
  - Collection of organizations
    - May have many partners and suppliers

Has a well-defined objective and set of missions

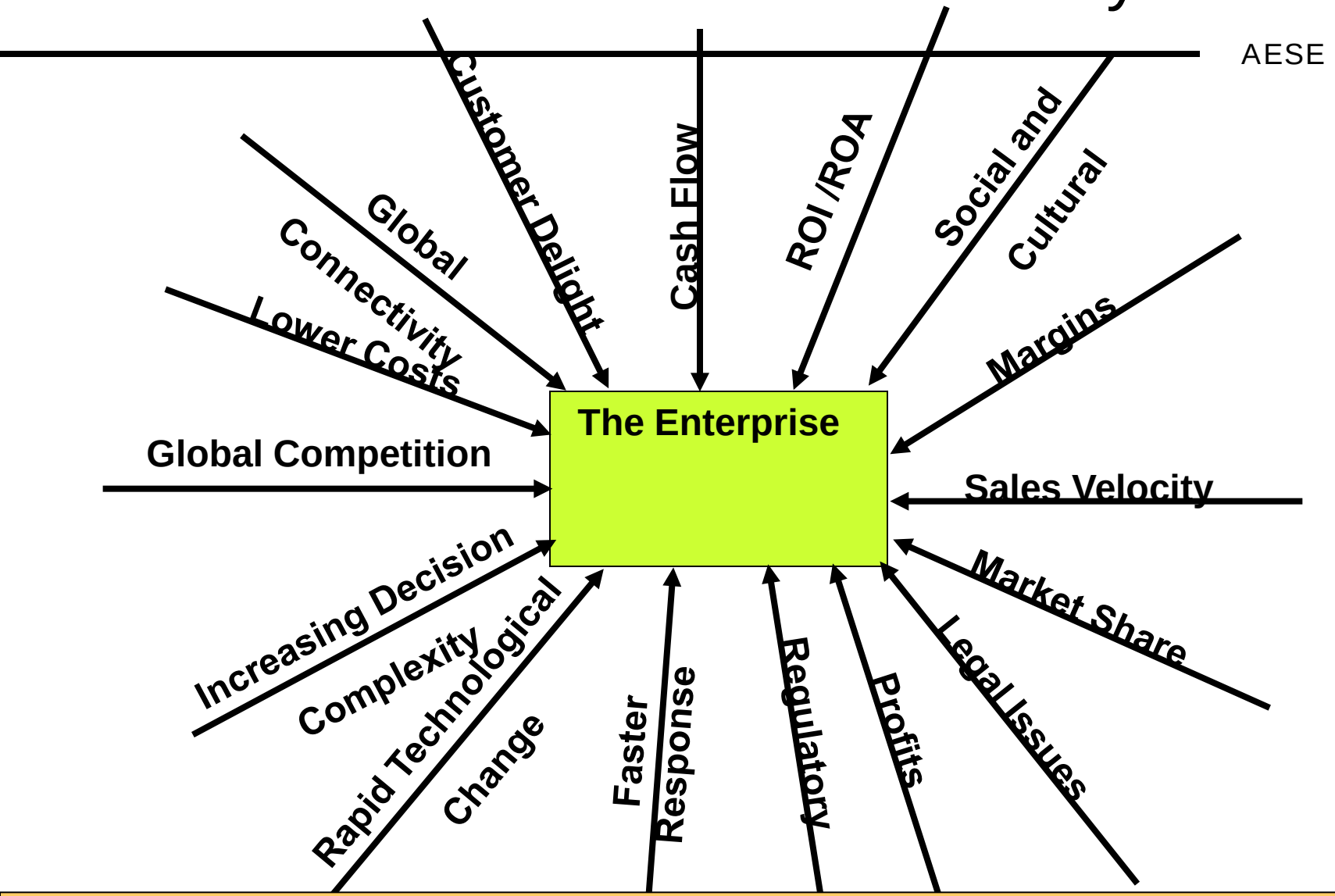
- An enterprise includes interdependent resources (i.e., *people*, *organizations*, and *technology*) that must coordinate their functions and share information in support of a common mission (or a set of related missions) in a *context* involving *culture*, *management*, and *processes*

ENTERPRISE(s)
(where are the boundaries?)



Enterprises are **nested entities** with interactions possible across different levels

Environmental Forces – Driven by Events



The environment is very complex and enterprises must adapt and respond effectively to survive

A Guiding Principle

This quote captures the essence of enterprise systems engineering

"To manage a system effectively, you might focus on the interactions of the parts rather than their behavior taken separately."

-- Russell L. Ackoff

Product systems engineering focuses on the behavior of the parts

Create effective interfaces for legacy systems

AESE

Gee, guys... that seems like an awful lot of protective gear for such a small chlorine gas leak..."

What Is the Problem?

- *The majority of information systems developments are unsuccessful*

- Some general observations
  - ***20 percent to 30 percent*** of all developments are ***total failures*** in which projects are abandoned
  - ***30 percent to 60 percent*** are ***partial failures*** in which there are time and cost overruns or other problems
  - The minority are those counted as successes

- *The larger the development, the more likely it will be unsuccessful.*

Reference: Shaun Goldfinch, University of Otago, New Zealand, Pessimism, Computer Failure, and Information Systems Development in the Public Sector, Public Administration Review, September|October 2007

Problems of Control (continued)

- Because of problems of ***agency***, immense complexity, and the interaction of ***human beings*** having, at best, only ***bounded*** or even limited ***rationality***, it is difficult to understand and control large IS developments
- The sheer complexity of IS developments means that humans with not-unlimited abilities are faced with ***informational overload***

Some key implications:

1. ***1.*** The development of complex, enterprise systems behave in manners analogous to ***nonlinear*** dynamic systems
2. The methods of ***reductionism*** can not be applied
3. A ***waterfall*** development process is doomed to failure

⇨ Apply the principles of ***systems thinking***

Don’t be restricted by the practices of the past

"And to think... those wimps at the power company use straps and cleats to get up this high!"

Return to the Basics of Systems Thinking[1]

AESE

- Operational definition of a "systems methodology" involves three ***interdependent*** variables
  - Structure
  - Function
  - Process

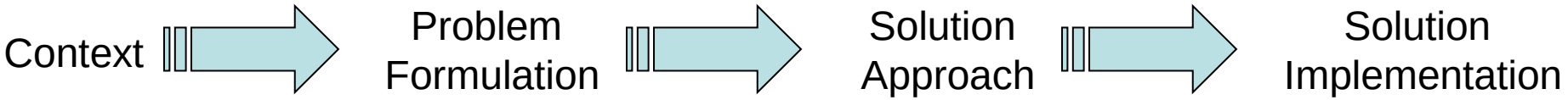
  that, together with the ***environment*,** define the ***context***
- ***Structure*** defines components and their relationships and constraints– synonymous with input, means, and effects
- ***Function*** defines the outcome – synonymous with output
- ***Process*** defines the sequence of activities required to produce the outcomes – how to do the function

➔ Development process is necessarily ***iterative***

[1] Gharajedaghi, Jamshid, Systems Thinking: Managing Chaos and Complexity, Butterworth Heineman, 1999

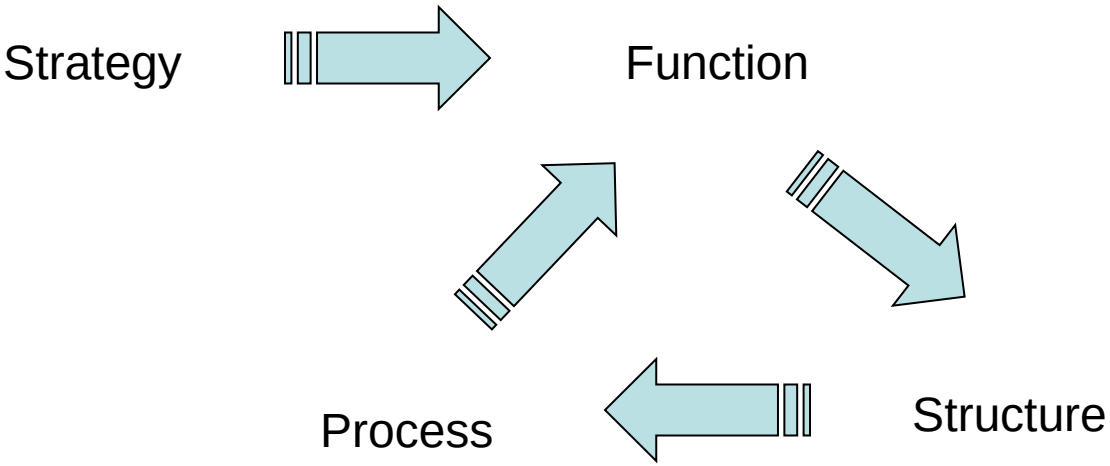
Enterprise System Development Fundamentals

The Problem Solving Structure



The Systems Methodology

Understanding the ***Context*** is manifested as a ***strategy*** which drives the systems problem in terms of ***three interdependent variables*** and the solution is developed ***iteratively***

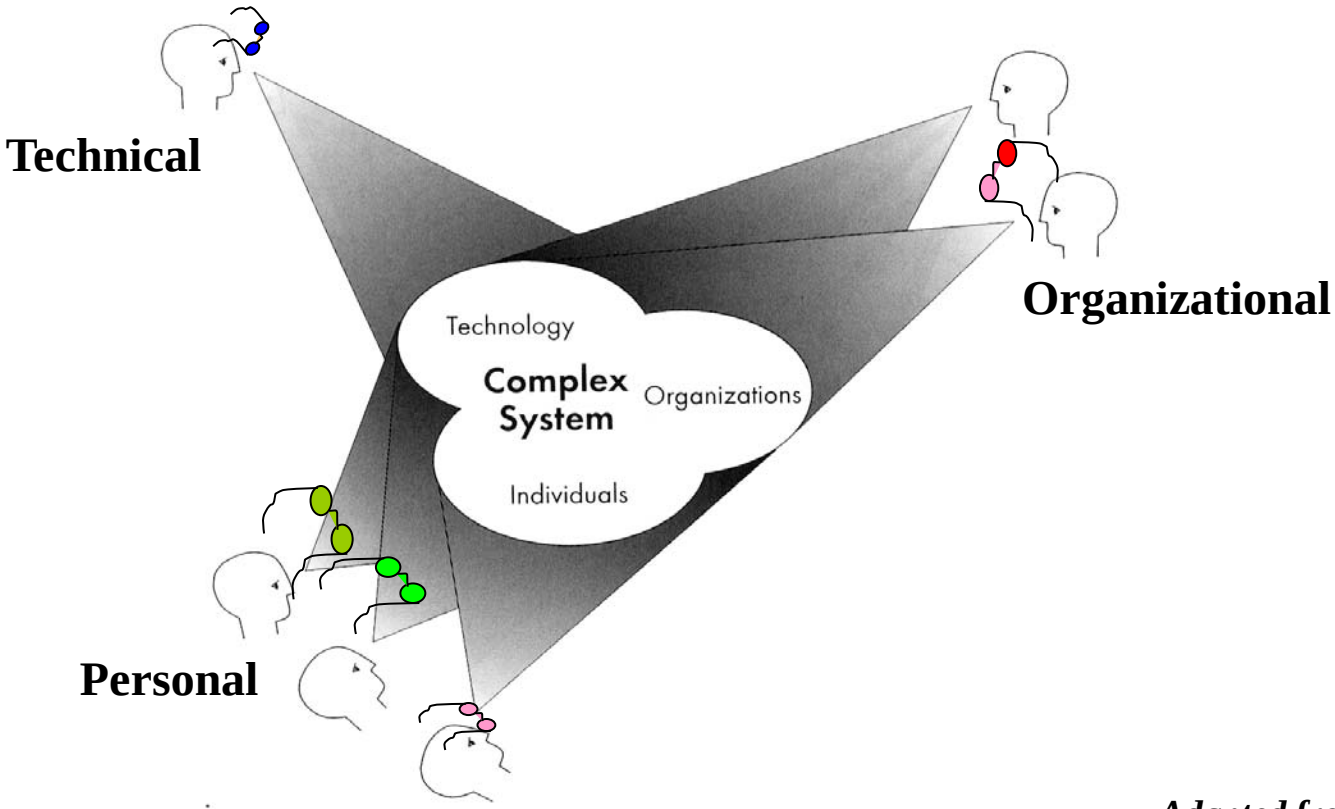


Make small changes quickly

"Jack stands? Hah! Who needs 'em?"

Multiple Perspectives on Strategy and Architecture Development

Perspectives must be considered for each *evolutionary step*



Adapted from Linstone

Key stakeholders from all relevant domains should be engaged and involved

Involve stakeholders in assessing the outcome space

Step 1: Remove shoes.

Step 2: Place metal ladder in water.

Step 3: Begin using power tools while standing barefoot on metal ladder in water.

What do we mean by “Architecture”?

- IEEE STD 610.12 defines *“architecture”* as *“the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time*.”

- Simply, “architecture” defines the ***structure*** of
  - “Components”
  - “Connections”
  - “Constraints”

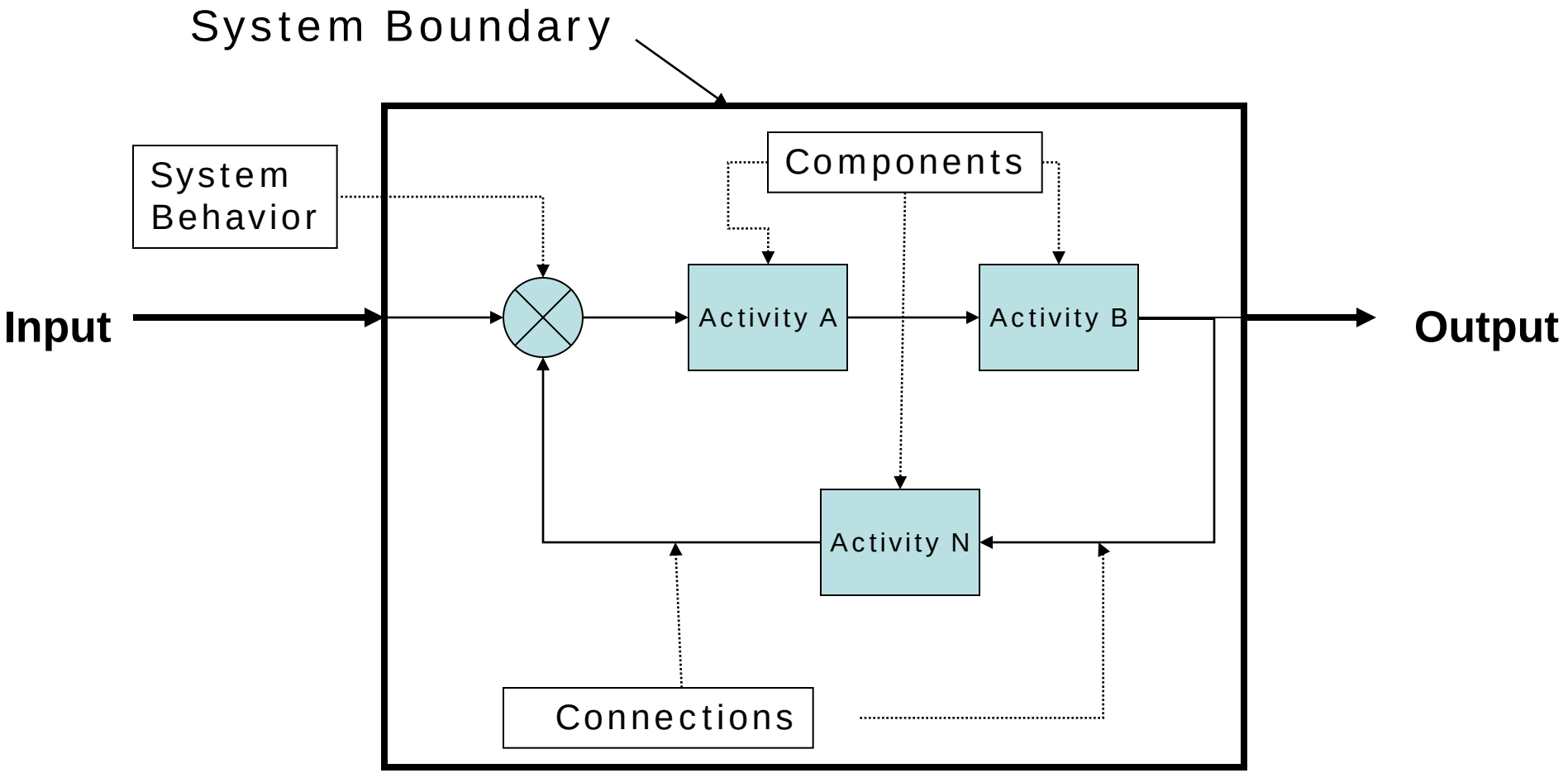
  for systems that must support *business-defined* “objectives” and “missions” providing business **capabilities --** yielding the **C4 model**)

The C3 model of architecture

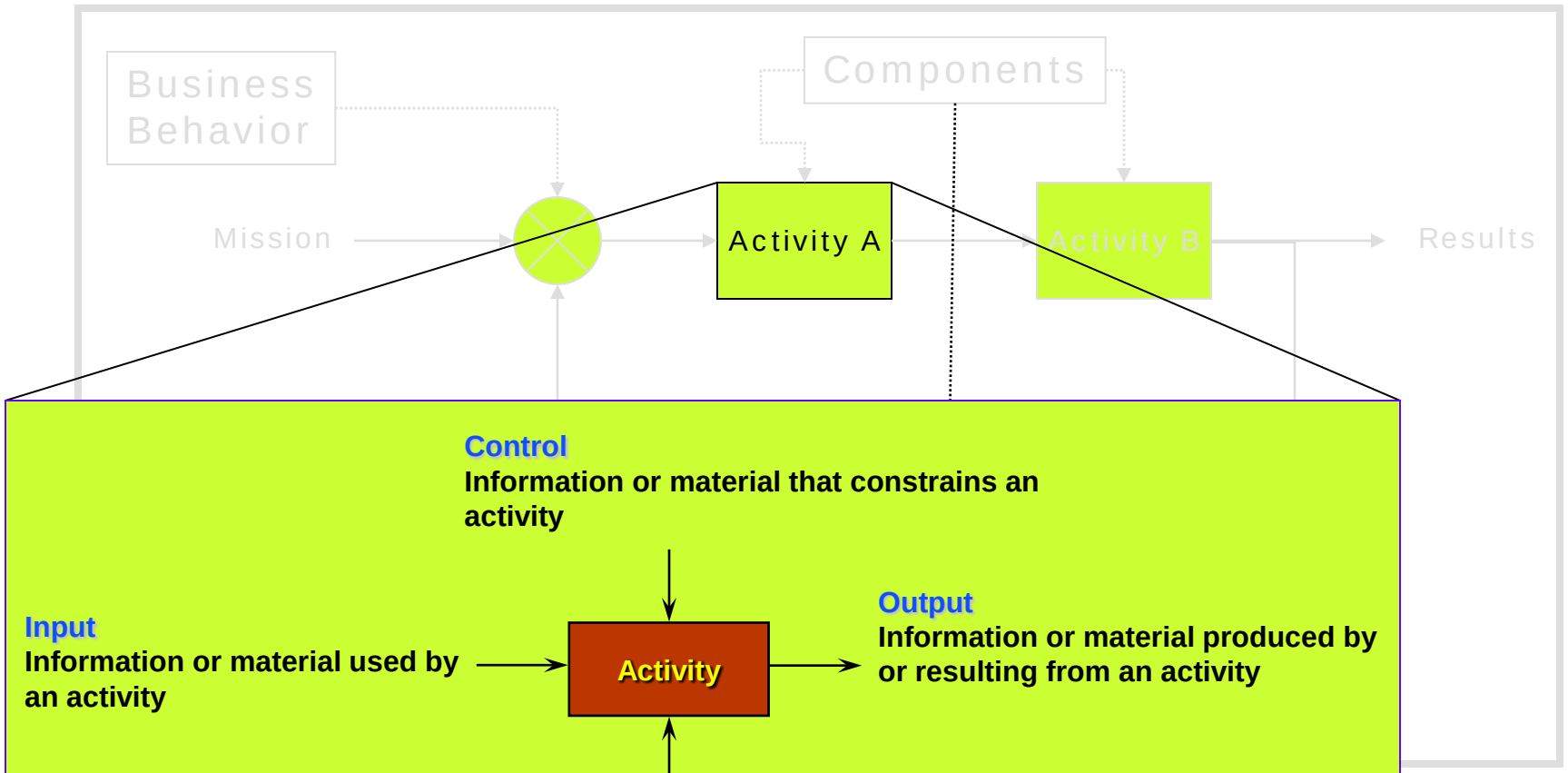
Conclusion: Architectures have been used for engineering systems since before they were called “architectures”

An Architecture Example

- A control system model



The "ICOM" Model



Control
Information or material that constrains an activity

Input
Information or material used by an activity

Activity

Output
Information or material produced by or resulting from an activity

Mechanism
Usually a person, machine, or existing system that performs or is used to perform the activity - physical model

An "activity" is the transformation (or process) of "inputs" into "outputs" performed by "mechanisms" under the constraints of "controls"

Upfront Engineering (from 2009 DSB Study)

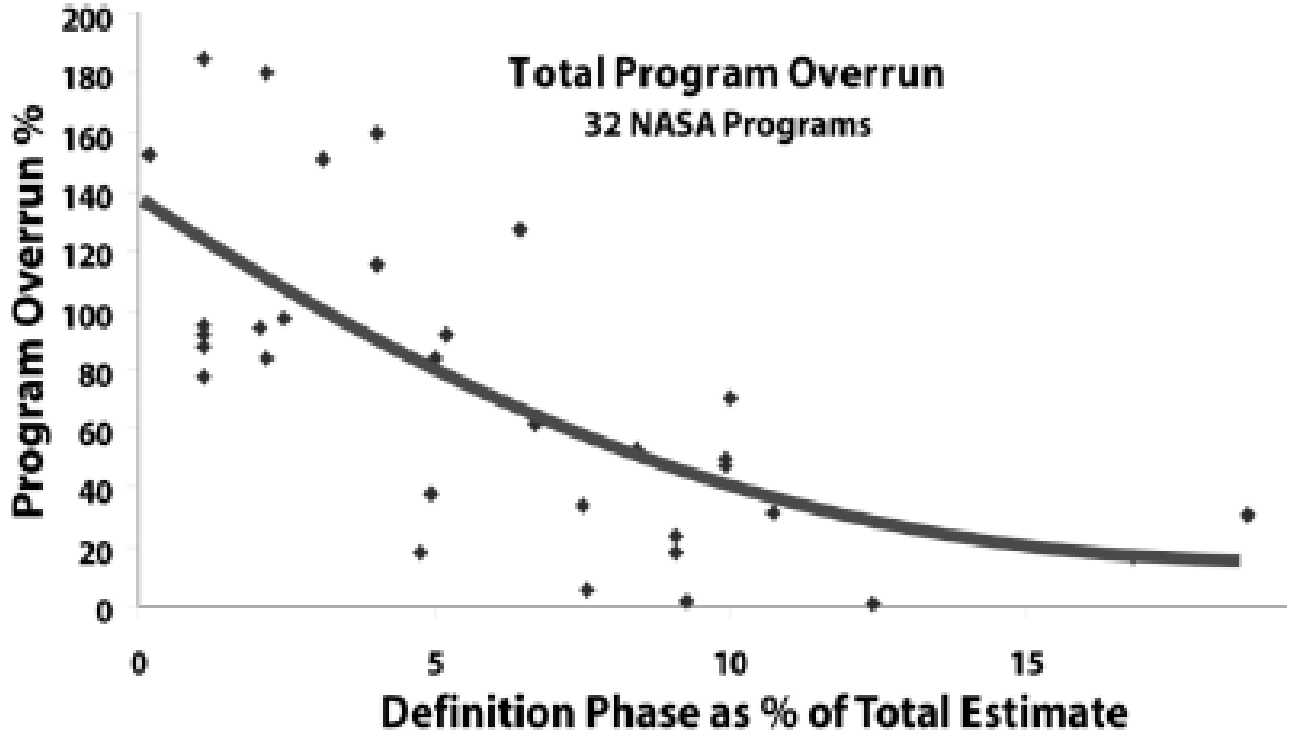
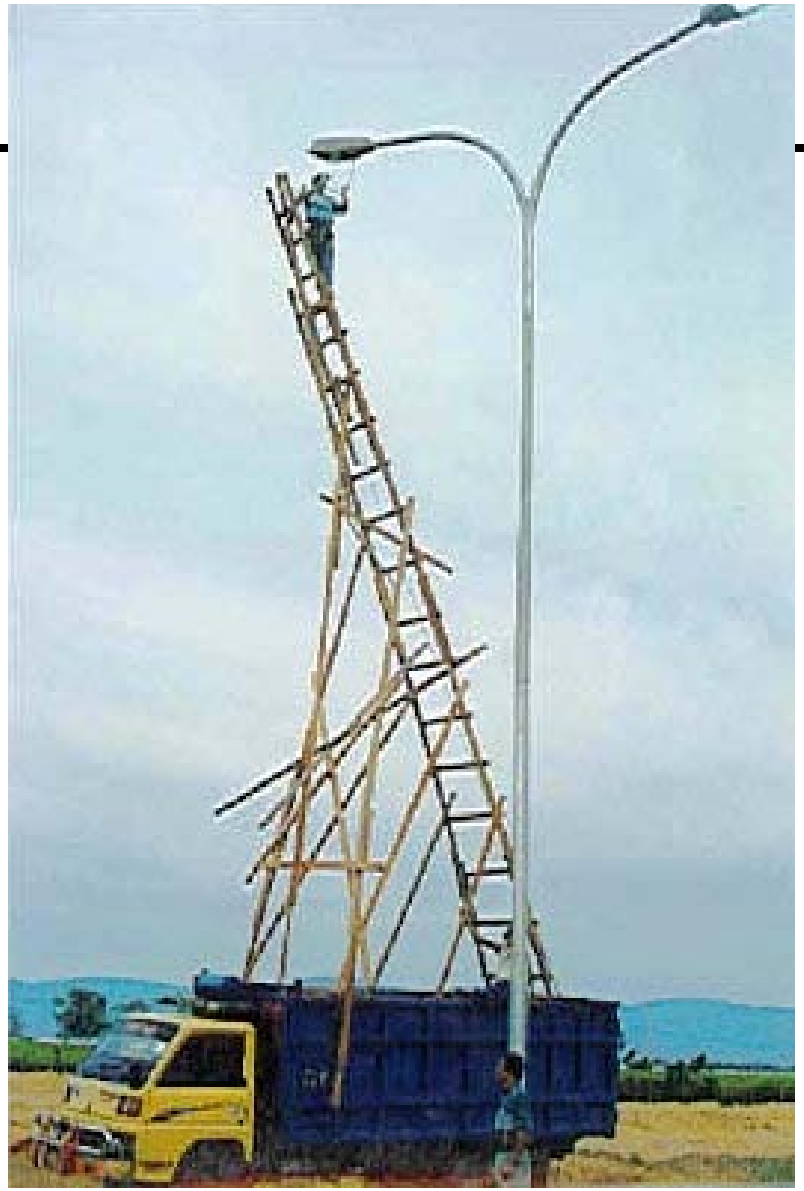


Figure 9. Rigorous Upfront Engineering Reduces Program Cost Overrun

Implication: Emphasize architecture development during Upfront Engineering!

Make use of well-tested patterns



How drunk do you have to be before this starts looking like a good idea?

Starting Points

- **Architecture Definition**
  - Components
  - Connections
  - Constraints

that support

  - Capabilities

- **Complexity**
  - Variety
  - Connectedness
  - Disorder

- How is a process described?
  - Components
  - Connections
  - Constraints
- What is a function?
  - Output of a component
- What is a structure?
  - Components
  - Connections
  - Constraints

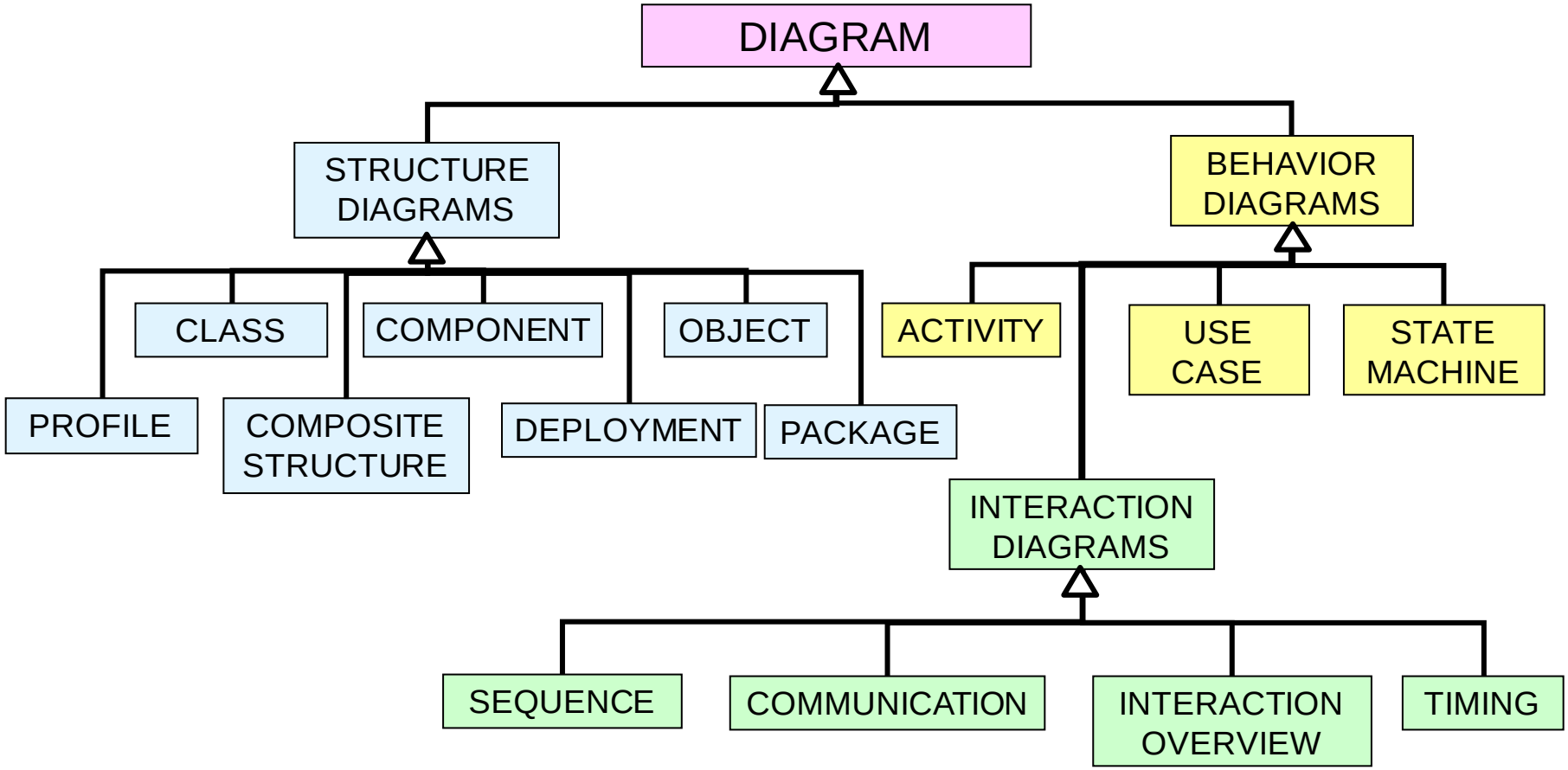
• Structure is a **static** description derived from strategy

•Process is a **behavioral** description derived from strategy

•Function is **output** of the structure and process

The UML 2.2 Diagrams

AESE



- UML 2.2 classifies 14 diagrams into three groups/views: Behavior Diagrams, Interaction Diagrams (a subset of behavior ones), and Structure Diagrams

Network Connections

- What is a ***small world*** network?
  - Six degrees of separation – Stanley Milgram in the 1960s
  - How can 6 billion people be so tightly linked?
- The Kevin Bacon game
  - 500,000 names in the Internet Movie Database
  - Bacon has played with 1472 other actors in his movies (one degree of separation)
  - 110,315 actors have been in a movie with these 1472 (two degrees of separation)
  - Average number of links to Bacon is 2.896
  - Rod Steiger is best connected – Bacon is 66th

Network Connections (continued-2)

- The Internet
  - Researchers have found that the Internet is a small world
    - Typically, four links are required to go from computer to computer with the most around ten
  - Further, the distribution of the number of routers as a function of the number of nodes follows a ***power law***
- World-wide Web
  - Barabassi found web pages have a link distribution that follows a power law also
  - And the Web is a small world, also, with a diameter (number of clicks to get from one document to another) of about nineteen

Network Connections (concluded)

- Small worlds of two structural types have been discussed
  - Clusters linked with a relatively few random links to other clusters (loose coupling)
  - A hierarchical structure with a few nodes with many links and many with few links, satisfying a power law distribution
- Both types are reflected in nature
- Implying that apparently complex systems have an order that repeats across many areas of chemistry, biology, and society
  - Reductionism plays no role in reaching these conclusions
  - Systems in their whole are addressed and analyzed

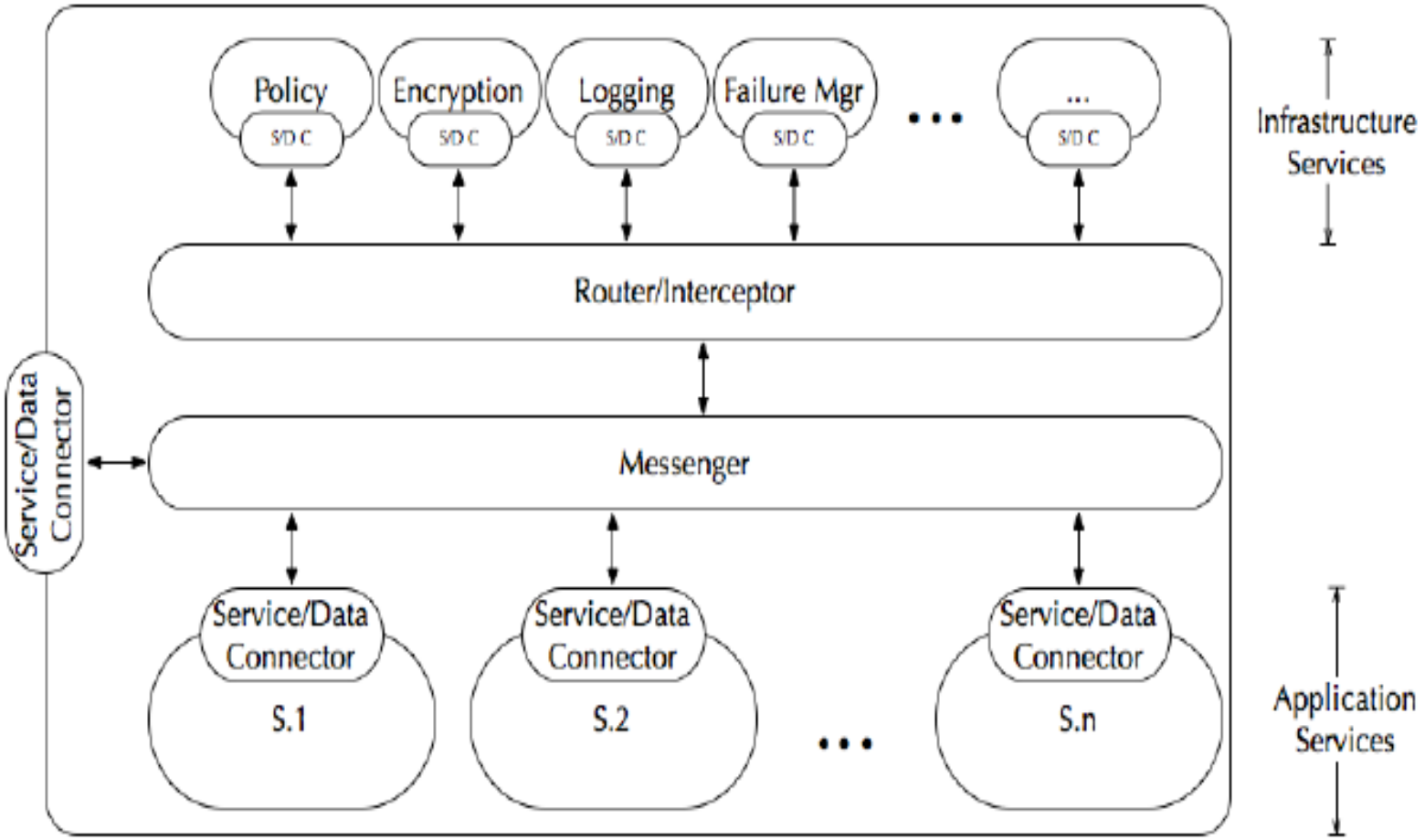
It is interesting to note that the brain has a small world structure

So Why Does This New Network View Matter?

- We need to view Enterprise systems with modern attributes:
  - Plan for iterative growth spirals
  - Emphasize the up-front development phase to reduce expensive, time-consuming later problems and errors
  - Development is guided by the use of enterprise architectures
  - Use loose couplers as the basis for data strategies
  - Enable components to come and go from the system
- An ***Enterprise Service Bus*** provides the linkages for a hub
  - Built using loose couplers
  - Federated SOAs (i.e., ESBs) provide the larger network model

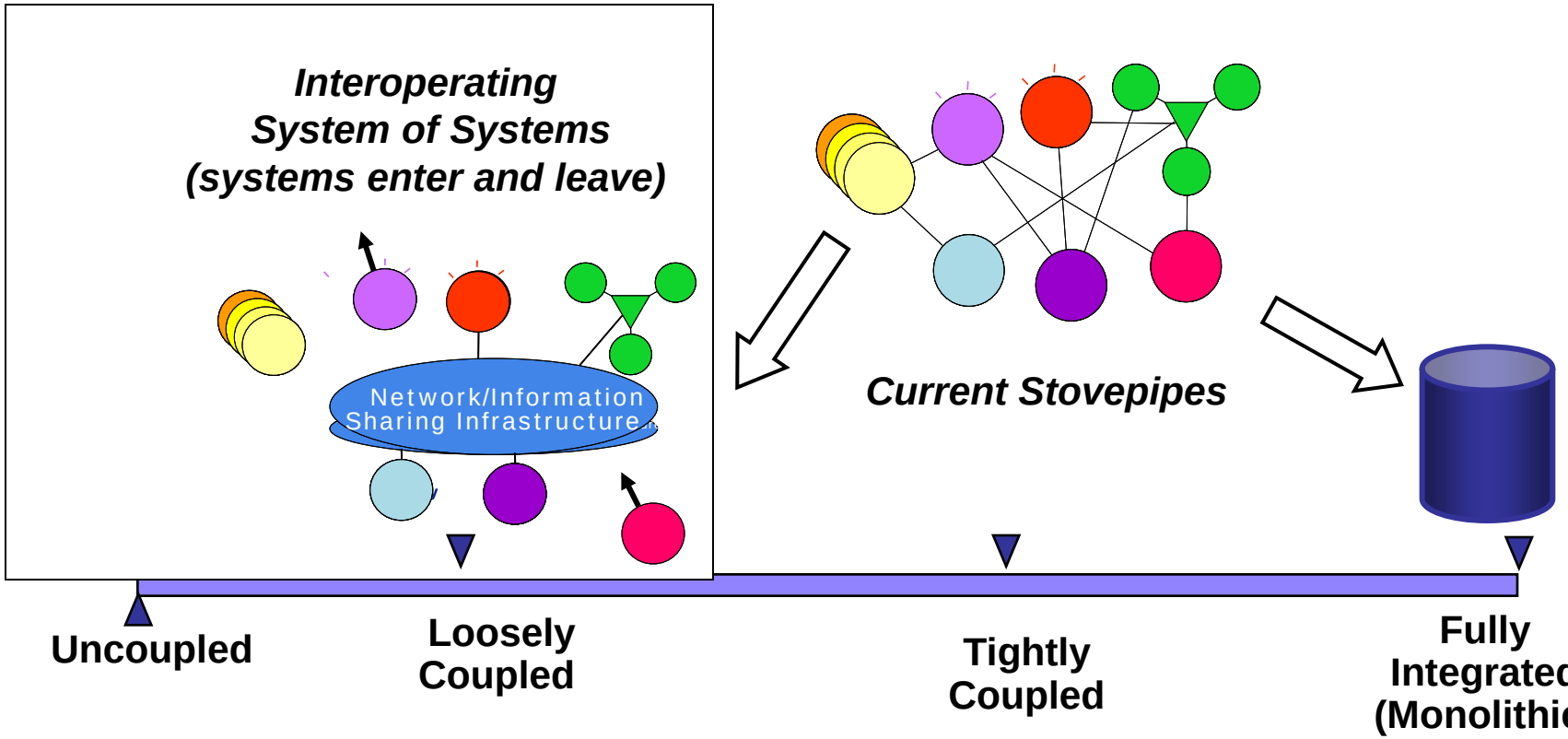
➔ ***Service-Oriented Architectures (SOA)*** suggests a development framework

Rich Services ESB



Enterprise integration covers the spectrum from tight (system) integration into a single large scale system to interoperation (information integration) in a loosely coupled system of systems

Focus of AESE Program



Enterprise Integration - the implementation of (widely) shared functional interfaces between domains which allow (but do not necessarily require) access to, use, or control of resources and capabilities within the domains.

The Service-Oriented Architecture (SOA) Approach

Past and Current Development Paradigm

- Function oriented
- Build to last
- Prolonged development cycles

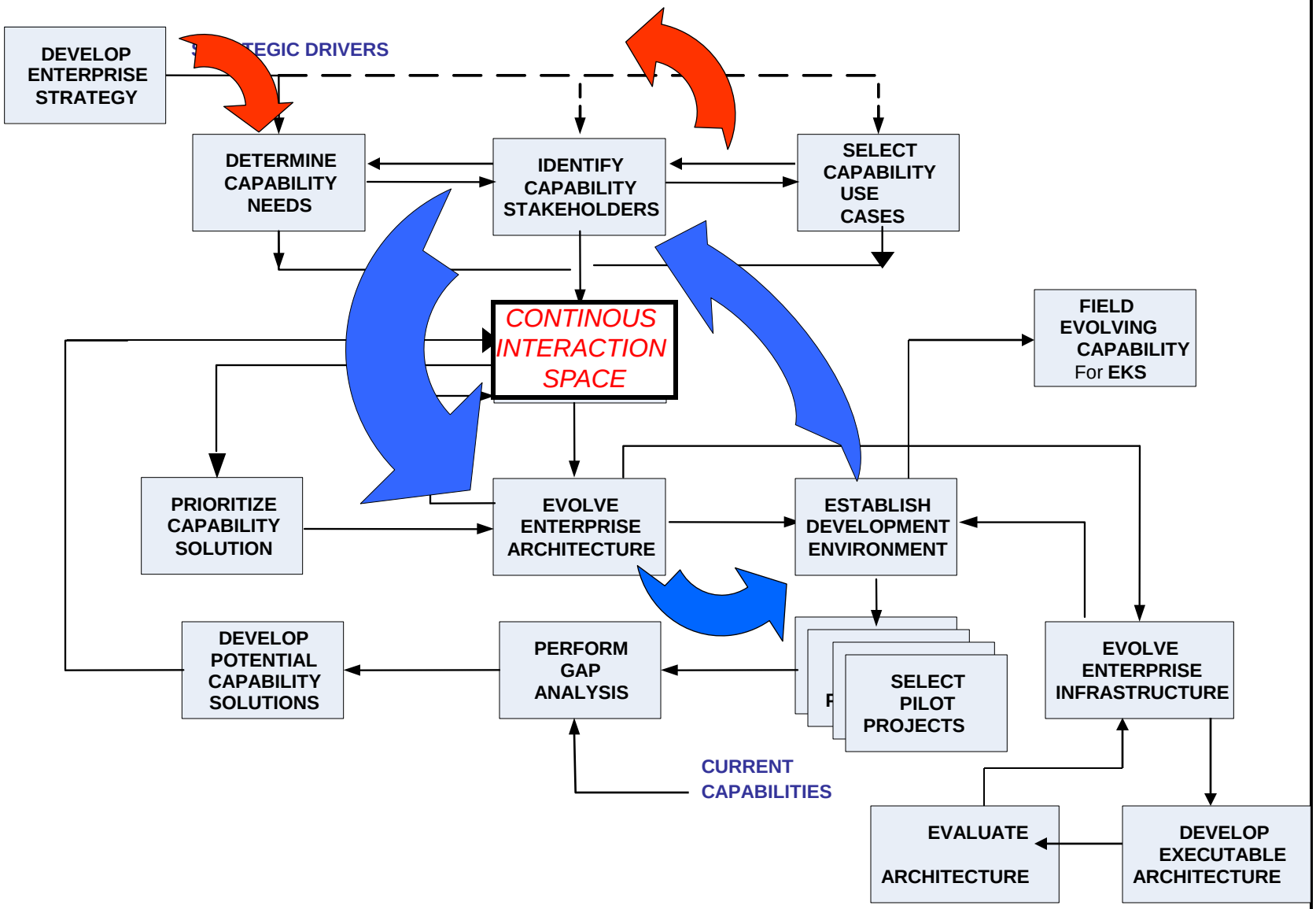
- Application silos
- Tightly coupled
- Structuring applications using components and objects
- Known implementation

The SOA Development Paradigm

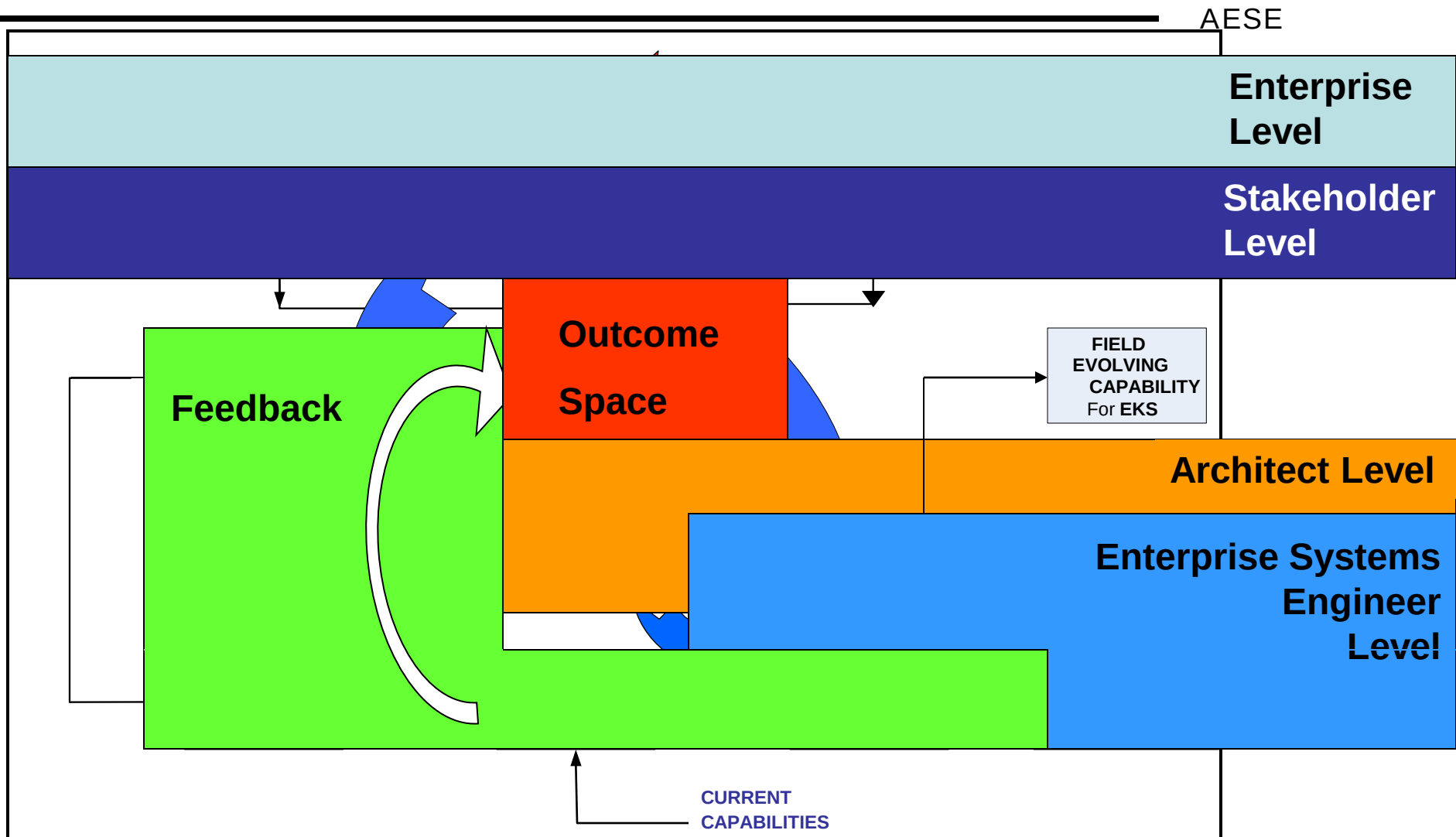
- Process oriented
- Build to change
- Incrementally built and deployed

- Orchestrated solutions
- Loosely coupled
- Structure applications using services
- Implementation abstraction

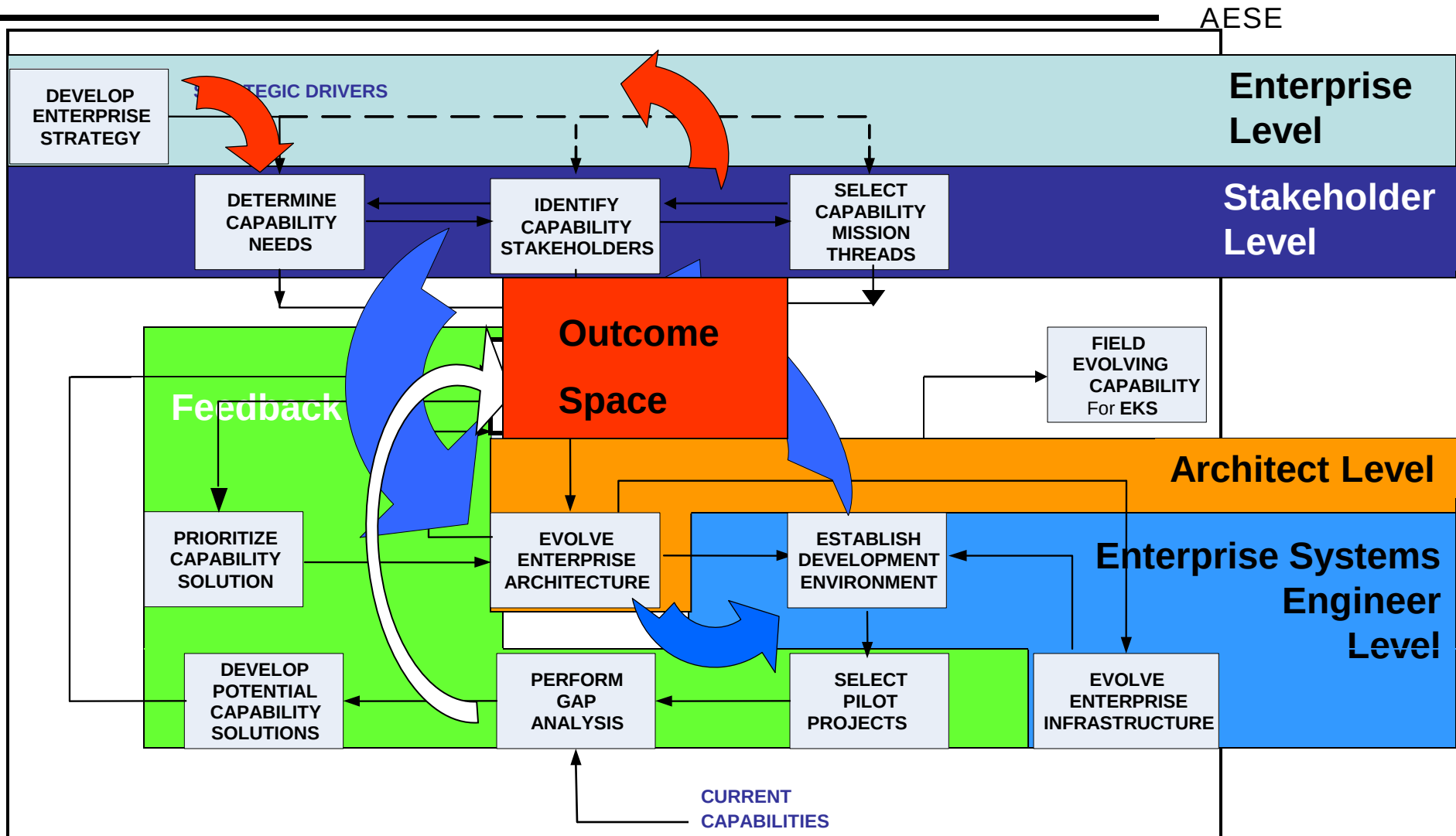
AESE Development Process



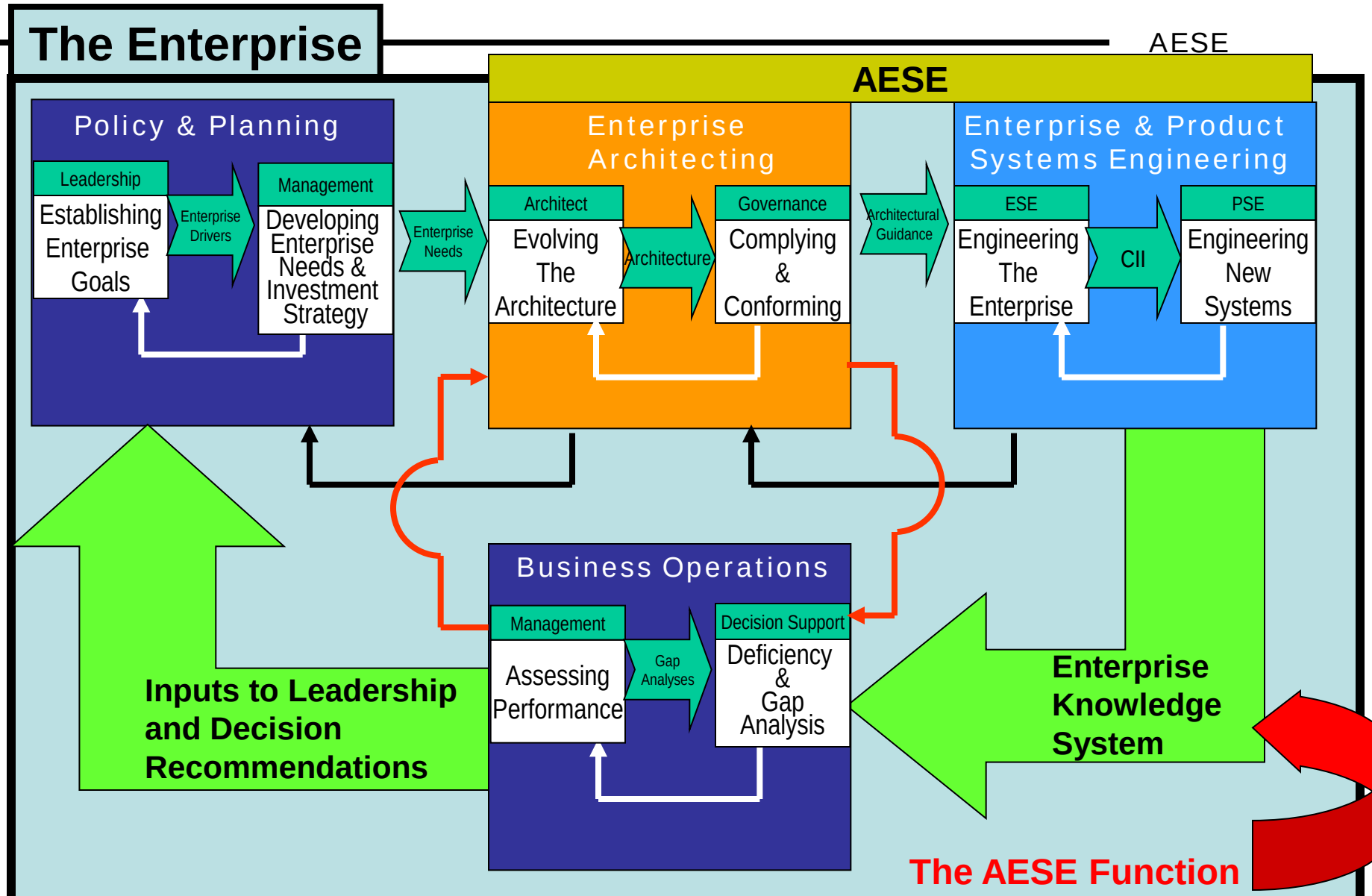
AESE Agile Development Process



AESE Agile Development Process



The AESE Structure



AESE Program Goals

- Win-Win-Win
  - ***University***

    Provides a ***unique*** degree offering that enhances the stature and relevance of UCSD
  - ***Students***

    Provides a ***unique*** learning opportunity to gain an integrated, state-of-the-practice, understanding of issues critical to enterprises in an increasingly networked world
  - ***Organizations***

    Provides an ***immediate return*** on investments in the graduate education of fast track employees
- ***Organizations*** are encouraged to ***sponsor a team*** of 3 to 5 employees who will work on a team project of interest to senior managers
- ***Individuals*** are encouraged to participate and will become a member of a team of 3 to 5

Summary

- The AESE Leadership Program was offered as UCSD Certificate Program for the past four years
  - A total of 72 students have completed the program over the course of these offerings
- The MAS graduate program is being offered this year with 12 students
- The following organizations have sponsored teams and individuals to the program

Boeing	QinetiQ-NA
Booz Allen Hamilton	Sentek Global
Calit2	Solar Turbines
The MITRE Corporation	SPAWAR Systems Center
Northrop Grumman AS	ViaSat
Northrop Grumman IS	

Joint MAS Graduate Program

- The program is a joint activity of
  - Jacobs School of Engineering
    - Computer Science and Engineering Department
  - Rady School of Management
- Jacobs School offers five courses
  - Complexity and Large-scale Systems (Fall)
  - Enterprise Architecting (Winter)
  - Modeling, Simulation, and Analysis (Winter)
  - Engineering Essentials for Distributed Systems (Winter)
  - Patterns for Enterprise Architecting (Spring)
- Rady School offers four courses
  - Essentials for Business Practice (Fall)
  - Leadership Skills, Values, and Teamwork (Fall)
  - Risk and Decision Analysis (Spring)
  - Managing Stakeholder Relationships (Spring)

The Academic Offerings

- AESE Leadership Program is offered during one academic year, including the summer
- Three courses are offered each of the Fall, Winter, and Spring quarters
- Each course meets for four, 8 hour days
  - Two courses/quarter meet on Friday and Saturday
    - Class is scheduled from 8 AM to 5 PM each day
    - Third and fourth lecture days are separated by two weeks
  - One course/quarter is conducted as a four day workshop
    - Meets from Wednesday through Saturday
    - Class is scheduled from 8 AM to 5 PM each day

The Academic Offerings (concluded)

- Each quarter, there is a two-day, team project workshop (1 graduate credit) during the last week of the quarter
- There is a four day, team project workshop (3 graduate credits) at the end of August
  - Teams work on the final presentation and final team project report
    - Tuesday, Wednesday, and Thursday
  - On Friday, teams present their team project results to an audience of corporate sponsors and AESE program faculty
- On Friday evening, there is an AESE Leadership Program dinner for students, corporate sponsors and faculty

Financial Information

AESE

- The AESE Leadership Program Fees are $675 per unit plus quarterly registration fees. Total for the program is $29,875.75 for 2011-2012

  Broken out by Quarter:

  Fall 2011 (13 units): $9,255.50
  $8,775 (Course Fees)
  $ 480.50 (Reg Fees)

  Winter 2012 (13 units): $9,255.50
  $8,775 (Course Fees)
  $ 480.50 (Reg Fees)

  Spring 2012 (13 units): $9,255.50
  $8,775 (Course Fees)
  $ 480.50 (Reg Fees)

  Summer 2012 (3 units): $2,109.25
  $2,025 (Course Fees)
  $ 80.25 (Reg Fees)

  The total amount due assumes the student waives the mandatory health insurance of $548 per quarter.

- NOTE: MAS programs are entirely self-supporting and receive no funding from the University of California

Graduate Course Descriptions

Fall Quarter

- **Essentials for Business Practice (Rady School)**
  - Strategy and Strategic Thinking
  - Finance and Investment Planning
  - Business Strategy and Operations
  - Marketing Strategy and Implementation
- **Leadership Skills, Values, and Team Building Workshop (Rady School)**
  - Understanding Self & Others
  - Building Collaboration
  - Influence
  - Group Dynamics
  - Emotional Intelligence
  - Team Building

Graduate Course Descriptions (continued)

Fall Quarter (continued)

- **Complexity and Large-scale Systems (Jacobs School)**
  - System and Event Complexity
  - Complexity Case Study: The Beer Game
  - Enterprise Transformation
  - Iterative and Spiral Development
  - Agile and Plan-driven Development
  - Managing Complex Projects
    - Case Study: The Oceans Observatory Initiative
- **Team Project 1**

Graduate Course Descriptions (continued)

Winter Quarter

- **Enterprise Architecting (Jacobs School)**
  - Architecture Frameworks
  - Enterprise Architecting and Use Cases
  - Ontologies and Domain Models
  - Service-Oriented Architectures and the Enterprise Service Bus
  - SOA Security
- **Engineering Essentials for Distributed Systems Workshop (Jacobs School)**
  - UML Basics and Enterprise Architect
  - Version Control
  - Exercises in Domain Modeling and Architecture Development
  - SOA Infrastructure
  - SOA Governance

Graduate Course Descriptions (continued)

Winter Quarter

- **Modeling, Simulation, & Analysis (Jacobs School)**
  - Architecture Description
  - An Object-oriented Architecture Design Process
  - Discrete Event Dynamic Systems and Colored Petri Nets
  - Executable Architectures
  - Business Process Modeling
  - Management of Architecture Development

- **Team Project 2**

Graduate Course Descriptions (continued)

Spring Quarter

- **Patterns for Enterprise Architecting (Jacobs School)**
  - Introduction to Pattern Concepts
  - Patterns for Enterprise Integration
  - Service Patterns
  - Event-driven Architectures and Decision Support Systems
- **Decision and Risk Analysis (Rady School)**
  - Human Decision-making
  - Competing on Analytics
  - Analytics
  - Risk & Utility Theory
  - Investment Valuation and Real Options

Graduate Course Descriptions (concluded)

Spring Quarter

- **Managing Stakeholder Relationships Workshop (Rady School)**
  - Build & Leverage Business Relationships
  - Create Business Development Strategies
  - Write Winning Proposals
  - Strategic Account Planning

- **Team Project 3**

Summer Quarter

  - Team Project Workshop and Final Presentation