



A Double Challenge in Software Requirements Management

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Philip Boxer, Lisa Brownsword (presenter),
Dennis Smith

16 October 2007



Briefing Objective and Agenda

Objective

- Instigate an alternative way of viewing requirements in a system-of-systems context

Agenda

- Explore implications of a changing world
- Describe an alternative reasoning framework



Our Changing World—Aircraft Example



Steady State Environments

- System functionality known and stable
- Designers can know how an aircraft will fly before it takes off
- Requirements can be fine-grained and very constraining

Dynamic Environments

- System functionality driven from unanticipated and changing behaviors by the end user community
- Using zero or negative static stability in an aircraft allows for dramatic enlargement of the aircraft's flight envelope
 - Stability achieved dynamically through flight control systems that mediate between pilot and aircraft
- Designers do not know how the aircraft will fly before it takes off



Key Challenge—Increasingly Turbulent Operational Contexts

- Customers and users want specialized solutions in ever shorter time frames continuously adapted to their changing and evolving situations.
- Suppliers and systems have to become more agile to respond.



Product-Based



Users want products or services that can be provided in a way that is unaffected by how they are used



Solution-Based



Users want integrated solutions that are customized to their context, but in a way that can be specified beforehand



Customer Experience-Based

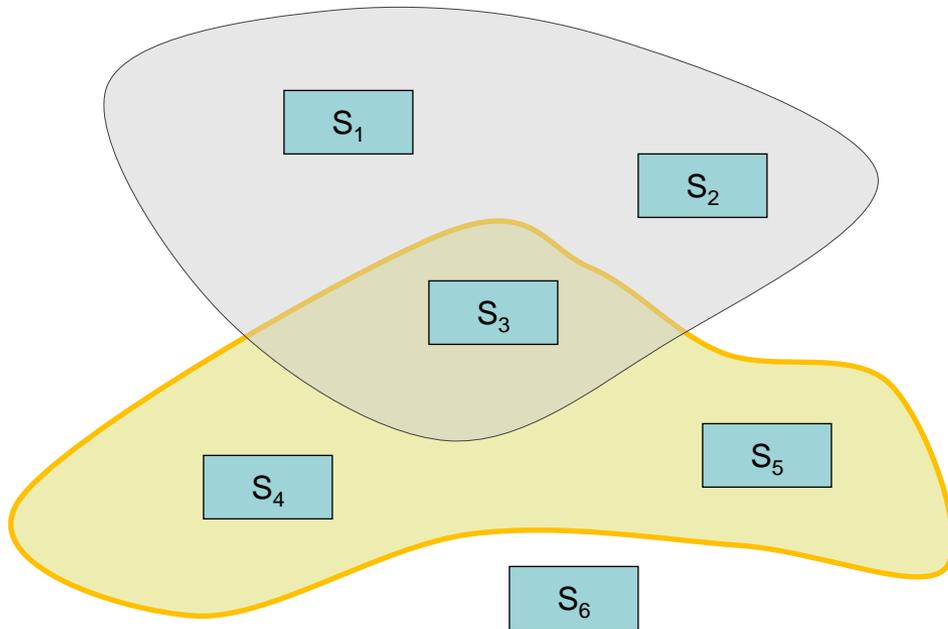


Users want integrated solutions that are customized in ways that change and evolve throughout the life of the mission that they support

*'Turbulence' as per "The Causal Texture of Organizational Environments", Emery F E and Trist E, Human Relations 1965, 18, pp 21-32.
Categories adapted from "The New Frontier of Experience Innovation", Prahalad and Ramaswamy, MIT Summer 2003*



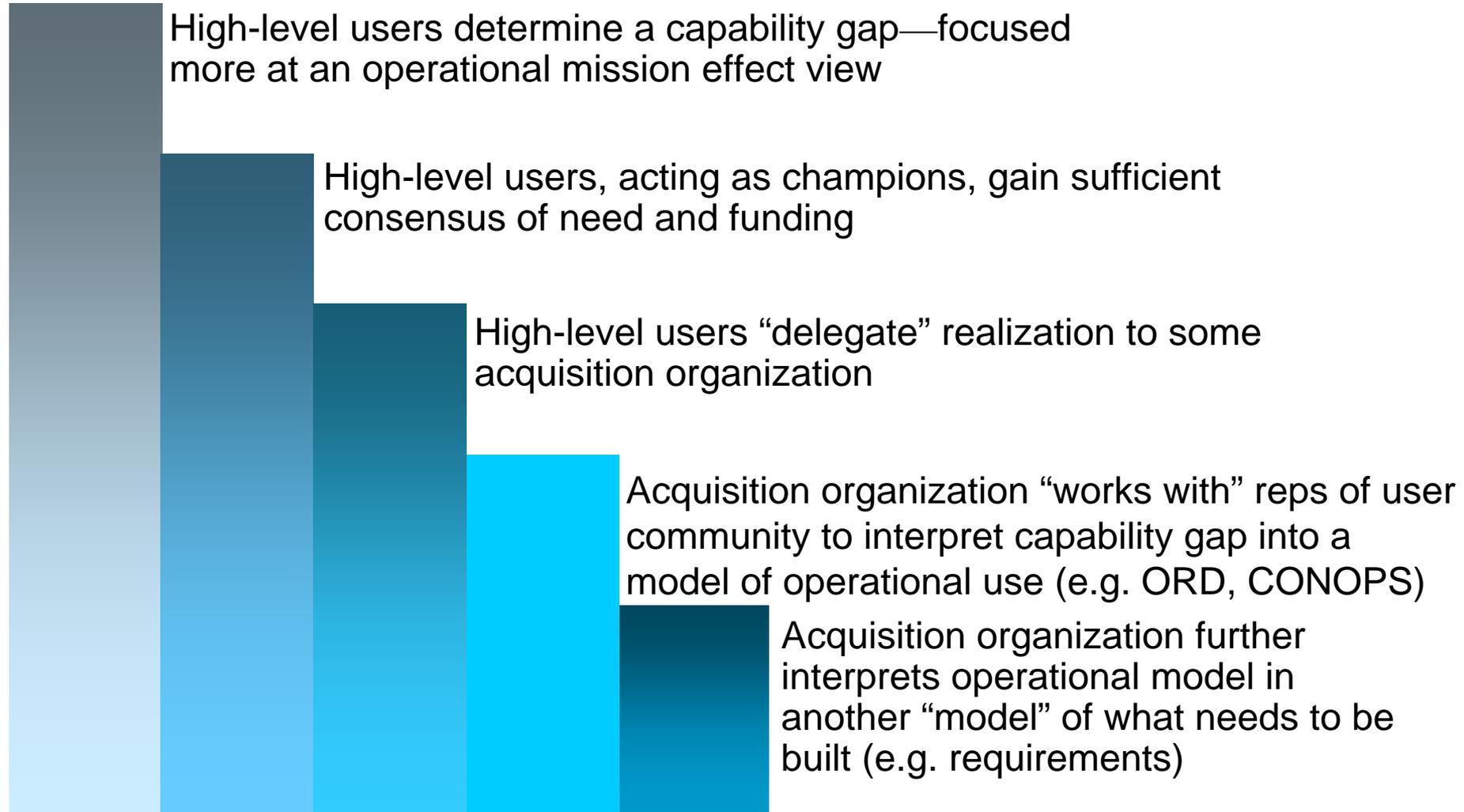
Creating, Using, and Evolving Composites of Systems



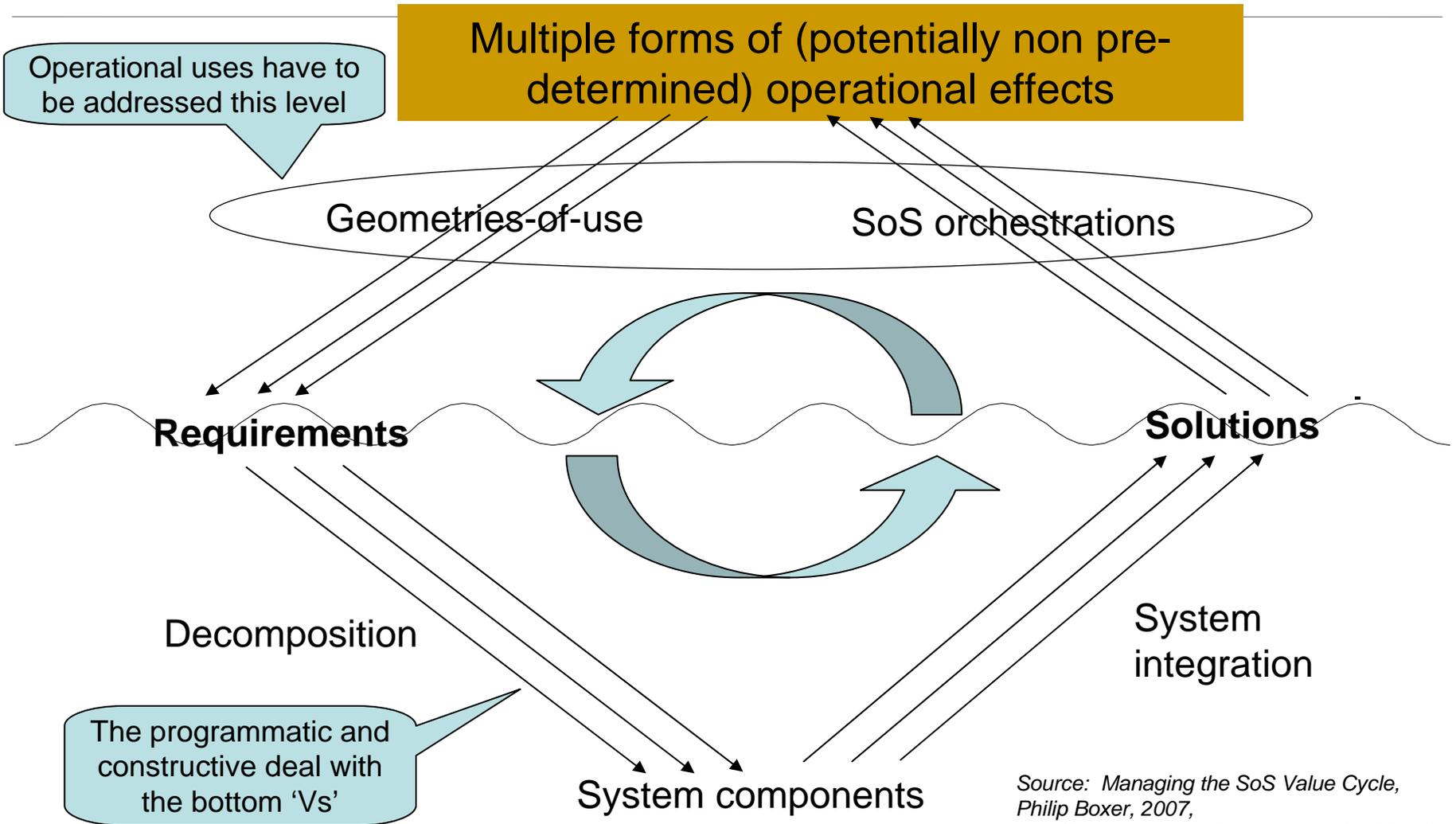
- Which systems are likely candidates?
- Will they continue to be effective in the system of systems?
- Many of the systems are well into their acquisition life cycle
- Many systems were specified and built as “standalone” capability



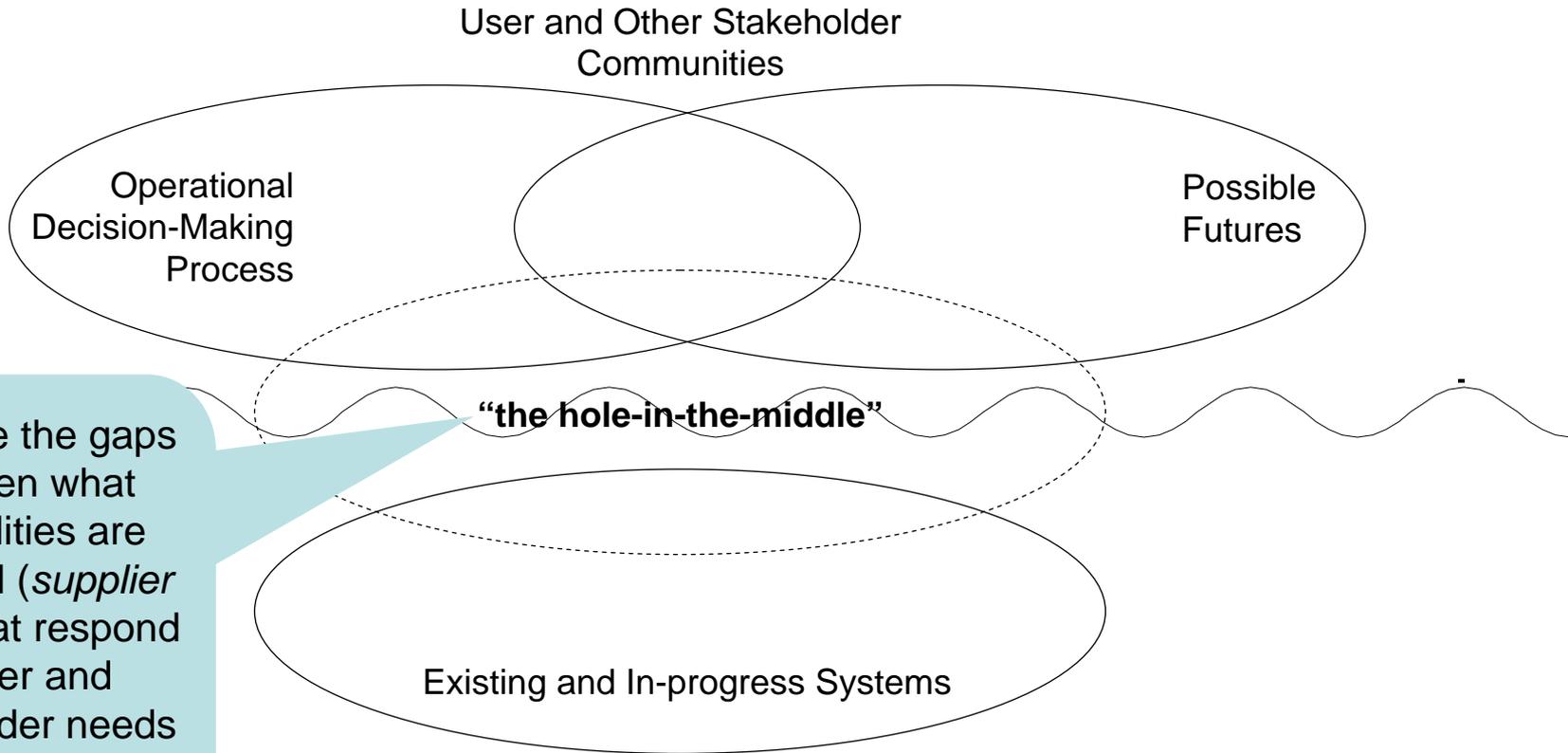
What We Typically Do Today— Formal Acquisition



What is Needed—Concept of “Operational” that Takes a Broader View



In Practice, There is an Increasing Lack of Alignment—a Hole-in-the-Middle



What are the gaps between what capabilities are provided (*supplier push*) that respond to user and stakeholder needs (*operational pull*)?



Key Challenge—How Entities Work Together and Resolve Conflicts

- Number, type, and roles of participants are increasingly diverse, reflecting differing vested interests.
- Scarce resources and the need for concurrent uses make a single decision authority increasingly unlikely.



Single Task System



A single program directs composition
—little potential for conflict



Single Enterprise System



A real or virtual entity directs how multiple entities collaborate to compose multiple programs
—resolves potential conflicts by imposing constraints



Multi-Enterprise System



Multiple real or virtual directing entities making competing demands on SoS
—conflict resolution requires negotiating mutual constraints



A Double Challenge—Diversity of Participants with Turbulent Usage Contexts and Needs

1 - Collaborating effectively across boundaries



Disruption due to addressing the multi-enterprise governance context

Comfort zone for traditional engineering

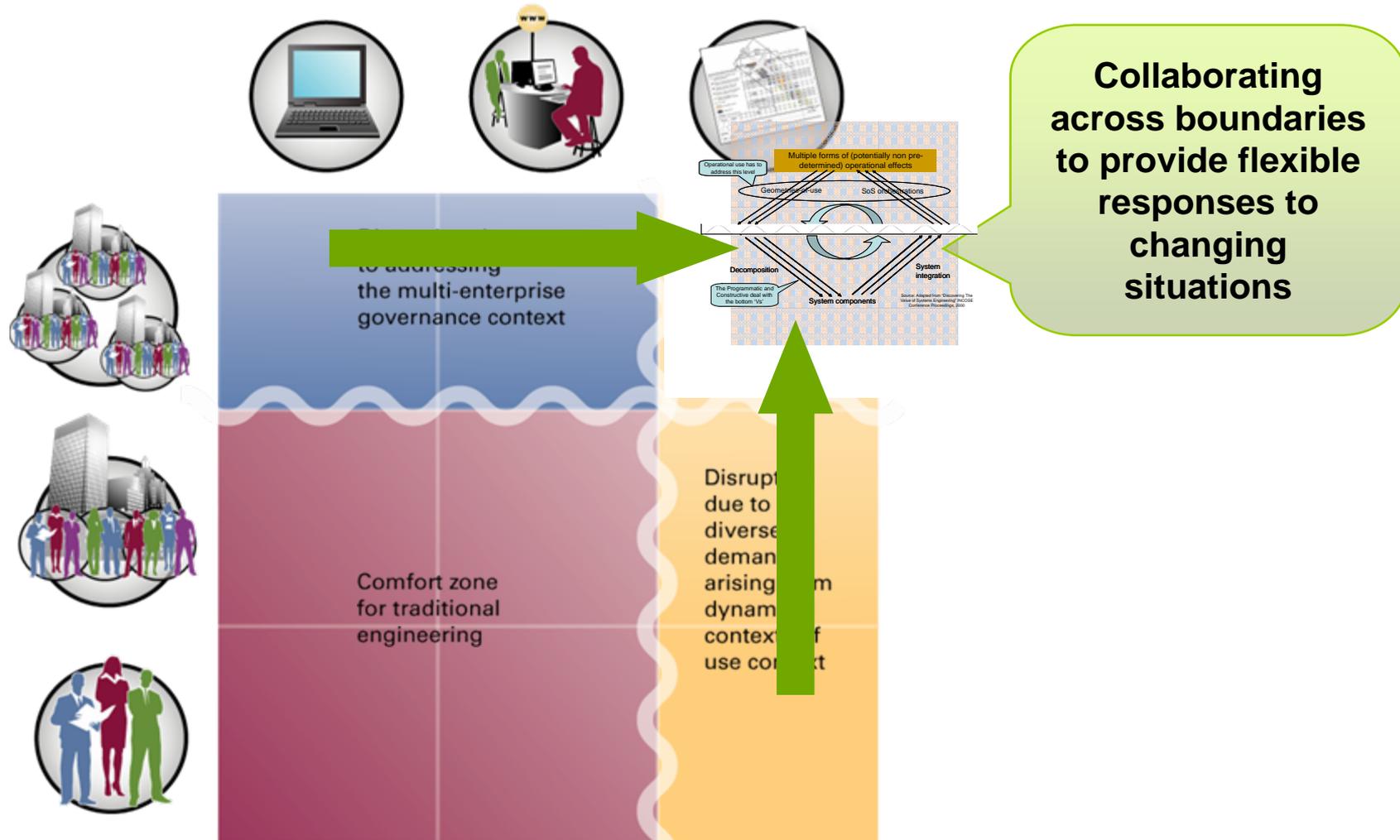
Disruption due to diverse demands arising from dynamic contexts of use context

2- Developing flexible responses to changing situations

Source: *The Double Challenge*, Philip Boxer, 2006; <http://asymmetricdesign.com/archives/16>



What is Needed— Leveraging the Double Challenge



Agility

What does it mean?

- Wikipedia
 - “ability of a firm to sense and respond to business opportunities in order to stay innovative and competitive in a turbulent and quickly changing business environment”
- Microsoft (Future Foundation 2003)
 - “the connected enterprise, and the talent to sense and respond to the outside world”
- Gartner
 - “the ability of an organization to sense environmental change and respond efficiently and effectively to that change” (2006)
 - “the ability to demonstrate flexible, efficient and swift responses to changing circumstances by maximising physical and human resources”

What is an agile organization?

- An **agile** organization (one that demonstrates *agility*) has the capabilities and processes to respond to environmental changes efficiently and effectively
 - Environmental changes can be internal or external; technological, business, or mission; local or global.



Why is Agility Relevant?

Traditionally, software-intensive system success characterized by managing

- System development (primary): cost, risk, schedule
- System functionality (secondary): functional and non-functional behavior

Steady state environments
(i.e., system functionality known and stable)

- Agility in relation to the environment can be ignored
- Can revert to the emphasis on system development rather than on system functionality

Dynamic environments
(i.e., system functionality driven from unanticipated and changing behaviors by end user communities)

- Agility is **the** fundamental driver for implementation and fielding
- Primary emphasis is on functionality demanded



Forms of Agility Required

Type II Agility

- Anticipate demands on the mission
- Anticipate how products or services will be used
- Multiple organizations each with own form of command

Type I Agility

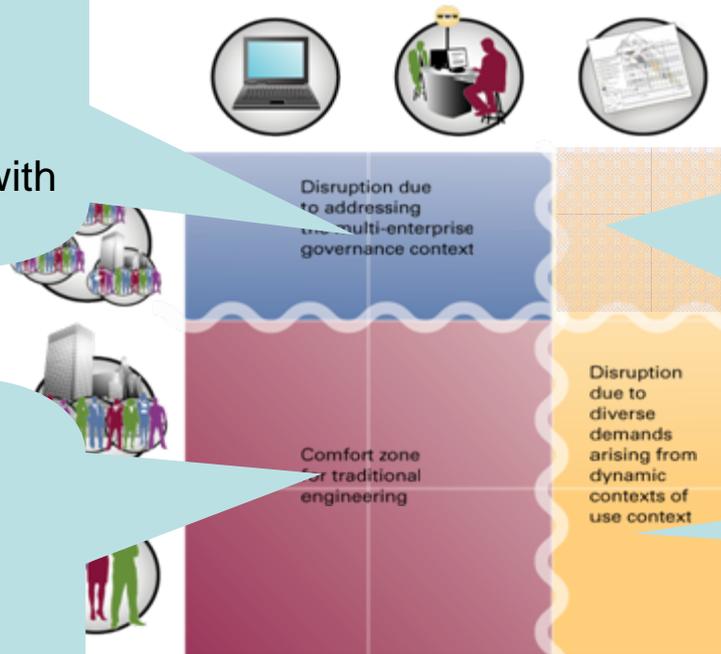
- Anticipate demands on the mission of defending against intrusion
- Anticipate how products or services will be used
- Ensure that managerial entities apply appropriate commands

Type III Agility

- Can't anticipate demands on the mission
- Can't anticipate how products or services will be used
- Multiple organizations each with its own form of command

Type I Agility +

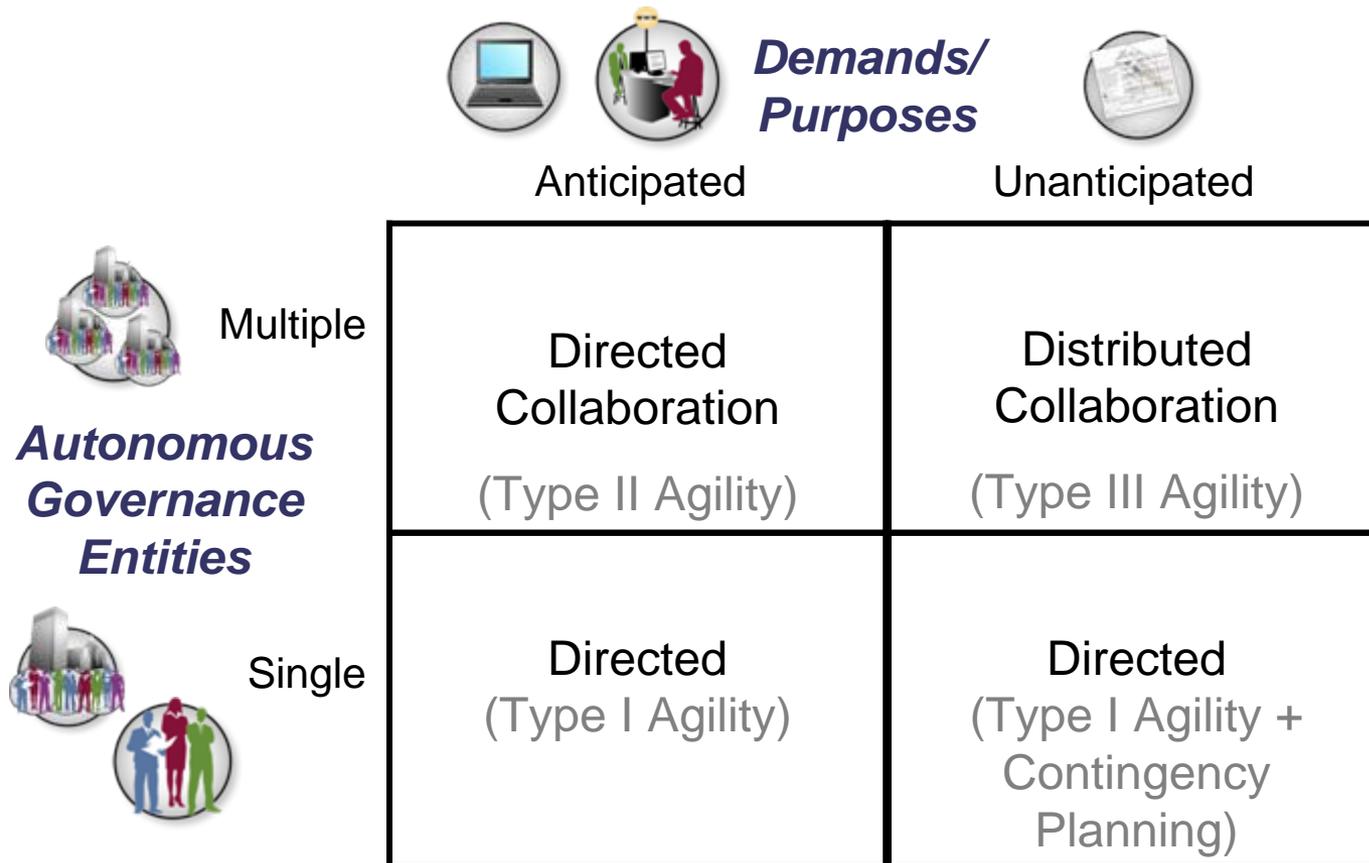
- Contingency planning



Source: *The Three Agilities*, Philip Boxer & Richard Veryard, 2006; <http://asymetricdesign.com/archives/18>



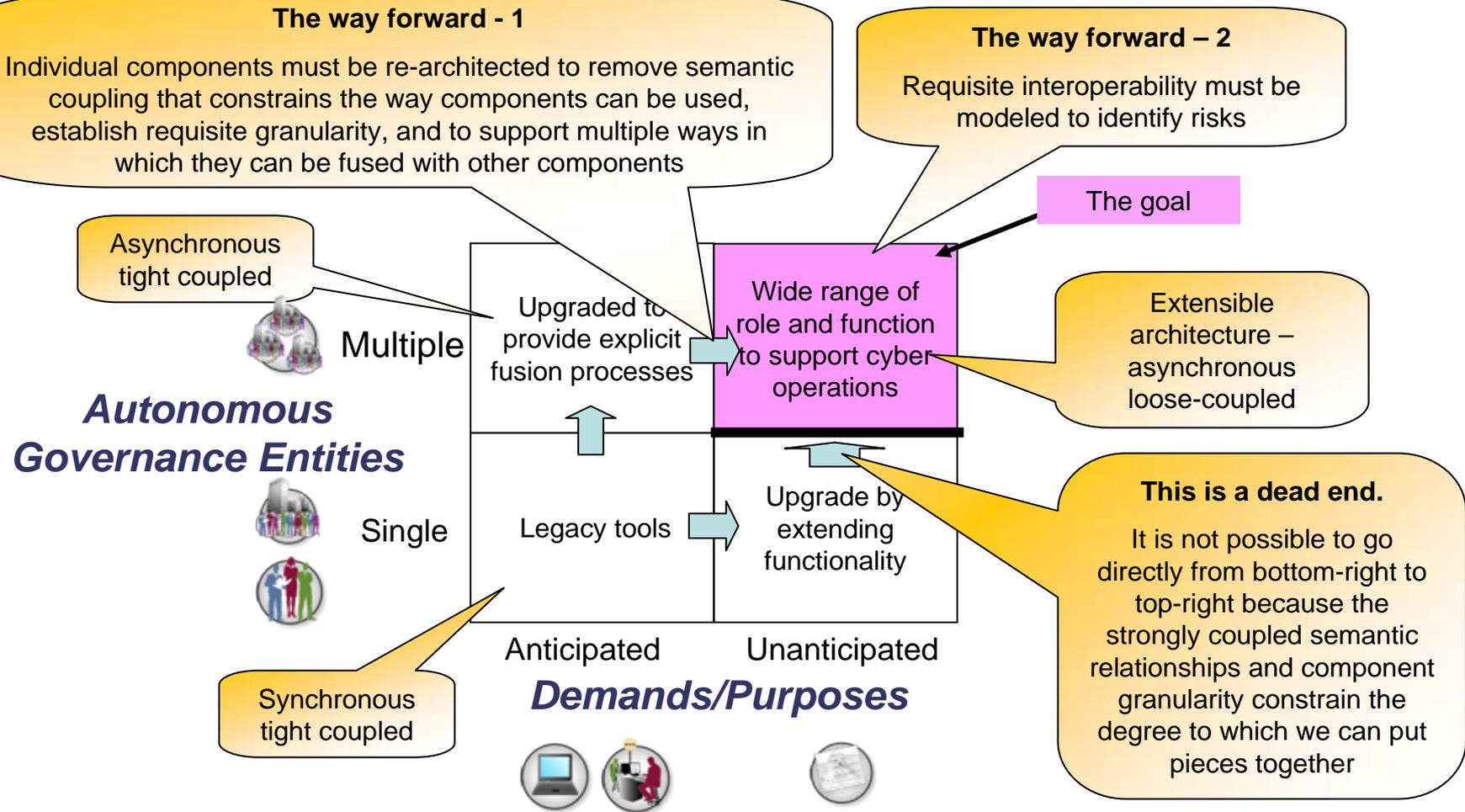
Mapping Agility Types to the Double Challenge



Forms of Collaboration from "Architecting Principles for Systems of Systems", by Mark W. Maier
<http://www.infoed.com/open/papers/systems.htm>



How Do We Get There?



What Does This Mean for Requirements?

Think “junkyard wars”

- Anticipate potential pieces (and the granularity of their functionality)
- Build the pieces to support more expressiveness (to enable semantic interoperability)
- Build on what is already available (stratifying multiple tiers)

This means

- Anticipating the unanticipated by conceiving of potential variety of future scenarios of need
- Focus on corresponding variety of operational needs of users within their effects environments
- Focus on “composable capability” that avoids over-constraining capabilities and requirements



For More Information

Lisa Brownsword, presenter

llb@sei.cmu.edu

Philip Boxer

pboxer@sei.cmu.edu

Dennis Smith

dbs@sei.cmu.edu





Software Engineering Institute

Carnegie Mellon



Software Engineering Institute

Carnegie Mellon

**The Double Challenge in Software
Requirements Management**

© 2007 Carnegie Mellon University