Jo Ann Lane
San Diego State University
September 2018

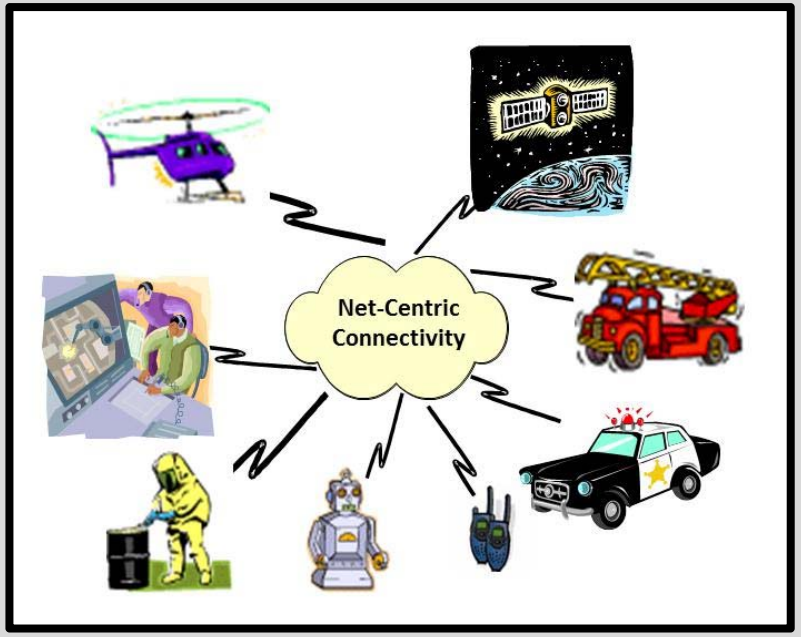
Managing Capability Evolution in Complicated Engineering Environments Through Measurement

From Systems Engineering to SoSE

Primarily Hardware

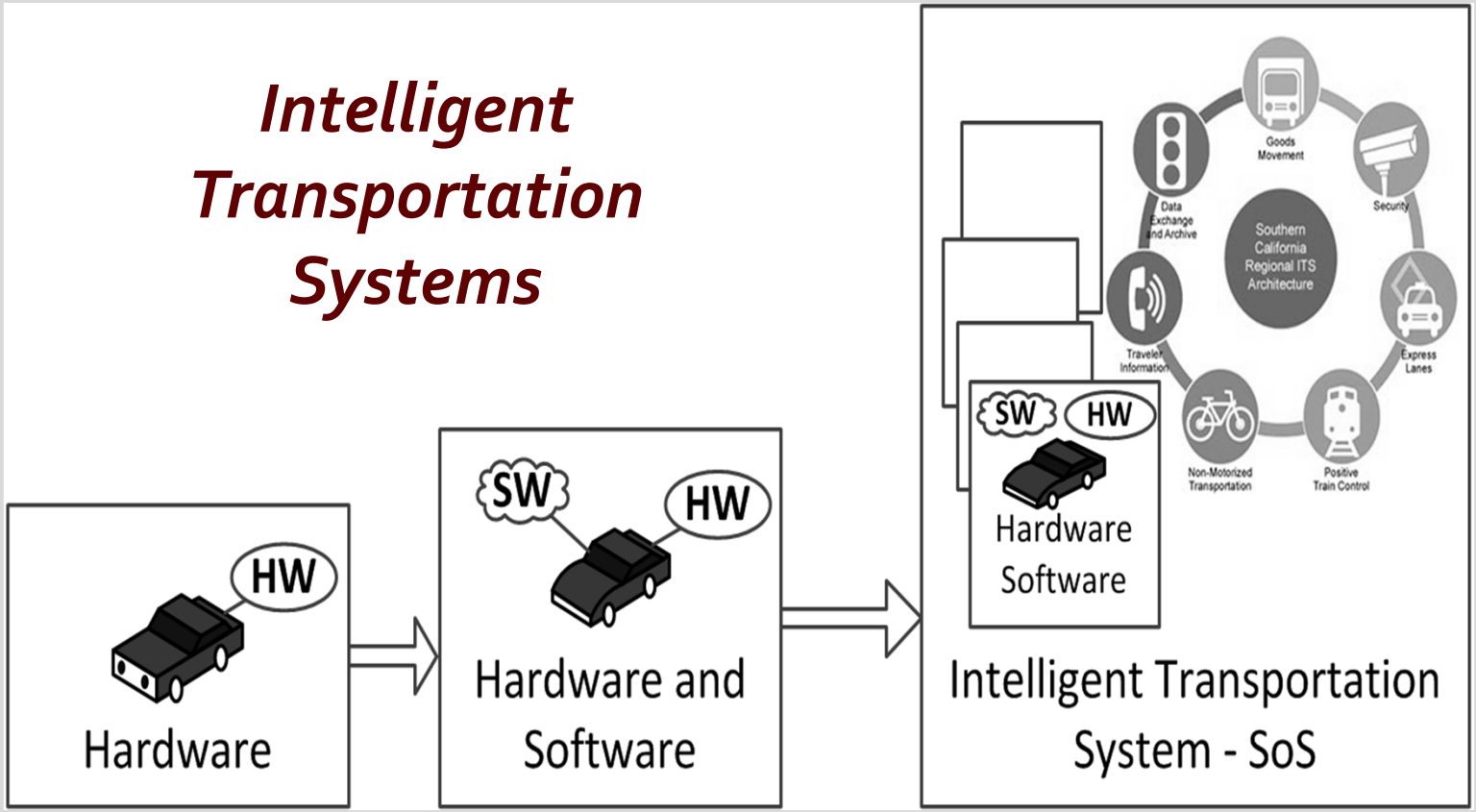
Hardware and Software

Evolution of Intelligent First Responder Systems

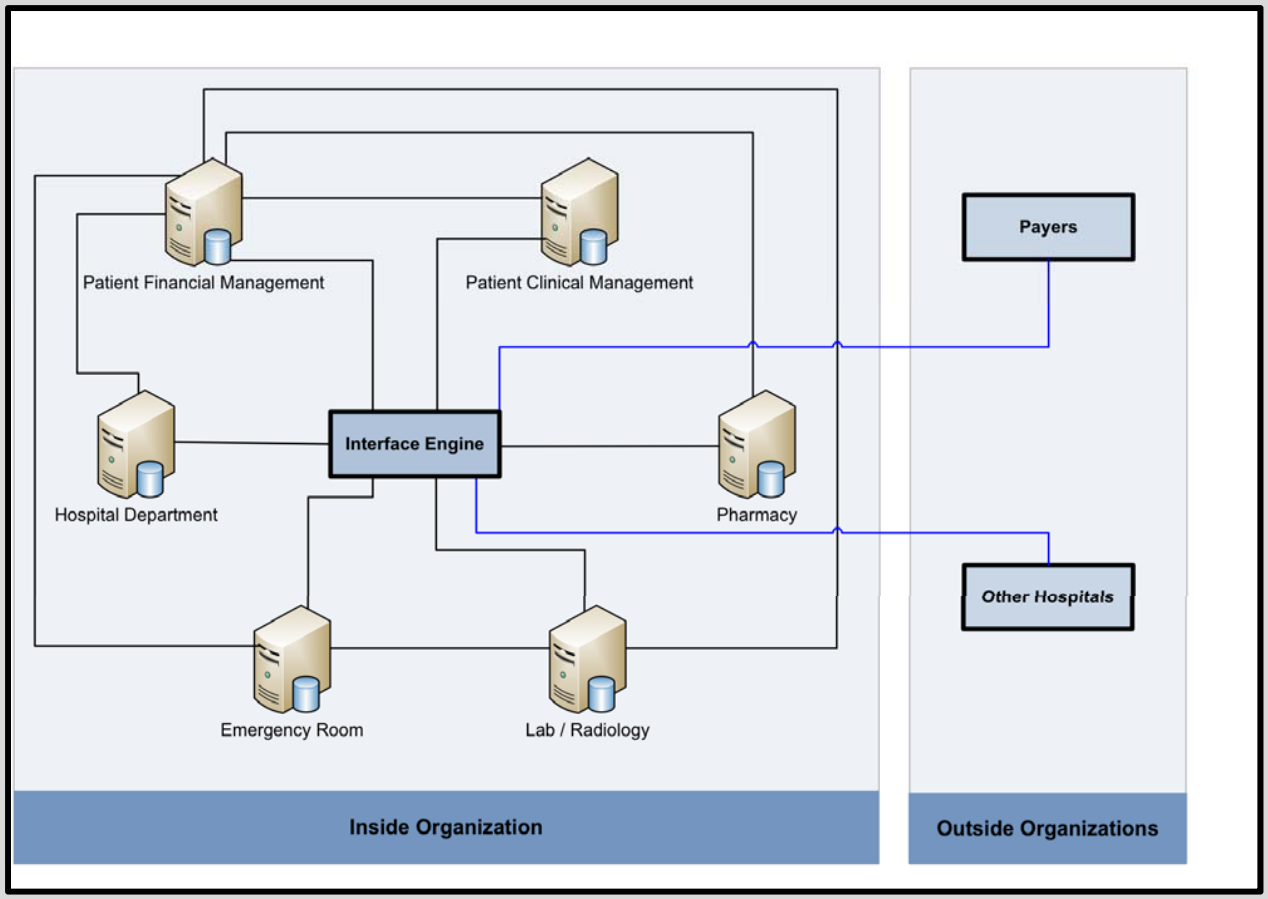


Interoperating Hardware/Software Systems

From Systems Engineering to SoSE



From Systems Engineering to SoSE: Healthcare Setting*



Research evaluated a Service Oriented Architecture (SOA) approach for interconnectivity and interoperability

Findings:

- *Compatible with existing systems*
- *Provided new opportunities to interoperate with other external systems*
- *Improved timeliness of service in ER room (savings ranged from 7% to 30%)*
- *Reduced patient length of stay in ER by 40%*

* Deepa Suri, *Analysis of Service Oriented Architecture in a Hospital Emergency Room Environment*, Computer Science Masters' Thesis, San Diego State University, 2009.

Systems and Software Engineering Tradespace

Tradespace:

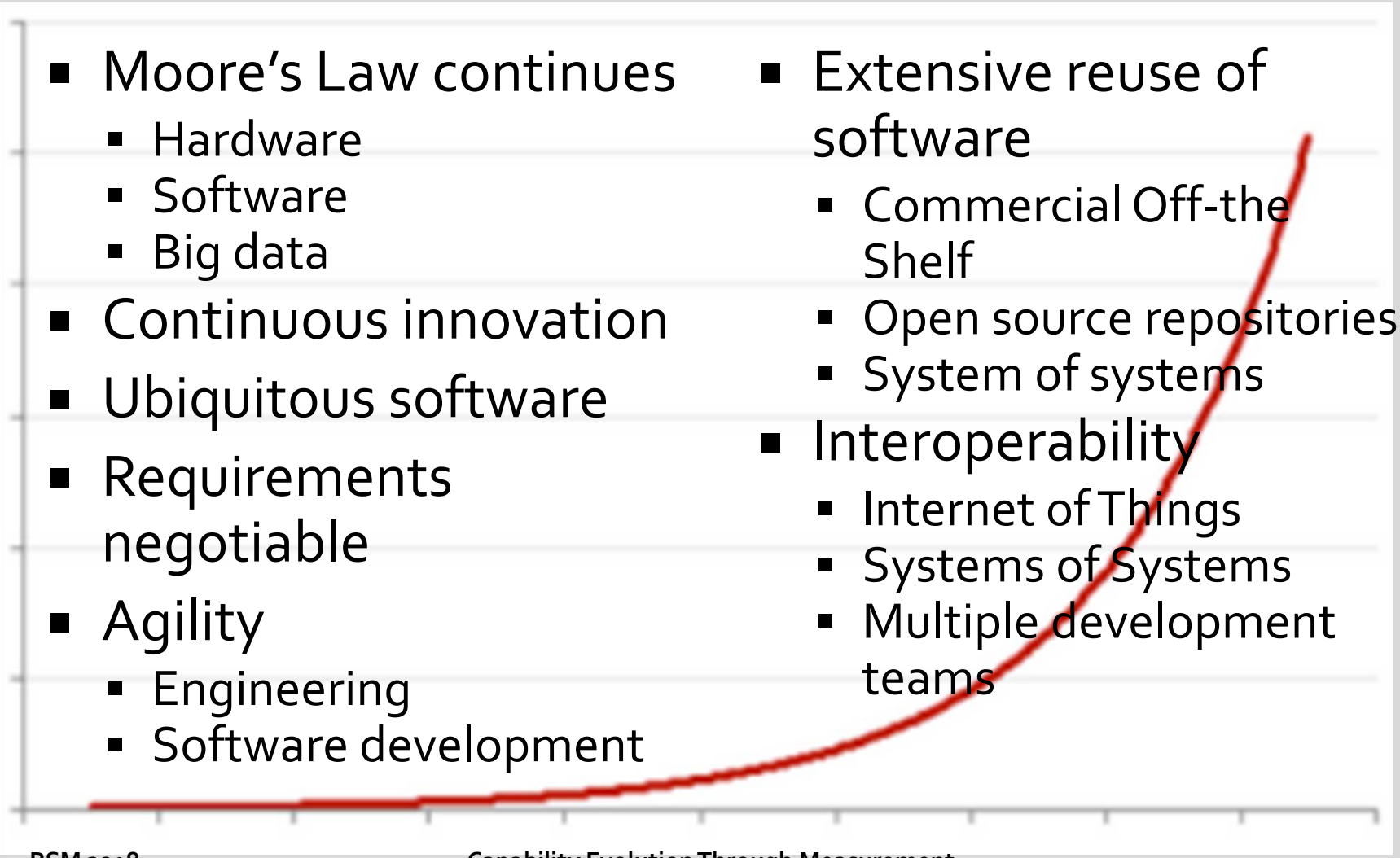
Desired balance:

Typical results:

* Technical Debt: Refers to delayed technical work or rework that is incurred when shortcuts are taken or short-term needs are given precedence over the longer-term objectives. It is the result of intentional decisions that impact the viability of a system and usually incur interest (i.e., additional cost) to eliminate.

Capability Development Trends

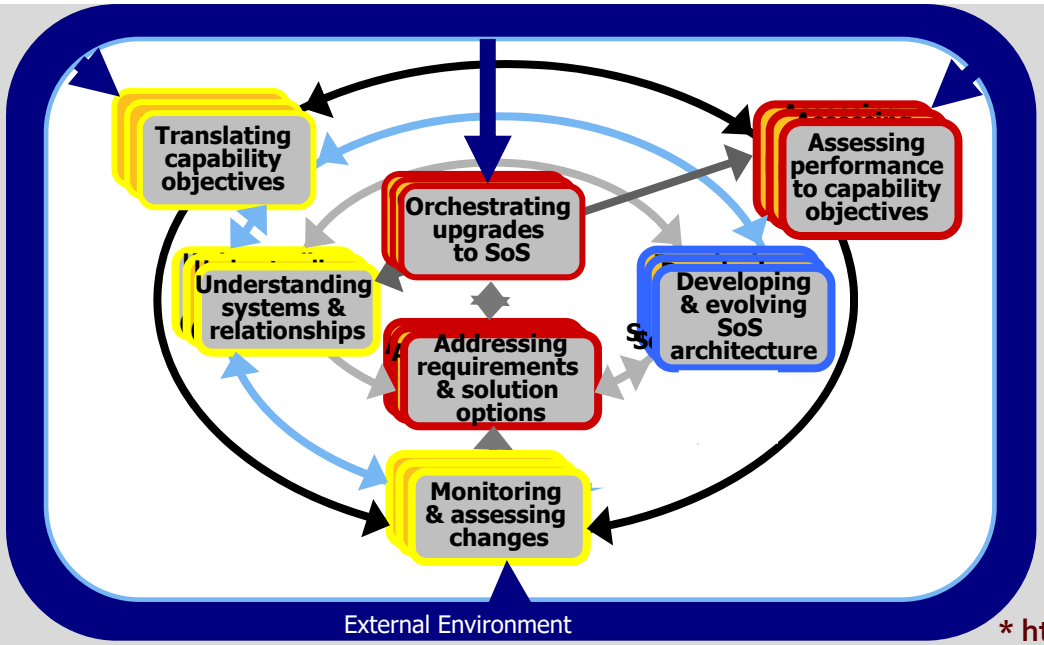
- Moore's Law continues
  - Hardware
  - Software
  - Big data
- Continuous innovation
- Ubiquitous software
- Requirements negotiable
- Agility
  - Engineering
  - Software development
- Extensive reuse of software
  - Commercial Off-the Shelf
  - Open source repositories
  - System of systems
- Interoperability
  - Internet of Things
  - Systems of Systems
  - Multiple development teams



Capability Development Challenges

- Multiple capabilities developed in parallel
- Changing priorities based on world events
- Responding quickly to world events
- Assessment of reuse "goodness"
- Interoperability standards
- Who has their eye on the "capability completion" ball?

Capability Engineering: Agile View



SoSE Guidebook* view based on interviews and analysis of 18 DoD SoSs in various stages:

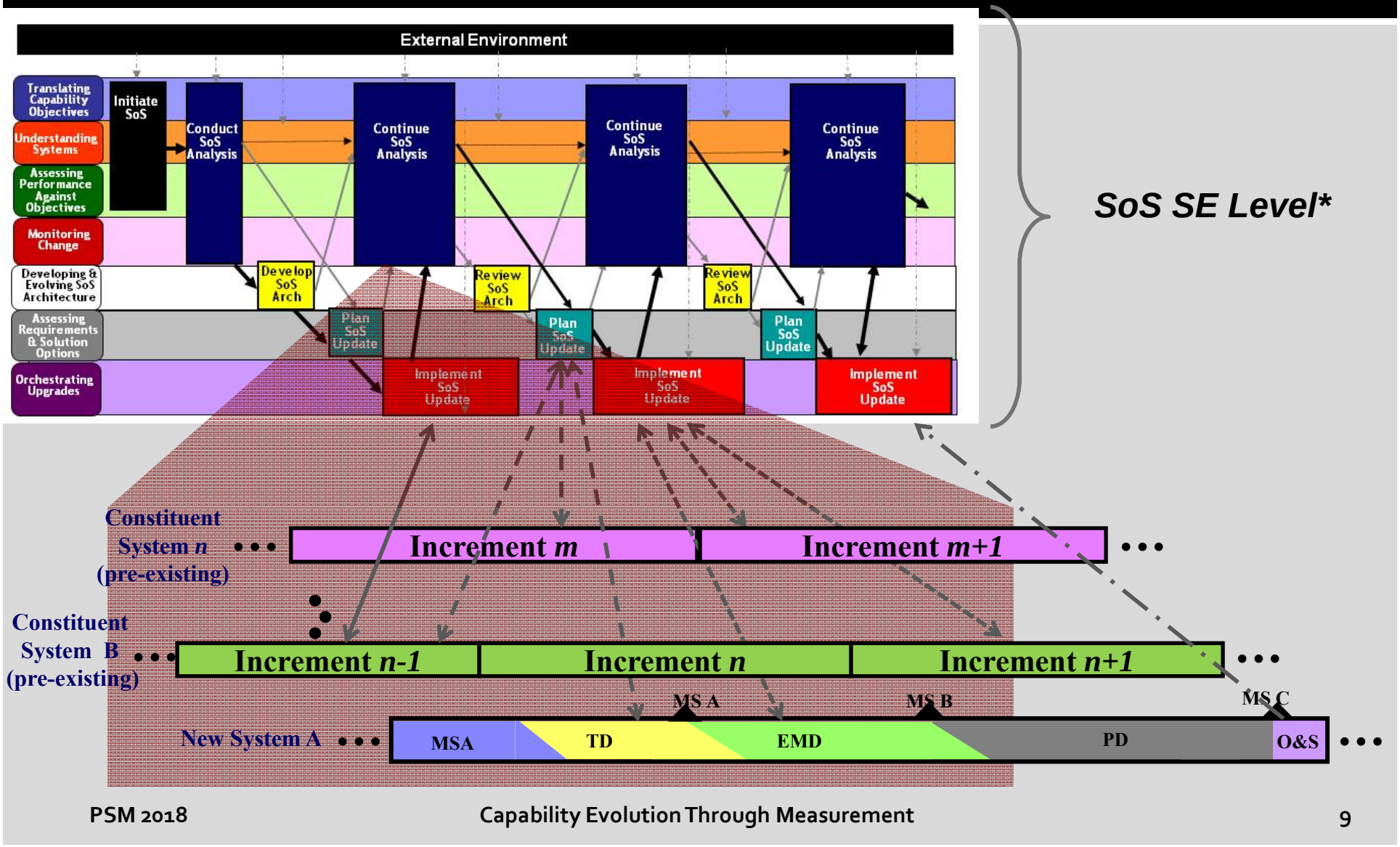
- Communications systems
- Command and control systems
- Integrated combat systems
- Ballistic missile defense systems
- Intelligence information systems
- Space-related systems

* http://www.acq.osd.mil/sse/docs/SE-Guide-for-SoS.pdf

Key challenges

- Focusing CSs on SoS needs and capabilities
- Coordinating development of new capabilities across CSs
- Creating SoS roadmap to guide CS activities
- Testing SoS capabilities in an asynchronous development environment

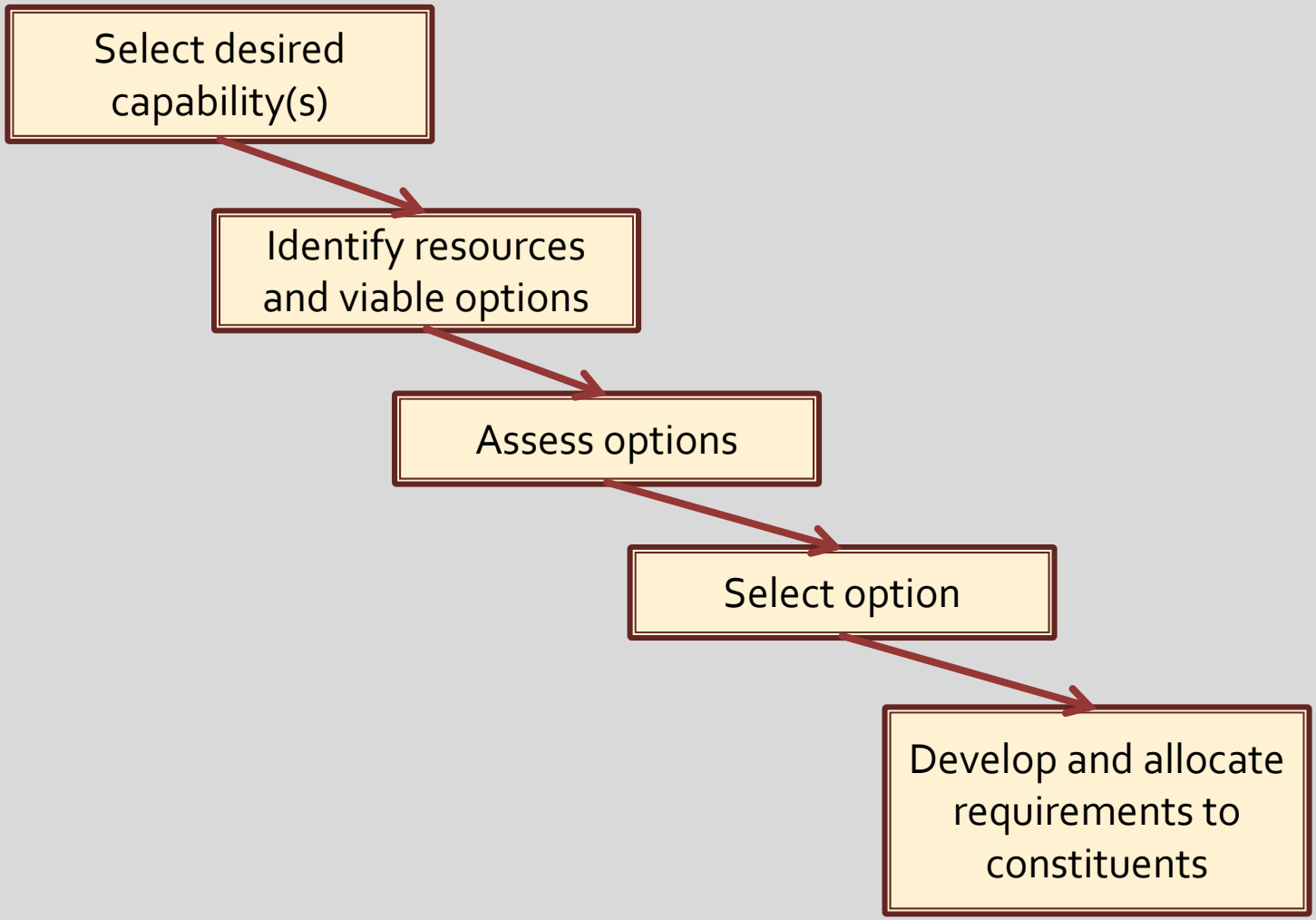
SoSE Synchronization Challenges: Agile View Unwound



Implications for Program Management

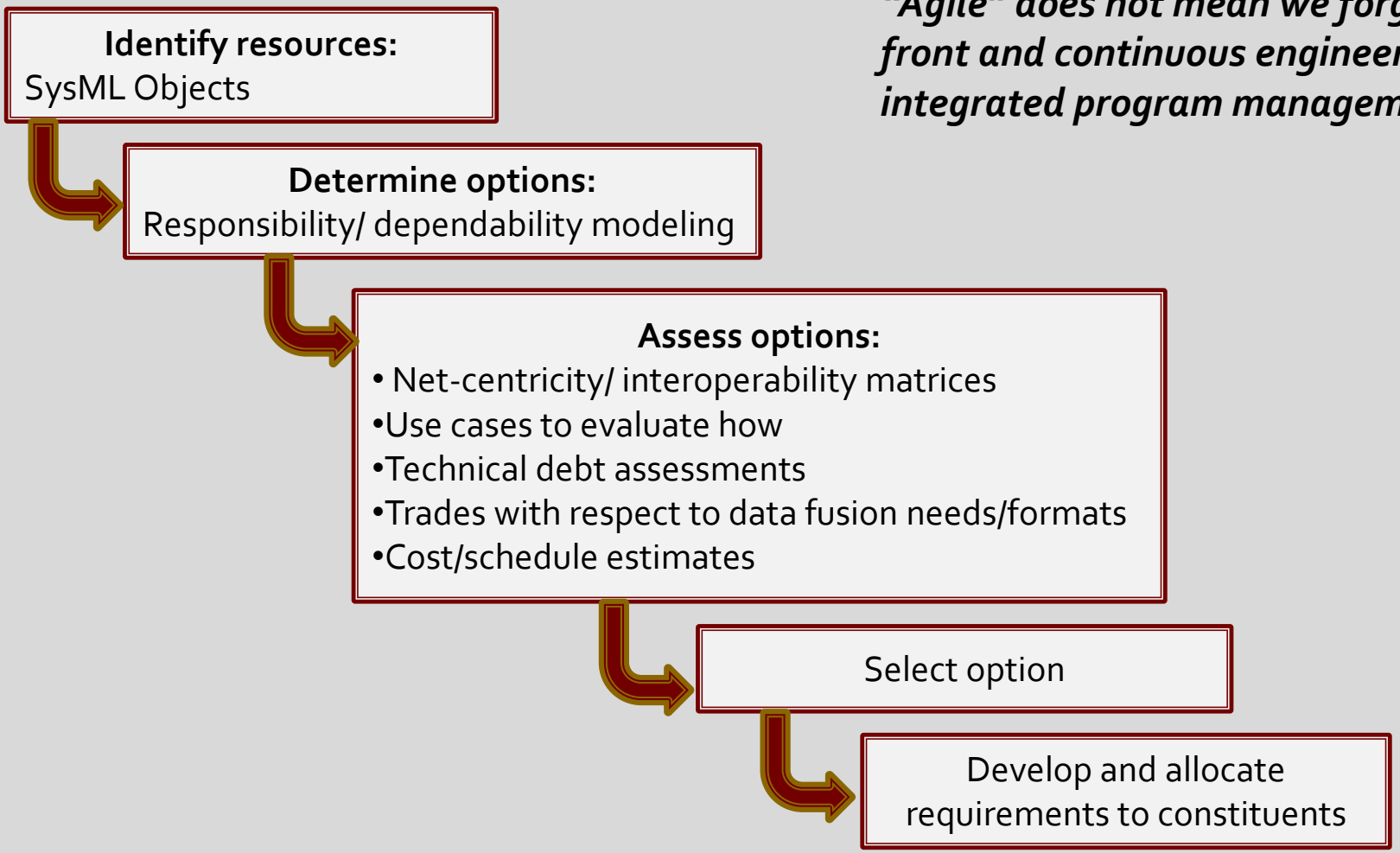
- Basis of estimates for go/no-go decisions
  - Based on key engineering decisions/approaches
  - Assessment of alternatives
  - Risks/probability of success
- Tracking progress
  - Incremental
  - Capability and priority-based
  - Agile
  - Automated
  - Difficult to define traditional "milestones"
  - Releases opportunity-driven for functional capabilities

Capabilities to Requirements



Capabilities Engineering Techniques

"Agile" does not mean we forget up-front and continuous engineering and integrated program management



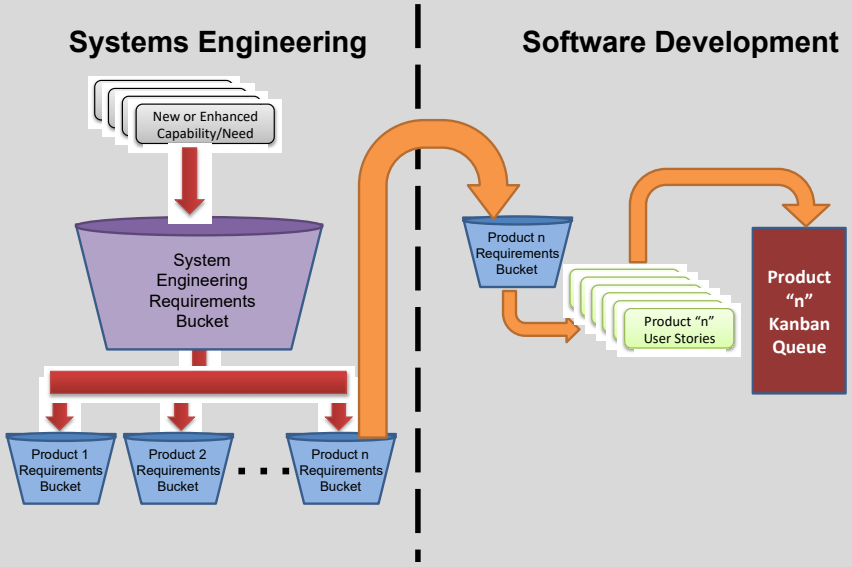
Planning and Estimation

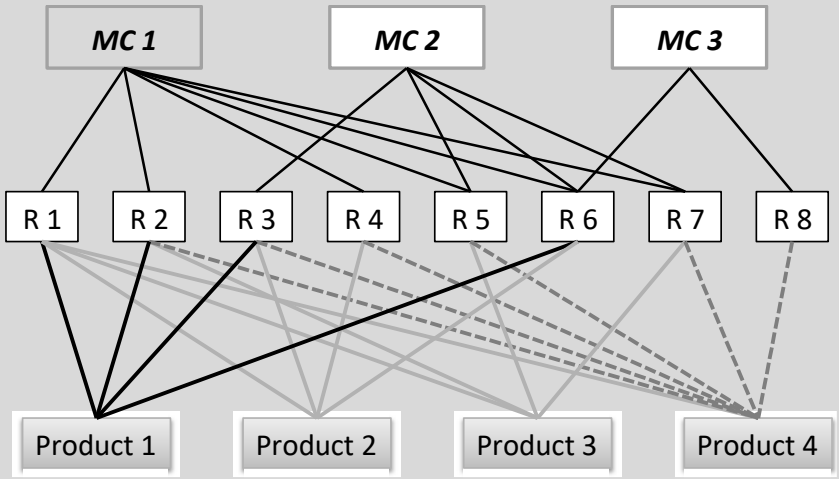
- Initial planning and estimation
  - Based on product, process, personnel characteristics
  - Often done with existing parametric models
  - Often overly optimistic
- Continual assessment and re-planning based on
  - Work completed to date
  - Cadence or velocity of teams
  - Understanding dependencies of tasks—especially between teams

Scheduling: Working Smarter

- Don't reinvent the wheel
  - Reuse existing COTS, open source
  - Understand strengths/limitations of COTS, open source
  - Develop common/reusable solutions
  - Continuous improvement/upgrade
- Understand task dependencies
  - Between hardware and software
  - Between COTS and custom
  - Capability "parts"

Understanding Relationships between Capabilities and System Features

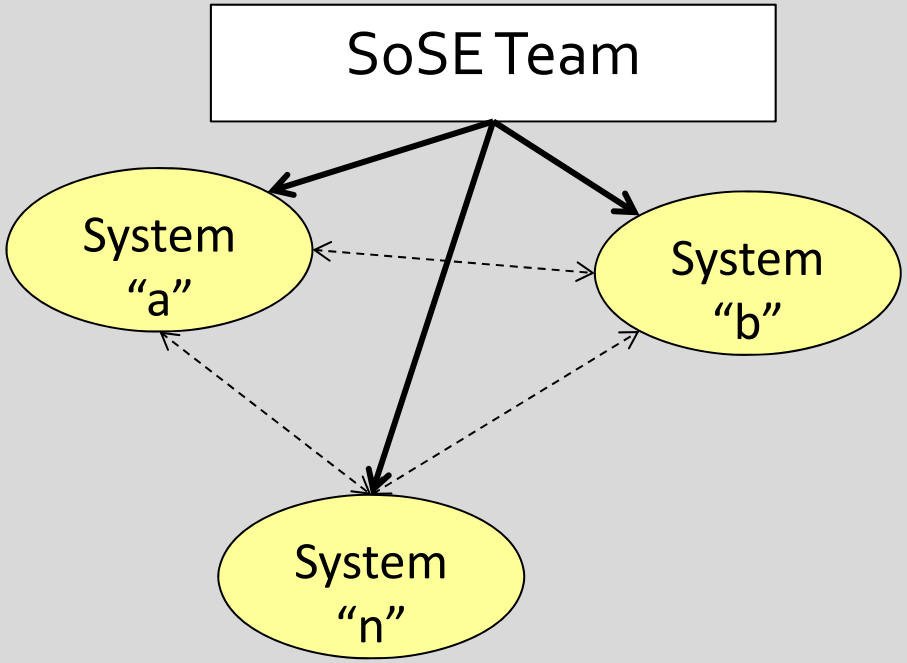




The goal is to not lose sight of:

- All the pieces of a given capability
- How one can reduce total effort by reusing requirements and design approaches between capabilities

SoS Capability Estimation: What to Include?

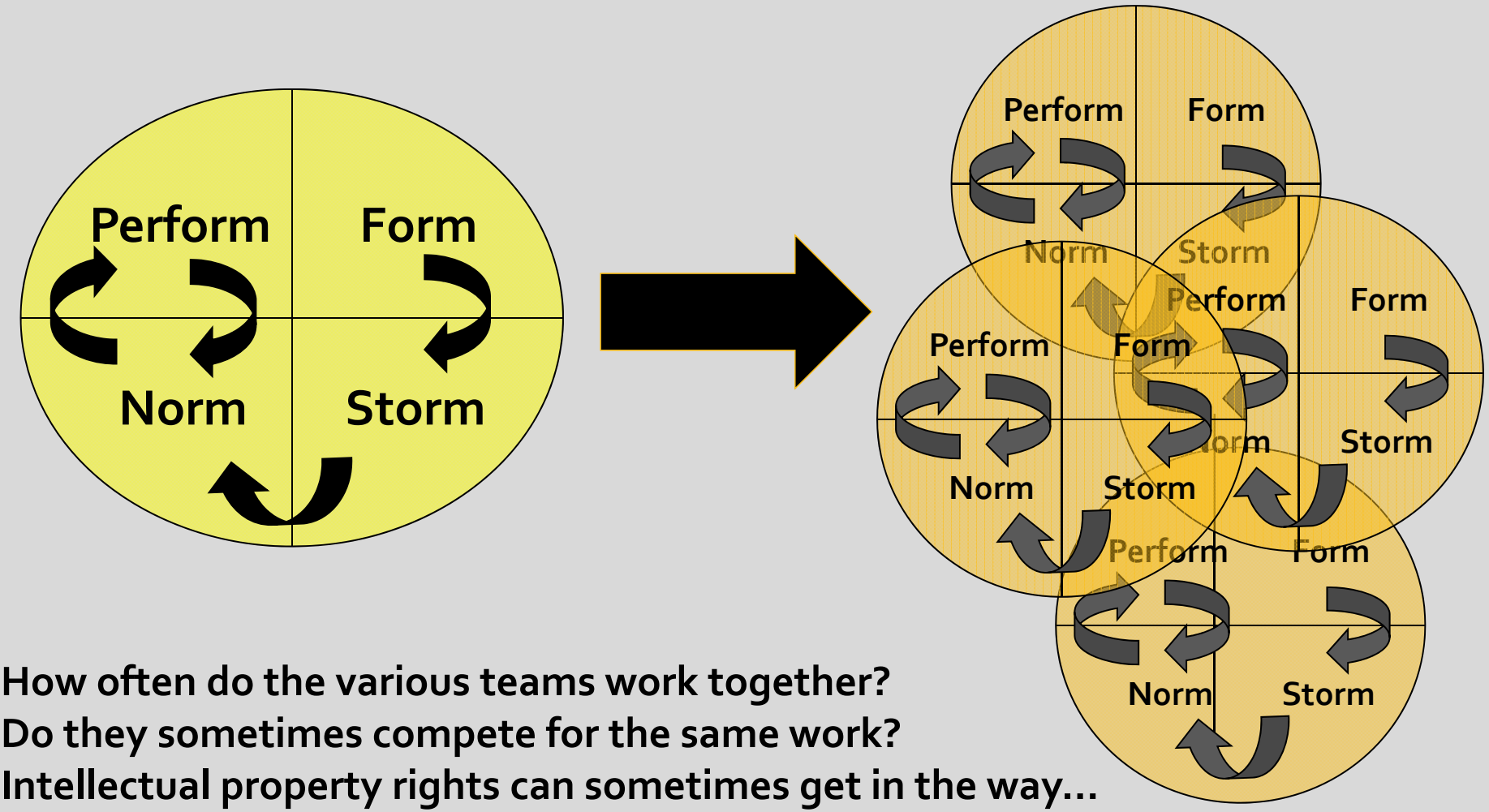


- All SoSE team effort
- SoS-related CS effort
  - Support SoS analyses
  - Engineer/develop SoS capability
- COTS upgrades
- Acquisition of new CSs or COTS

Considerable attention to interoperability between CSs and data flow

Opportunities to reduce effort when combining multiple related capabilities?

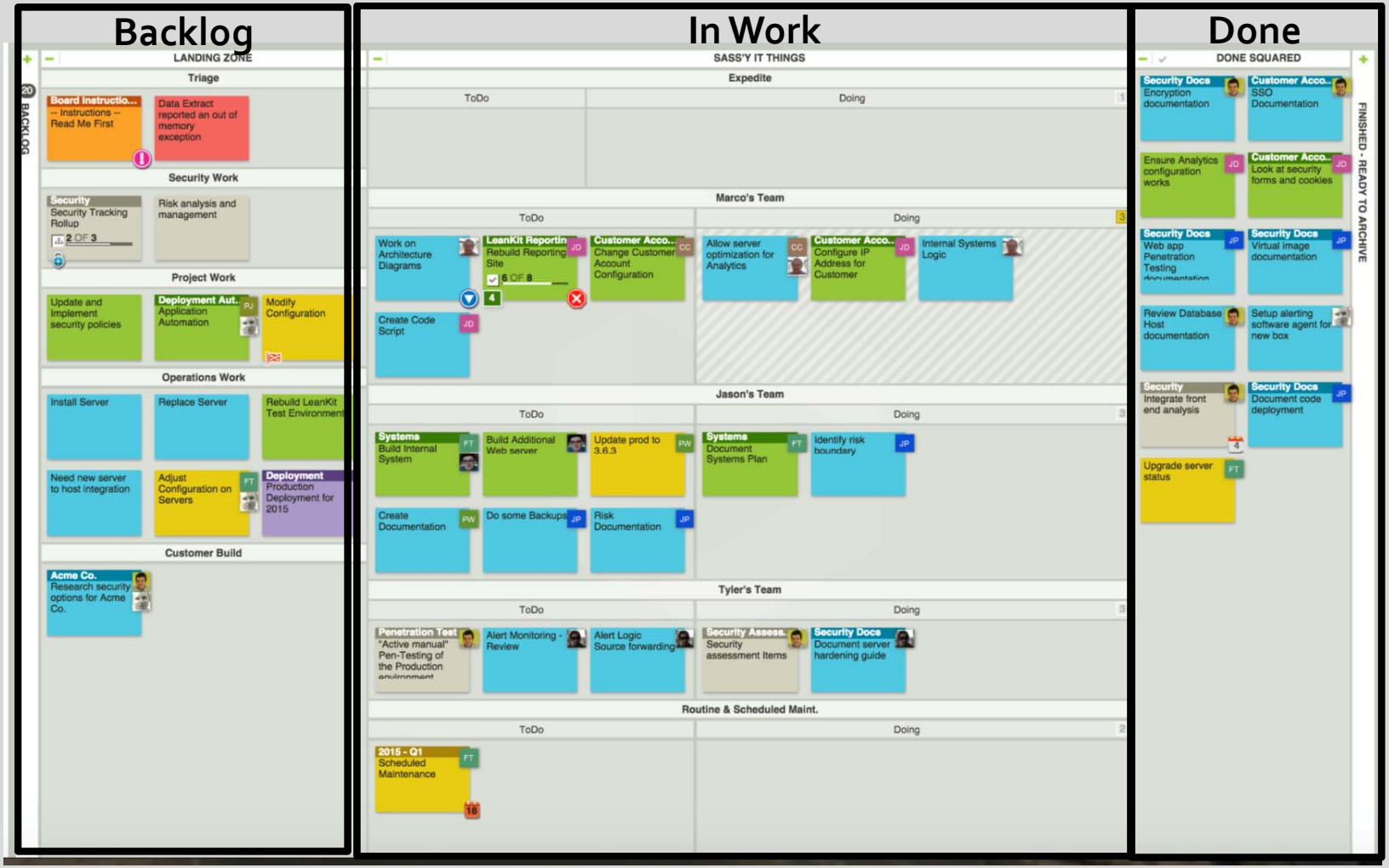
Also Consider Team Organizations



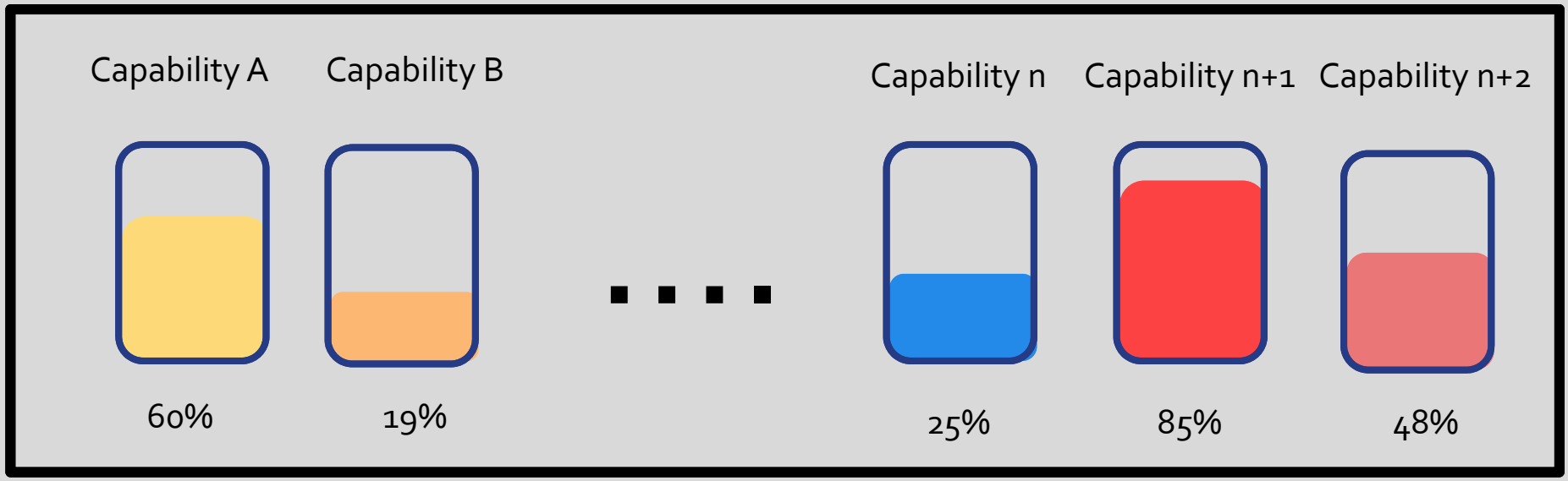
How often do the various teams work together?
Do they sometimes compete for the same work?
Intellectual property rights can sometimes get in the way…

Visibility through Kanban Boards

(Example Kanban board from leankit.com)

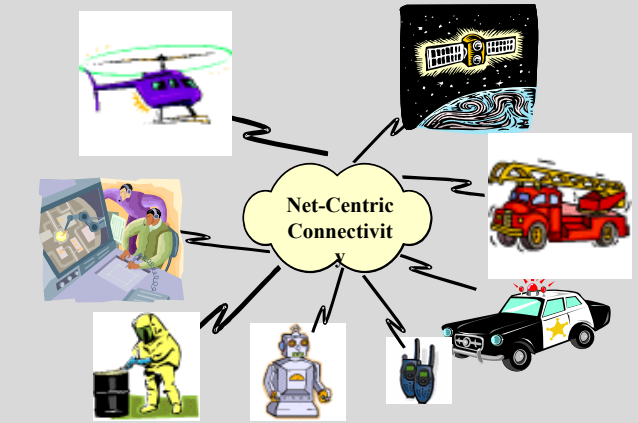


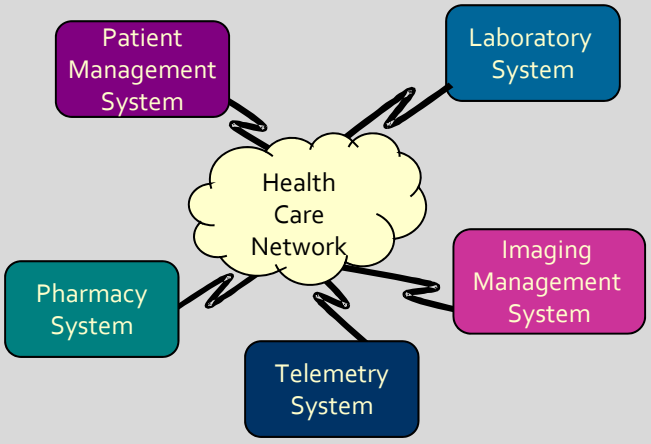
Capability Dashboard

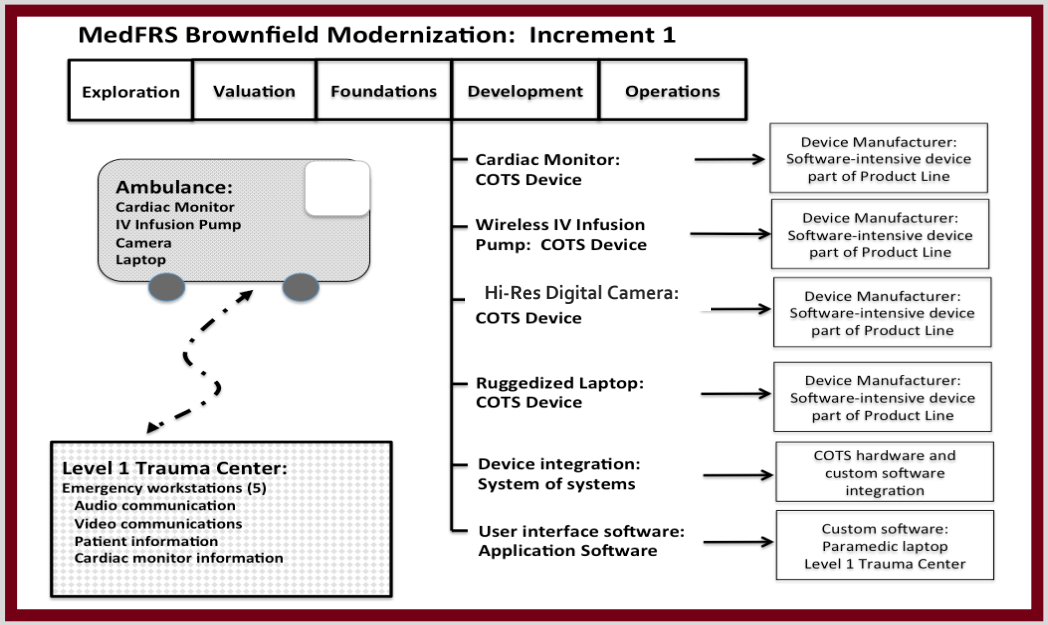


- Based on % of completed Kanban tasks
- Kanban tasks: either 0% or 100% complete
- Can be augmented to show associated
  - Technical debt
  - Integration status
  - Test status

Case Study







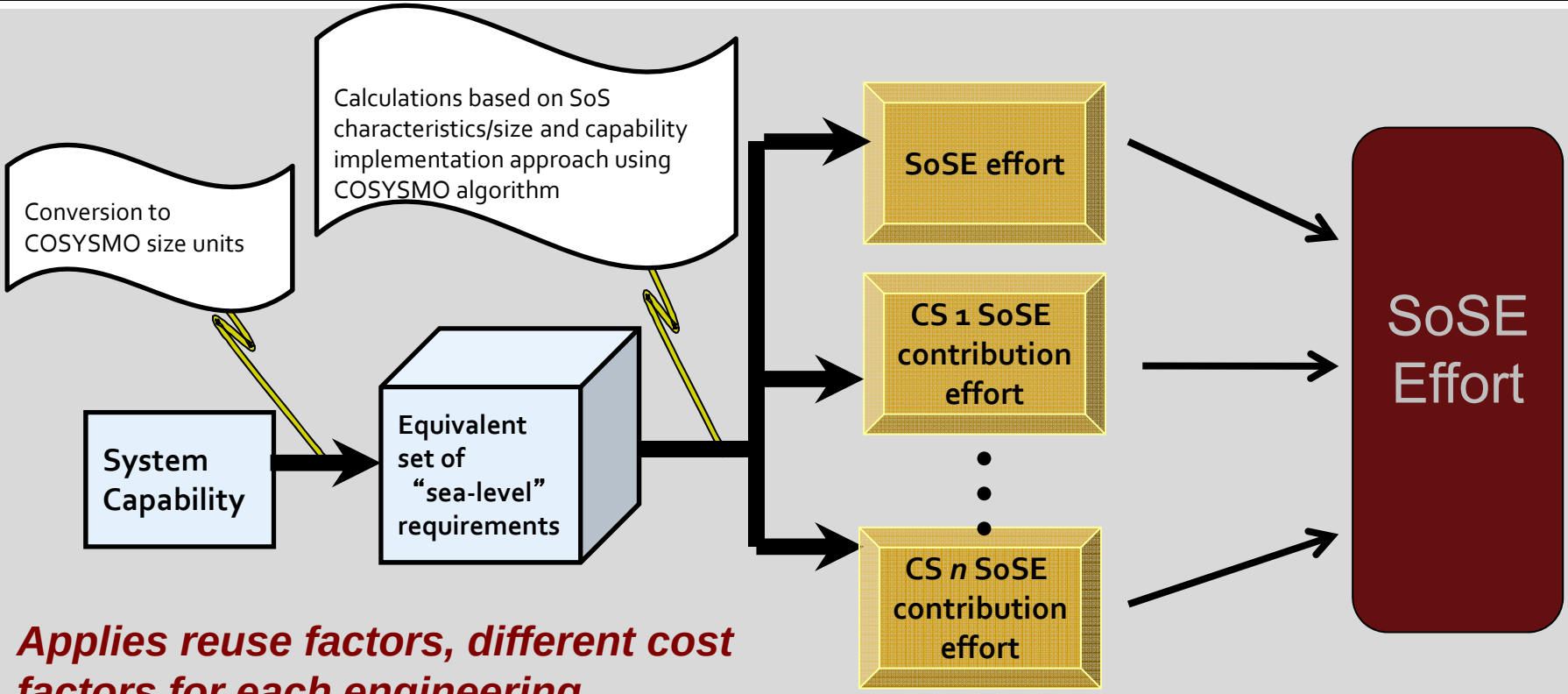
Add telemedicine capability to tie together

- ***First responder SoS***
- ***Hospital SoS***

Estimation Scope/Considerations for SoS Capability Options

- What new or enhanced capabilities are desired?
- Which existing constituent systems are affected?
- What is the current state of each affected system?
- How interoperable are the current constituent systems?
- Are any new systems/components required?

SoSE Effort Cost Estimation



Applies reuse factors, different cost factors for each engineering organization at each system level, and diseconomy of scale for SoS and CS-level requirements implemented in the same upgrade cycle….

J. Lane (2009); Cost model extensions to support systems engineering cost estimation for complex systems and systems of systems. 7th Annual Conference on Systems Engineering Research, Loughborough University, UK.

References for Additional Information

1. B. Boehm, J. Lane, S. Koolmanojwong, R. Turner (2014); *The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software*, Addison-Wesley, ISBN-13: 978-0-80822-6.
2. Department of Defense. 2008. *Systems engineering guide for system of systems,* version 1.0.
3. J. Lane and T. Bohn. 2010. Using SySML to evolve systems of systems.
4. J. Lane and R. Valerdi 2011. System Interoperability Influence of System on Systems Engineering Effort, Proceedings of the Conference on Systems Engineering Research, 14-16 April, Los Angeles, CA.
5. Greenwood, D. and I. Sommerville. 2011. Responsibility modeling for identifying sociotechnical threats to the dependability of coalitions of systems. *Proc. of the 2011 6th International Conference on Systems of Systems Engineering,* Albuquerque, New Mexico.
6. Solano, M. 2011. SoSE architecture principles for net-centric multi-int fusion systems. *Proc. of the 2011 6th International Conference on Systems of Systems Engineering,* Albuquerque, New Mexico.
7. J. Lane (2009); Cost model extensions to support systems engineering cost estimation for complex systems and systems of systems. 7th Annual Conference on Systems Engineering Research, Loughborough University, UK.

Backup Slides

SoS/SoSE Challenges: Implications for Capability Sizing and Cost Estimation

- Estimation focus
  - Forward looking flow of enhancements (multiple increments)
  - Based on alternatives
    - Maintenance/upgrade of existing capabilities
    - Desired new capabilities
    - Feature – capability relationships and priorities
- Alternative analysis/selection impacts
  - Which systems are "touched"
  - Whether new systems are procured or developed
  - When to retire/replace aging systems/components
  - Schedule for capability availability
  - Effort

Considerations for SoS Capability Options

- Which constituent systems can support desired capability/capability enhancement?
- How responsible/dependable is system owner?
- How interoperable is constituent system with rest of constituent systems?
- Does the constituent system have engineering/development resources available?
- What is the most likely increment in which desired changes can be incorporated?

Considerations for SoS Capability Options

- What technical debt must be addressed (by option)
- What underlying performance problems must be addressed (by option)
- What changes need to made to existing systems?
- What new components/systems need to be incorporated?
- How is the user going to learn/manage all of this additional capability/complexity?