Jo Ann Lane
San Diego Chapter of INCOSE
23 July 2016

Planning for SoS Evolution

Goal of Tutorial and Bottom Line Up Front

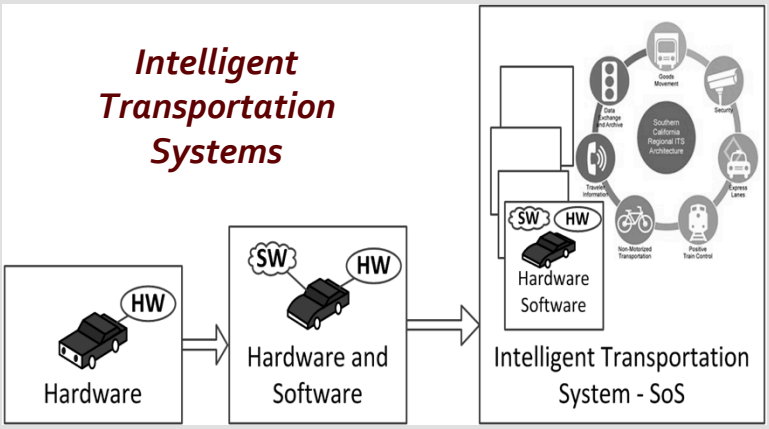
- Identify SoS evolution planning focus areas and challenges critical to the success of SoS
- To plan for SoS evolution, you need to know
  - What is the current state of the SoS
  - What are the desired SoS capabilities
  - What is the priority associated with each SoS capability
  - What currently exists/what do we have to work with
  - The desired/required schedule
  - What it will cost

Overview

- ***SoS and SoSE basics***
- Getting started with SoSE
- Incremental commitment and evolution focus
- SoS capability to requirements engineering
- Keeping it lean

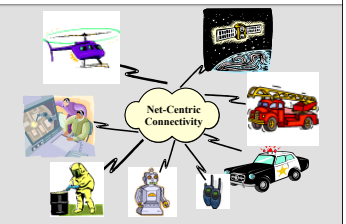
Time permitting:
- Case study analyses

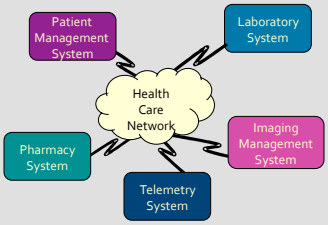
From SE to SoSE



What is a "System of Systems"?

- Very large systems using a framework or architecture to integrate constituent systems (CSs)
- Exhibits emergent behavior not otherwise achievable by CSs
- SoS CSs
  - Independently developed and managed
  - New or existing systems in various stages of development/evolution
  - May include a significant number of COTS products
  - Have their own purpose
  - Can dynamically come and go from SoS
- Typical domains
  - Military/Crisis Response: Dynamic communications infrastructure
  - Business: Enterprise-wide and cross-enterprise integrations





Based on Mark Maier's SoS definition [Maier, 1998]

Related to SoS: Internet of Things (IoT)

From Wikipedia:

- *The internet of things (IoT) is the network of physical devices, vehicles, buildings and other items—embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data.[1]…*
- *… when IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure.*

[1] "Internet of Things Global Standards Initiative". ITU. Retrieved 26 June 2015.

What is SoS Engineering (SoSE)?

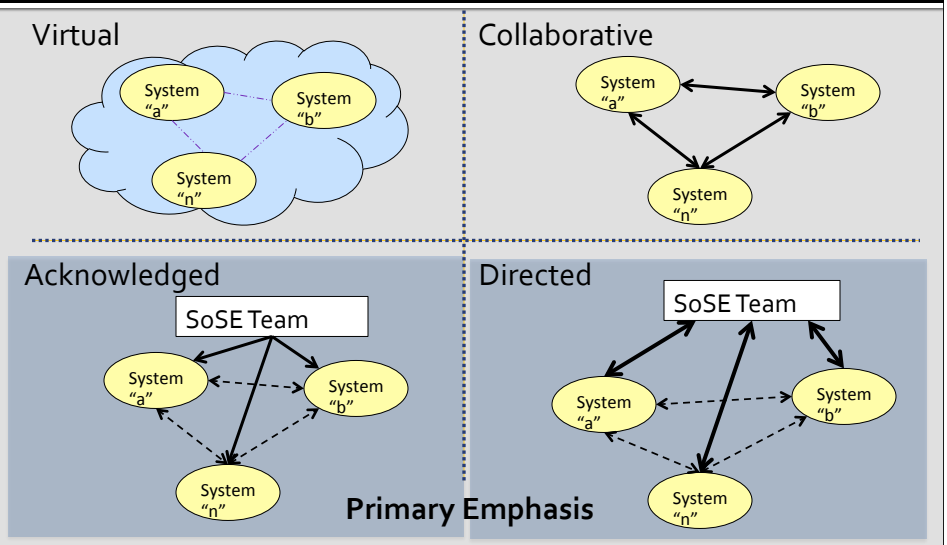
DoD *Defense Acquisition Guidebook*, Chapter 4 "System of Systems Engineering"(2004):

- SoS systems engineering *deals with planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into an SoS capability greater than the sum of the constituent parts.*

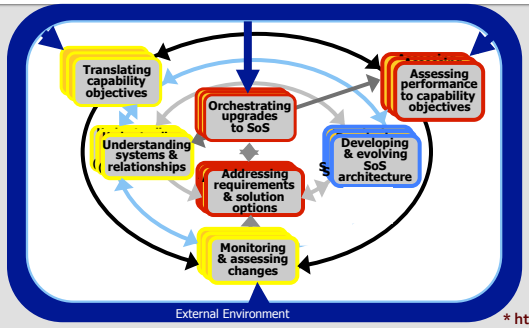
DoD *Systems Engineering Guide for Systems of Systems* (2008) adds to the above:

- *Consistent with the DoD transformation vision and enabling net-centric operations (NCO), SoS may deliver capabilities by combining multiple collaborative and autonomous-yet-interacting systems. The mix of systems may include existing, partially developed, and yet-to-be-designed independent systems.*

Types of SoS: Organizational Structures



SoSE Activities and Challenges for "Acknowledged" SoS



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**

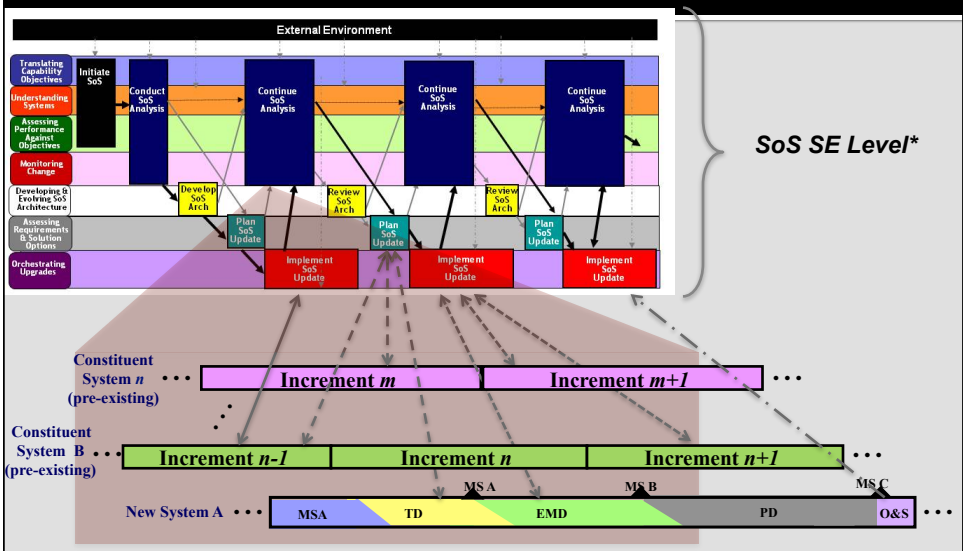
*** http://www.acq.osd.mil/sse/docs/SE-Guide-for-SoS.pdf**

- 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 Core Element Description

- Translating Capability Objectives
  - Starts with an SoS need or new capability
  - Works to understand new capability and alternatives for providing it
- Understanding Systems and Their Relationships
  - Collects and maintains information about current state of the SoS and its CSs
- Assessing Performance to Capability Objectives
  - Evaluation of current performance and how performance meets current and future needs
- Developing/Evolving SoS Architecture
  - Evaluation of existing SoS architecture and identification of alternatives to mitigate limitations and improve performance
- Monitoring and Assessing Changes
  - Monitoring of CS non-SoS changes
- Addressing Requirements and Solution Options
  - Evaluation/prioritization of SoS reqs
  - Evaluation of solution options and selection of option
- Orchestrating Upgrades
  - Oversight activity to monitor progress of the CS SoS capability upgrades and mitigate obstacles

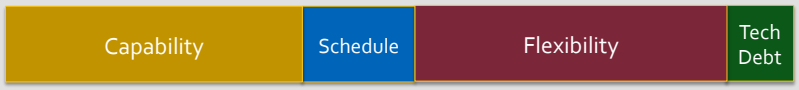
SoSE Synchronization Challenges



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.

© USC CSSE

Overview

- SoS and SoSE basics
- ***Getting started with SoSE***
- Incremental commitment and evolution focus
- SoS capability to requirements engineering
- Keeping it lean

Time permitting:
- Case study analyses

Typical Starting Points

- Existing set of stove-pipe systems that need to interoperate
  - Examples
    - Multiple systems across multiple departments, often with overlapping data/features/capabilities (e.g., healthcare systems)
    - Multiple first responder systems that must collaborate/ interoperate during major crises
- Existing collaborative SoS
  - Increasing in importance
  - Increasing challenges requiring additional coordination or oversight
  - Not working well
- Desire to create a new SoS (*not common*)
  - Example: Future Combat Systems (FCS)

Setting up for SoSE Success

- Recognize that engineering an SoS is different than single system engineering
- Going fast can mean going slow at first
- Make sure you have a strong credible champion
- The "bigger and messier" the SoS, the more important clear roles and responsibilities become

Understand the Objectives

- As you get started, be sure you are grounded in the operational objective which is driving the SoS
- Is there an accepted concept of operations, scenario set or set of mission threads?
- Is there data to support the need for attention in this area?
- Who are the big players in the capability area?

Create an SoS Identity

- Identify motivation for the creation of the SoS
- Determine authority for establishing the SoS
- Formal paper work issued to establish the SoS?
- Identify "traceability" to a legitimate need to establish the 'identity of the SoS'
- Define constituent system roles in the SoS

Identify Systems in Current SoS

- Need to start early to understand systems that impact your objective
- Typically you want to try to identify everything that can impact the SoS results
- Initially, best to focus on a core set of systems

Understand the Systems

- Focus on selected systems and use available data
- Establish relationship with the system SEs… and don't alienate them
- Identify how the programs currently work together technically to support
  - Current SoS capabilities
  - Proposed SoS capability objectives
- Understand the broader organizational relationships and contexts

Establish SoS Programmatic Visibility

- Provides "legitimacy", authority, and responsibility for the SoS
- Typically done by creating program element budget/funding/schedule for SoSE activities
  - Separate from the systems
  - Associated with multiple-system SoS actions
  - Tracks/incentivizes progress/success at the SoS level
    - Supports evidence-based decision-making at SoS level
    - Tied to a capability
    - Addressing a critical performance parameter
    - Based on collaboration/coordination

Establish SoS Leadership

- Eventually need to address management and governance issues
- Key SoS leadership principles
  - Enable allegiance to the broader capability
  - Recruit highly qualified and motivated program management and SE talent
  - Stay focused on the SoS capabilities and performance
  - SoS program management should report to as high an authority as possible

Develop Forward-Looking Strategy

- What if nothing is done, what is likely to happen?
- What kinds of changes are needed in the core systems to make an impact?
- What are the next opportunities to influence these changes?
- Identify the key parameters that should drive your strategy

Be mindful of SoSE justification, stakeholder goals, value of capabilities, and authority/responsibility roles

Overview

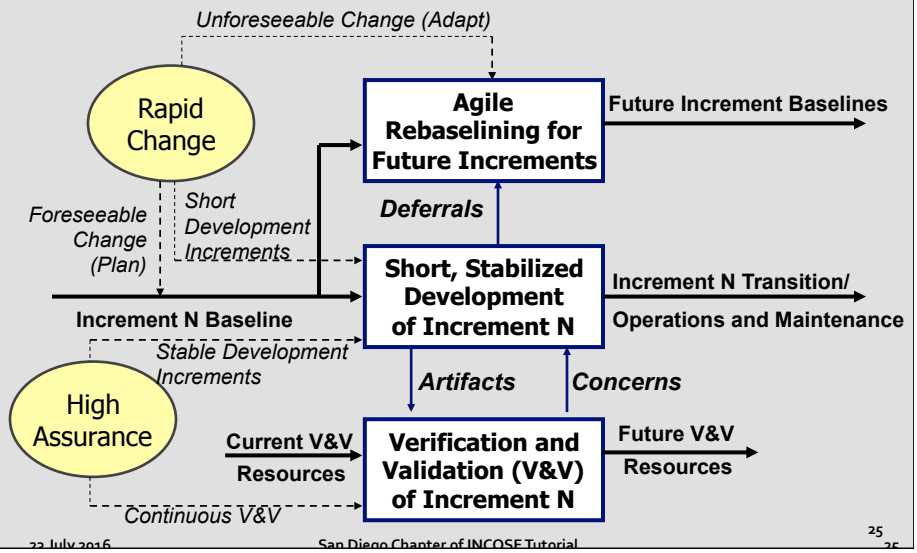
- SoS and SoSE basics
- Getting started with SoSE
- ***Incremental commitment and evolution focus***
- SoS capability to requirements engineering
- Keeping it lean

Time permitting:
- Case study analyses

Incremental Commitment Spiral Model (ICSM) Key Principles

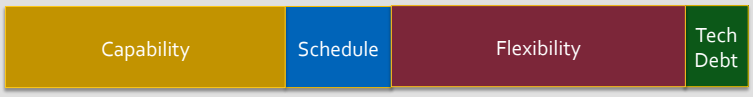
- Stakeholder value-based guidance
  - Identify and know your success-critical stakeholders
  - Sets priorities based on stakeholder value
- Incremental commitment and accountability
  - Bases commitments on knowledge
  - Two-way accountability between stakeholders and developers with respect to commitments
- Concurrent system engineering
  - Strength from agile/lean communities that avoids invalid assumptions, avoids hard-to-undo early commitments, and minimizes rework
- Evidence and risk-driven decisions
  - Results in plans based on knowledge
  - Avoids invalid assumptions and minimizes rework
  - Avoids investment in impractical or overly risky system development efforts

ICSM: Increment View



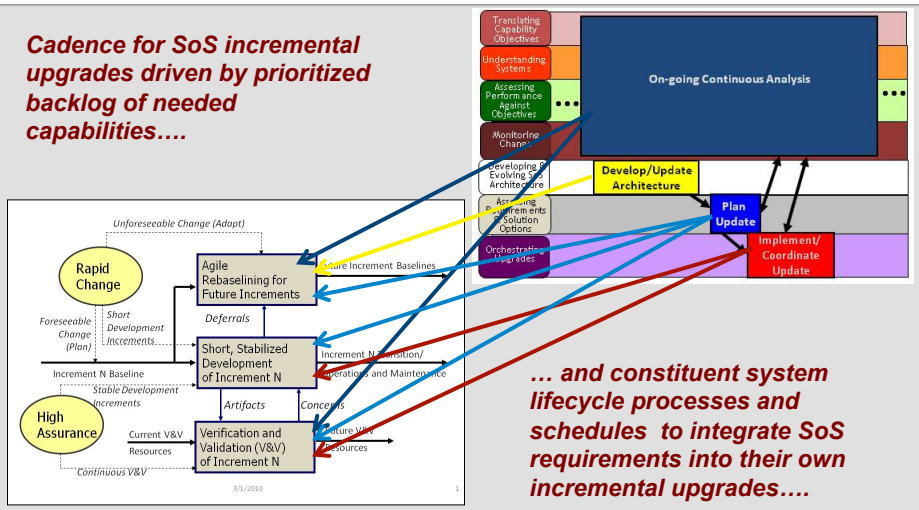
SoS Challenges

- Numerous independently evolving external systems or services outside span of control
- Complicated/complex acquisition, development and evolution environment
- Satisficing among multiple stakeholders
- Wide diversity of needed capabilities
- No one-size-fits-all solutions or processes
- Finding appropriate balance between
  - Cost
  - Schedule
  - Risk
  - Level of capability
  - Future adaptability/flexibility



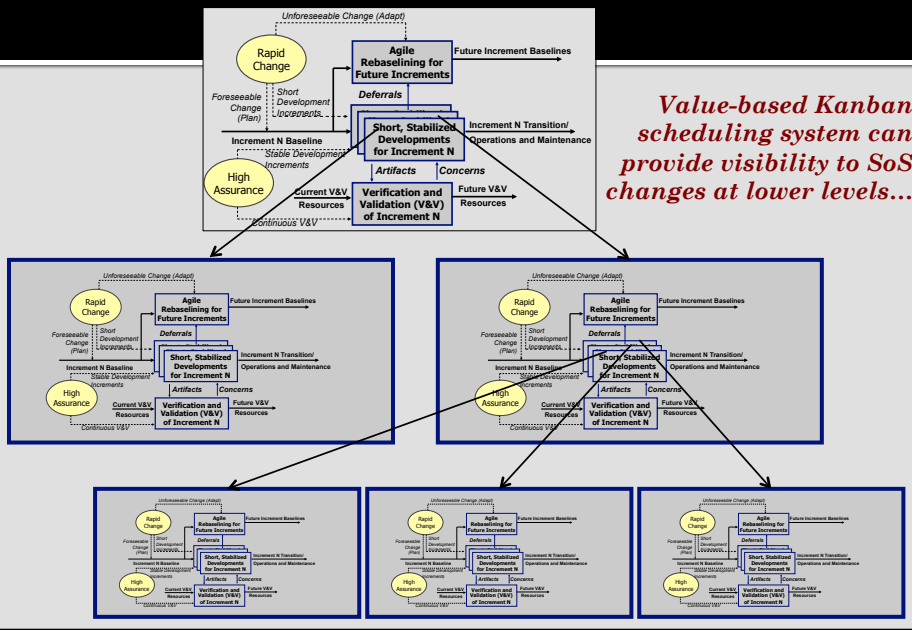
SoSE Process Strategies: Incremental Commitment Model for SoS

Cadence for SoS incremental upgrades driven by prioritized backlog of needed capabilities….



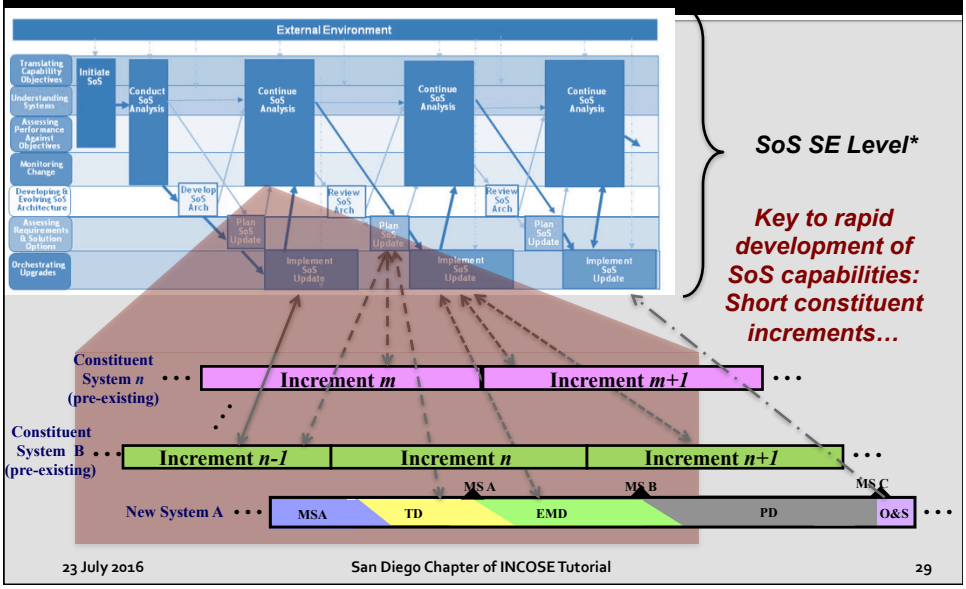
… and constituent system lifecycle processes and schedules to integrate SoS requirements into their own incremental upgrades….

Reality for Large/Complex Systems and SoS



Value-based Kanban scheduling system can provide visibility to SoS changes at lower levels…

SoSE Incremental Approach to Capability Synchronization



More on Feasibility Evidence for SoSE

- Evidence can include analysis/results of
  - Prototypes
    - Of networks, robots, algorithms, response times, COTS interoperability, etc.
    - To evaluate performance, scalability, accuracy, etc.
  - Exercises: for mission performance, interoperability, security
  - Models: for cost, schedule, performance, reliability; tradeoffs
  - Simulations: for mission scalability, performance, reliability
  - Analysis of infrastructure, data fusion, legacy compatibility
  - Previous experience
  - Combinations of the above
- Validated by independent experts
  - Realism of assumptions
  - Representativeness of scenarios
  - Thoroughness of analysis
  - Coverage of key off-nominal conditions

Sample Stage II Questions to Guide SoSE Activities

- What is the current status associated with capabilities/changes under development
  - Cost
  - Schedule
  - Quality assessments
  - Risks/risk mitigations
- For potential threats to success
  - Status of risk mitigations
  - Alternatives if constituent system is not successful with capability changes
- When and how to enable new capability(s)

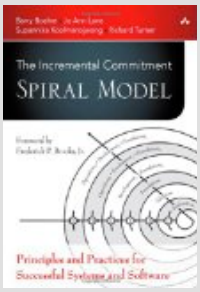
Much of Stage II technical work is done by constituent system developers using an appropriate ICSM common case for their system

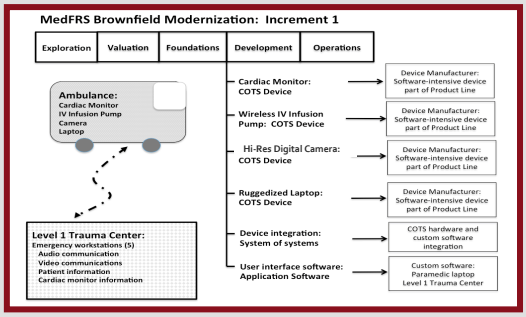
Pitfalls Common to SoSE

- Lack of attention to CS organizational and technical issues
- Understanding CS limitations (e.g., CS priorities vs. SoS priorities, interoperability, fragile systems that are difficult to change)
- Overly complex or complicated design
- Prototyping shortfalls
- Lack of attention to tech refresh coordination issues, especially those that may impact interoperability between systems
- Lack of planning for data/database conversions required for system upgrades
- Deployments using "all or nothing" approach vs, incremental rollout
- Inadequate attention to
  - How users are using constituent systems/SoS
  - User suggestions/complaints
  - Changing external systems and services that may impact operation
- Lack of attention to any required SoS level safety or security certifications
- Lack of integration and test planning/execution at the SoS level

More Available on ICSM for SoS

- Medical First Responder SoS case study
  - How the ICSM principles can be applied in the SoS case
  - Feasibility analysis summaries for each phase
  - Risk and risk mitigation strategies at each phase
- Guidance for incrementally adopting ICSM
- How ICSM fits with other standards and frameworks



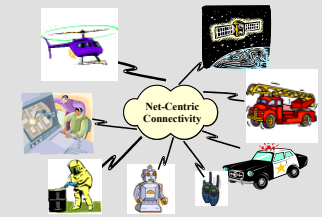


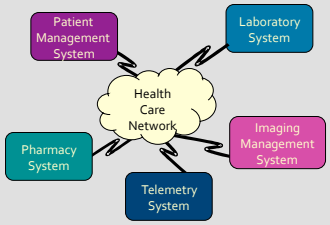
Overview

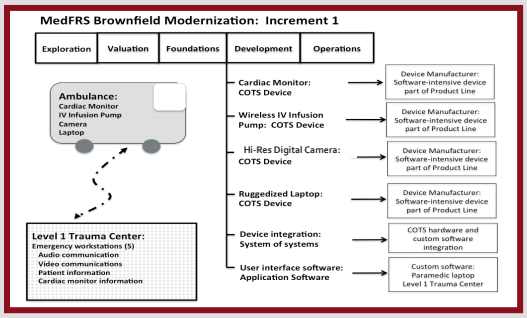
- SoS and SoSE basics
- Getting started with SoSE
- Incremental commitment and evolution focus
- ***SoS capability to requirements engineering***
- Keeping it lean

Time permitting:
- Case study analyses

Case Study Context







Capabilities to Requirements

Select desired capability(s)

↓

Identify resources and viable options

↓

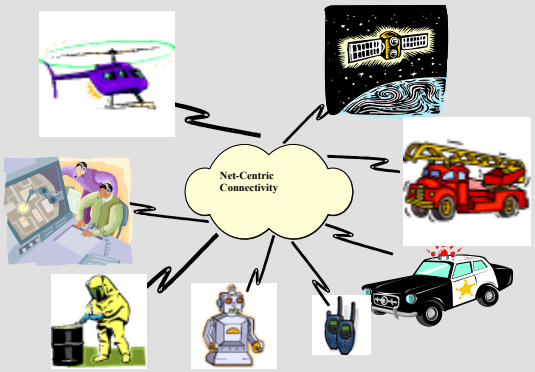
Assess options

↓

Select option

↓

Develop and allocate requirements to constituents



Illustrated using Regional Area Crisis Response SoS (RACRS)

Capability Engineering: Methods, Processes, & Tools

Identify resources:
SysML Objects

Determine options:
Responsibility/ dependability modeling

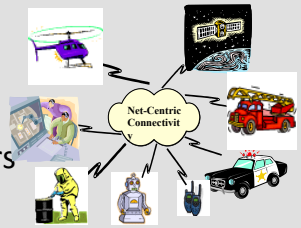
Candidate Feasibility Assessments for options:
- Net-centricity/ interoperability matrices
- Use cases/simulations to evaluate aspects of "how"
- Technical debt assessments for candidate constituents
- SCRAM assessments for candidate constituents
- Trades/simulations with respect to data fusion algorithms/formats
- Cost and schedule estimates

Select option

Develop and allocate requirements to constituents

RACRS Needs and Goals

- RACRS primary needs
  - Increase number of fire-fighting resources
  - Reduce official personnel effort to evacuate areas
  - Protect evacuated areas from looters
- Related goals
  - Minimize local government expense
  - Minimize risk to human life
  - Minimize workload on skilled personnel

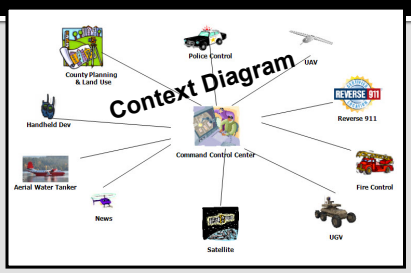


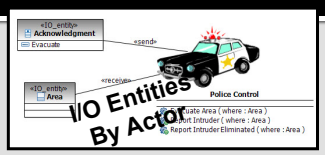
Potential RACRS Resources/ Systems Available to Support Need

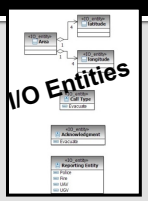
- Personnel
  - Local: fire fighters, police, and sheriff personnel
  - Volunteer civilians
  - Military personnel at local bases
  - National Guard personnel
  - Low-risk inmates incarcerated in local jails
  - TV/radio station announcers
- Local government systems
  - Fire systems
  - Sheriffs systems
    - 911 system
    - Command and Control Center
    - Cars
    - SWAT robots
  - Police systems
  - Paramedic systems
- Military systems
  - Unmanned aerial vehicles
  - Unmanned ground vehicles
- Other systems
  - Private airplanes that can drop fire retardant
  - Private helicopters that can drop water
  - Canadian aerial water tanker for major water drops (potential lease)
  - Satellite and local road camera images showing crisis areas and traffic status
  - Local news cast systems
  - Buses for evacuating people
  - Reverse-911 system (potential new local government system)
  - Homeowner alarm/security systems

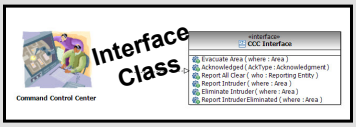
SysML Modeling for SoS:

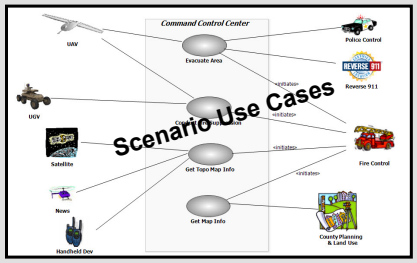
Regional Area Crisis Response SoS (RACRS)

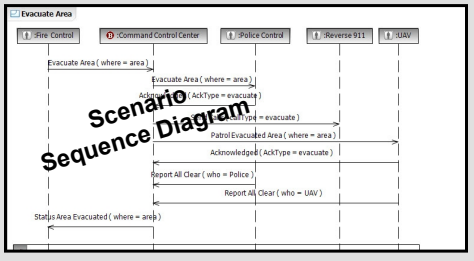












SysML Models that Support SoS Engineering Needs

- Object classes
  - Characterize each SoS CS and its capabilities
- Interface classes
  - Describe each CS interface
- Input/output entity classes
  - Express the associated data attributes of each data item transferred over that interface
- Use cases
  - Characterize both CS and SoS capabilities from the different user perspectives
- Sequence diagrams
  - Characterize and analyze the operational flow for an SoS capability
- Logical data models
  - Details about available information within each CS

RACRS Responsibility/Dependability Model

Responsibility	Resources							
	Fire trucks	Sheriff cars	Water tankers	UAV	Reverse 911 system	Ambulances	Buses	Manual
Fight fire	Local Regional Military	Local	Canadian company	Military				Local Regional Military Volunteer Low-risk inmates
Evacuate homes and businesses	Local Volunteer				Local CCC personnel			
Evacuate assisted living homes	Local	Local			Local CCC personnel		Local transit Charter	Assisted living staff Volunteers
Evacuate hospitals						Public Private		Hospital staff Volunteers
Prevent looters		Local		Military				

Agents

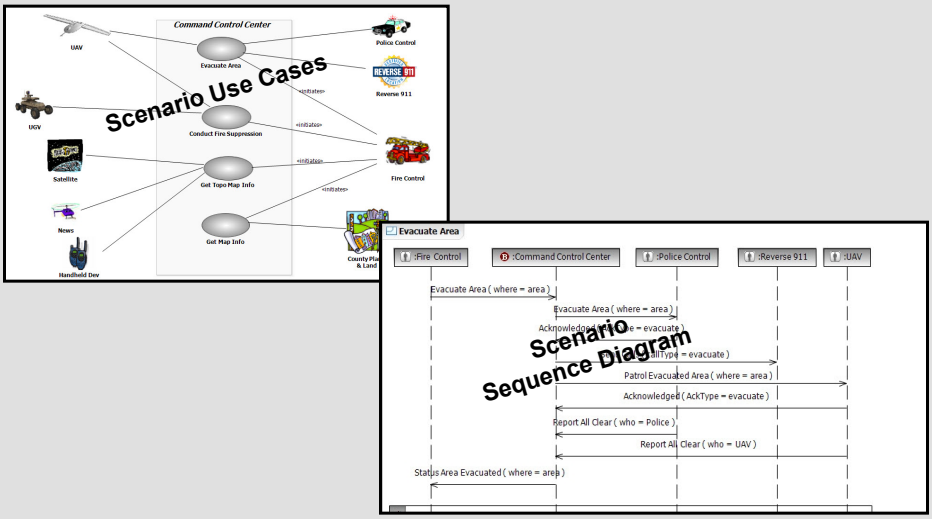
RACRS Responsibility/Dependability Model *(continued)*

Risk	Target	Hazard	Condition	Consequence	Severity
2	Canadian water tanker	*Late* or *never* due to other commitments	Reduced fire-fighting capability	More extensive fire damage, longer to put fires out	Medium to high, depending on other available resources
4	Reverse 911 system	*Insufficient*	Not all residents notified to evacuate	Residents at risk for being trapped/affected by crisis e.g., fire, hazardous material	Low to high, depending on type of crisis requiring evacuation
5	Low-risk inmates	*Various*—may be unskilled, may escape custody	Fire-fighting capability is less than that of experts	Additional resources required to train and monitor inmates	Low severity with respect to crisis, but medium severity with respect to costs related to training and monitoring

Net-centricity/Interoperability Matrices

Fire-fighting constituents	Local	Regional	Military	Canadian	Volunteer	Low-risk inmates
Local						
Regional	Functional					
Military	Isolated					
Canadian	Connected	Connected	Isolated			
Volunteer	Connected via handheld devices	Isolated	Isolated	Isolated		
Low-risk inmates	Connected via handheld devices	Isolated	Isolated	Isolated	Connected via handheld devices	

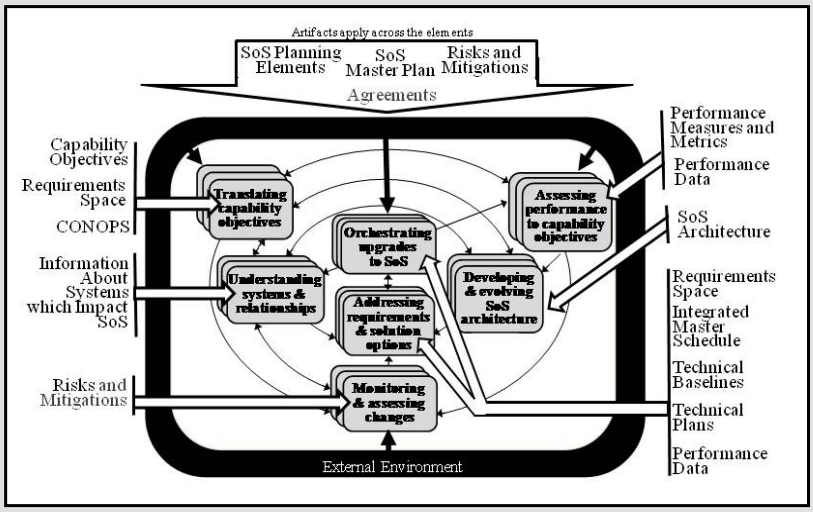
Assess Options Using SysML Modeling for SoS



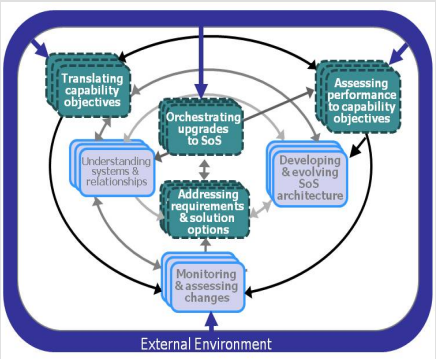
SoS Data Fusion Trade Analyses

- Trade considerations
  - Level of fusion
  - Functional aggregation
  - Producer-consumer reconciliation
  - Incongruent or inconsistent metadata management
  - Concept of operations with respect to the fusion(s)
  - Fusion lifecycle
  - Network topology
  - Information assurance
- Related to several previous analyses

SoSE Management/Engineering Artifacts

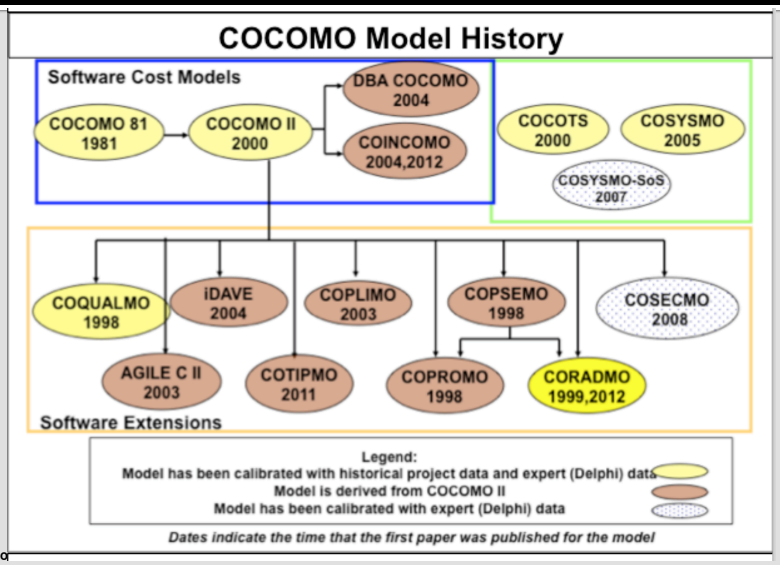


SoSE Test and Evaluation

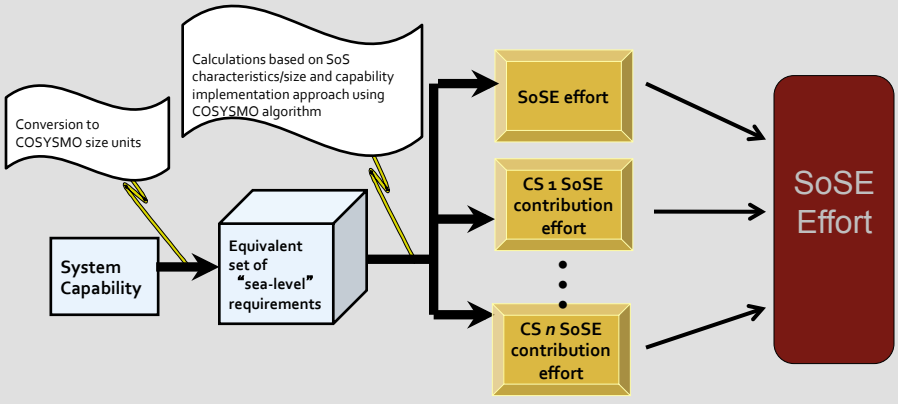


- Key strategies
  - SoSE team/framework responsible for SoS-level testing
  - Build on CS-level testing
  - Use of test ranges and operational testing
  - Risk and evidence-based approach
  - Focus on
    - SoS capabilities
    - Network
    - Interoperability
    - Emergent behaviors
  - Performance assessment over time
  - Feedback process for fielded SoS

COCOMO Family Models



SoSE 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….

Overview

- SoS and SoSE basics
- Getting started with SoSE
- Incremental commitment and evolution focus
- SoS capability to requirements engineering
- ***Keeping it lean***

Time permitting:
- Case study analyses

Initial Lean Indicators in SoSE

- Holistic view across SoS, focus on long term goals
- Guided by prioritized stakeholder capability needs
- Continuous learning organization
  - Monitoring and assessing changes
  - Assessing performance to capability objectives
- SoSE "battle rhythm"
  - Make decisions slowly by consensus
  - Consider all options
  - Continuous process flow through concurrent engineering
  - "Pull" CS engineering knowledge as needed
  - Implement rapidly
  - Level the workload—use systems with available "bandwidth"
- Minimize waste by
  - Avoiding duplication of engineering efforts at CS level
  - Eliminating none-valuing adding engineering activities
- Respect and challenge suppliers (CS engineering teams)

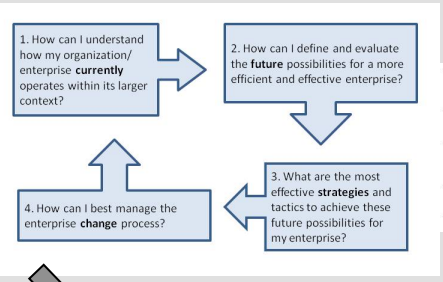
Objectives of Lean Enterprise Principles*

- Minimize waste
- Be responsive to change
- Right thing at the right place, at the right time, and in the right quantity
- Effective relationships (people and organizations) with the value stream
- Continuously improve processes and products
- Focus on quality from the beginning

* E. Murman, et al., Lean Enterprise Value: Insights from MIT's Lean Aerospace Initiative. New York , NY: Palgrave, 2002

SoSE Process Strategies: Incorporation of Lean Enterprise Principles

LAI Lean Enterprise 4 Grand Questions



Lean Enterprise 4 Grand Questions mapped to DoD SoSE case studies….

SoS ID	Lean Enterprise Grand Questions				Scope of Key Stakeholder Considerations	Number of Constituents
	Q1: Understand Current	*Q2: Future Possibilities*	*Q3: Strategies and Tactics for Future*	*Q4: Change Process*		
1	X	X	X	Monitor and attempt to guide, but do not manage	Across all DoD services/coalitions	16
2	X	X	X	Work in concert with constituent systems to manage upgrades	Across multiple DoD services and government agencies	9
3	X	X	X	Monitor and attempt to guide, but do not manage	Across all DoD services	10
4	X	X	Coordinated by large	Managed from above		Many – exact

Lean Enterprise 4 Grand Questions mapped to SoSE core elements….

SoSE Core Element	Lean Enterprise Grand Questions				*Stakeholder Considerations*	*Holistic Enterprise View*
	Q1: Understand Current	*Q2: Future Possibilities*	*Q3: Strategies and Tactics for Future*	*Q4: Change Process*		
Translating Capability Objectives		X			X	X
Understanding Systems and Relationships	X				X	X
Assessing Performance to Capability Objectives	X	X			X	X
Developing and Evolving an SoS Architecture	X	X	X			X
Monitoring and Assessing Changes	X			X	X	X
Addressing Requirements and Solution Options			X	X	X	X
Orchestrating Upgrades to SoS				X	X	X

Lean Principles in SE*

- Identify Value
- Map the Value Stream
- Create Flow
- Establish Pull
- Seek Perfection

* LEfSE: Oppenheim, B. W. (2011). Lean for systems engineering with lean enablers for systems engineering (Vol. 82). John Wiley & Sons.

55

Identify Value

- LEfSE:
  - Establish the Value of the end product or system to the customer
  - Frequently involve the customer
- SoSE:
  - New SoS capabilities requests mapped and translated to changes in constituent systems
  - SoS stakeholders priorities are integrated with single system changes and priorities

56

Map the Value Stream

LEfSE:

- Map the systems engineering and product development value streams and eliminate non-value added elements
- Plan for frontloading the program
- Plan to develop only what needs developing
- Plan to prevent potential conflicts with suppliers development
- Plan leading indicators and metrics to manage the program

SoSE:

- The SoSE team continuously monitors and adjusts plans and priorities
- No "front-loading" is required, it is a continuous process
- "nice to have" changes are tracked as a low priority and wait for opportunities to include them with high value changes
- Obsolete changes dropped
- Track the SoSE performance and how it meets capability objectives

57

Create Flow

LEfSE:

- Clarify, derive, and prioritize requirements early and often during execution
- Front load architectural design and implementation
- Systems engineers to accept responsibility for coordination of product development activities
- Use efficient and effective communication and coordination
- Promote smooth SE flow
- Make program progress visible to all
- Use lean tools

SoSE:

- SoS capabilities drive SoS evolution and its architecture, the SoS architecture is rarely developed from scratch
- SoSE team works to understand the current "as is" state of the architecture and then works to improve the existing architecture and associated performance
- Critical to success: collaboration with the constituent system engineers, understanding current development activities of constituent systems
- The SoSE team and system engineering teams of constituent systems may use Lean tools (e.g. multi-level Kanban boards).

58

Establish Pull

LEfSE:

- Pull tasks and outputs based on need, and reject others as waste

SoSE:

- The flow of SoSE work to constituent systems for development is typically based on a value-based priority scheme
- There are ways to use pull scheduling in work queues, but at the SoS level work often require specialized engineering skills, which limits options who can pull work from the queue.
- Pull scheduling can be used within the constituent systems.

59

Seek Perfection

LEfSE:

- Strive for excellence of SE processes
- Use lessons learned from past programs for future programs
- Develop perfect communication, coordination, and collaboration policy across people and processes
- For every program, use a chief engineer role to lead and integrate the program from start to finish
- Drive out waste through design standardization, process standardization, and skill-set standardization
- Promote all three complementary continuous improvement methods to draw best energy and creativity from all employees

SoSE:

- Most SoSE teams have limited resources and committed to optimization of their processes.
- Most SoSE teams have limited authority over the constituent systems; therefore, they have limited abilities to seek perfection in processes used by constituent systems.
- The SoSE team seeks perfection on the SoS level by evolving architecture, improving performance and collaboration with constituent systems.

60

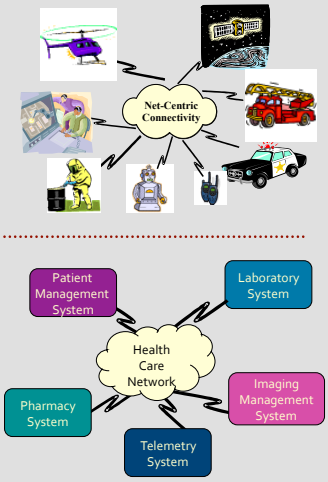
Overview

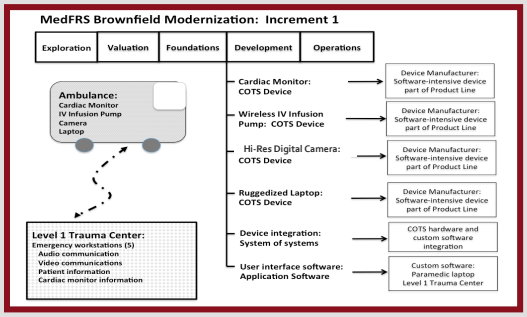
- SoS and SoSE basics
- Getting started with SoSE
- Incremental commitment and evolution focus
- SoS capability to requirements engineering
- Keeping it lean

Time permitting:

- ***Case study analyses***

Case Study Context





2003: Cedar Fire

Started: October 25 at 5:37 p.m.
Acreage Burned: 273,246
Damage:
- Structures destroyed:
  - 2,232 residences
  - 22 commercial properties
  - 566 outbuildings
  - 148 vehicles
  - 53 residences damaged
  - 10 outbuildings damaged
- Deaths: 14 civilians, 1 firefighter
- Injuries: 113 firefighters
- Firefighting Cost: $32 million

2007: Witch Creek Fire

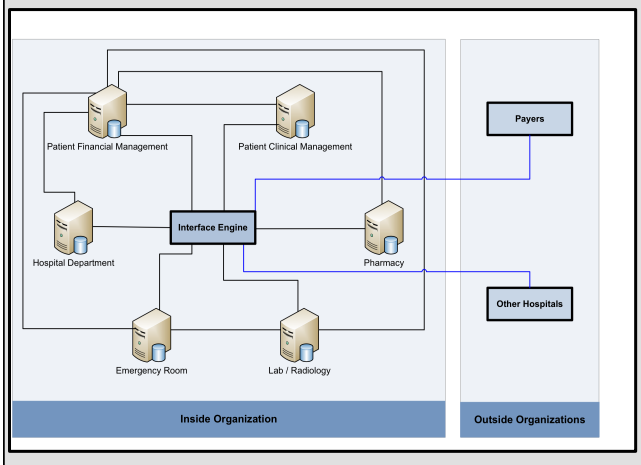
Started: October 21 at 11:00 a.m.
Acreage burned: 197,990
Damage:
- Structures destroyed:
  - 1040 homes
  - 414 outbuildings
  - 239 vehicles
  - 70 homes damaged
  - 25 outbuildings damaged
- Deaths: 2
- Injuries: 39 firefighters, 2 civilians
- Firefighting costs: $11.3 million

Four Years Later in San Diego

http://en.wikipedia.org/wiki/October_2007_California_wildfires
http://www.goldenstatephoto.com/weather/ca/fires/witchcreekfire

63

Interconnectivity and Interoperability: Challenges and Opportunities in a 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.

© USC CSSE

64

Discussion Questions: Background

Key desired features for Medical First Responder SoS

1. Upgrade and integrate constituent systems
2. Provide telemedicine capabilities to key trauma care facilities in the region
3. Connect to a variety of medical devices and telemetry systems using standard interfaces and providing a common user "look and feel" across the devices and systems, including consistent yet distinguishable alarms and messages
4. Incorporate an electronic health record compatible with regional health care facilities and trauma centers
5. Provide a high level of patient privacy and safety
6. Integrate information from patient "smart" stents that monitor blood chemistry and pressure into the telemedicine system once the stents have received US Food and Drug Administration approval
7. Provide a learning capability for future improved patient care.

Discussion Questions Assume Acknowledged SoS

- Where to start?
- What are key risks?
- What feasibility evidence should be pursued?
- What are the key engineering artifacts to be developed and evolved?
- How should increments be defined?
  - Which capabilities
  - Which components/systems
  - In what order
- How to test?
- How should increments be released/deployed?

Acknowledgements

- DoD Director, Defense Research and Engineering (DDR&E)
  - Stevens-USC Systems Engineering Research Center (SERC) support
  - SoS case study work that has provided considerable engineering insights into SoSE
- LAI for their research into lean enterprise concepts
- Dr. Ricardo Valerdi's COSYSMO cost model upon which the SoSE cost model is based

References

- A. Tregubov and J. Lane (2016); What Does it Mean to Be Lean in SoSE?, Proceedings of 26th Annual INCOSE Symposium, 18-21 July, Edinburgh, UK.
- A. Tregubov and J. Lane (2015); Simulation of Kanban-Based Scheduling for Systems of Systems: Initial Results, Proceedings of the Conference on Systems Engineering Research, 17-19 March, Stevens Institute of Technology, Hoboken, NJ.
- B. Boehm, J. Lane, S. Koolmanojwong, R. Turner (2014); The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software, Addison Wesley.
- D. Greenwood 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.
- Department of Defense (2008); Systems engineering guide for system of systems,
- G. Wang, G., R. Valerdi, A. Ankrum, C. Millar, and G. Roedler (2008); COSYSMO reuse extension, Proceedings of the 18th Annual International Symposium of INCOSE, The Netherlands.
- J. Dahmann and K. Baldwin (2008); Understanding the current state of US defense systems of systems and the implications for systems engineering. Proceedings of the IEEE Systems Conference, April 7-10, in Montreal, Canada.
- J. Dahmann, G. Rebovich, J. Lane, and R. Lowry (2010); System engineering artifacts for SoS. Proceedings of the IEEE Systems Conference, 5-8 April, in San Diego, CA.
- J. Dahmann, J. Lane, G. Rebovich, and R. Lowry (2010); System of systems test and evaluation challenges. Proceedings of the IEEE System of Systems Engineering Conference, 22-24 June, in Loughborough, UK.
- 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 *(continued)*

- J. Lane, (2014); Systems of Systems Capability to Requirements Engineering, Proceedings of the IEEE 9th Annual System of Systems Engineering Conference, Adelaide, Australia.
- J. Lane and R. Valerdi. 2010. How to Accelerate Understanding and Optimization of System of Systems Engineering through Lean Enterprise Principles, IEEE Systems Conference.
- J. Lane, A. Pitman, B. Clark, and A. Tuffley (2015); SoS Capability Schedule Prediction, Proceedings of the IEEE System of Systems Engineering Conference, 17-20 May, San Antonio, TX.
- J. Lane and R. Turner (2013); Improving Development Visibility and Flow in Large Operational Organizations, Proceedings of the Lean Enterprise Software and Systems Conference, December 1-4, Galway, Ireland.
- J. Lane, S. Koolmanojwong, and B. Boehm (2013); Affordable Systems: Balancing the Capability, Schedule, Flexibility, and Technical Debt Tradespace, Proceedings of the INCOSE Symposium, 24-27 June, Philadelphia, PA.
- J. Lane and T. Bohn (2010) Using SySML to evolve systems of systems. USC CSSE Technical Report USC-CSSE-2010-506.
- M. Maier (1998) Architecting principles for systems-of-systems. Systems Engineering 1, no. 4: 267-284.
- M. Solano (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.
- R. Valerdi (2005); Constructive systems engineering cost model. PhD. Dissertation, University of Southern California.
- R. Valerdi, R. and M. Wheaton (2005); ANSI/EIA 632 as a standardized WBS for COSYSMO, AIAA-2005-7373, Proceedings of the AIAA 5th Aviation, Technology, Integration, and Operations Conference, Arlington, Virginia.