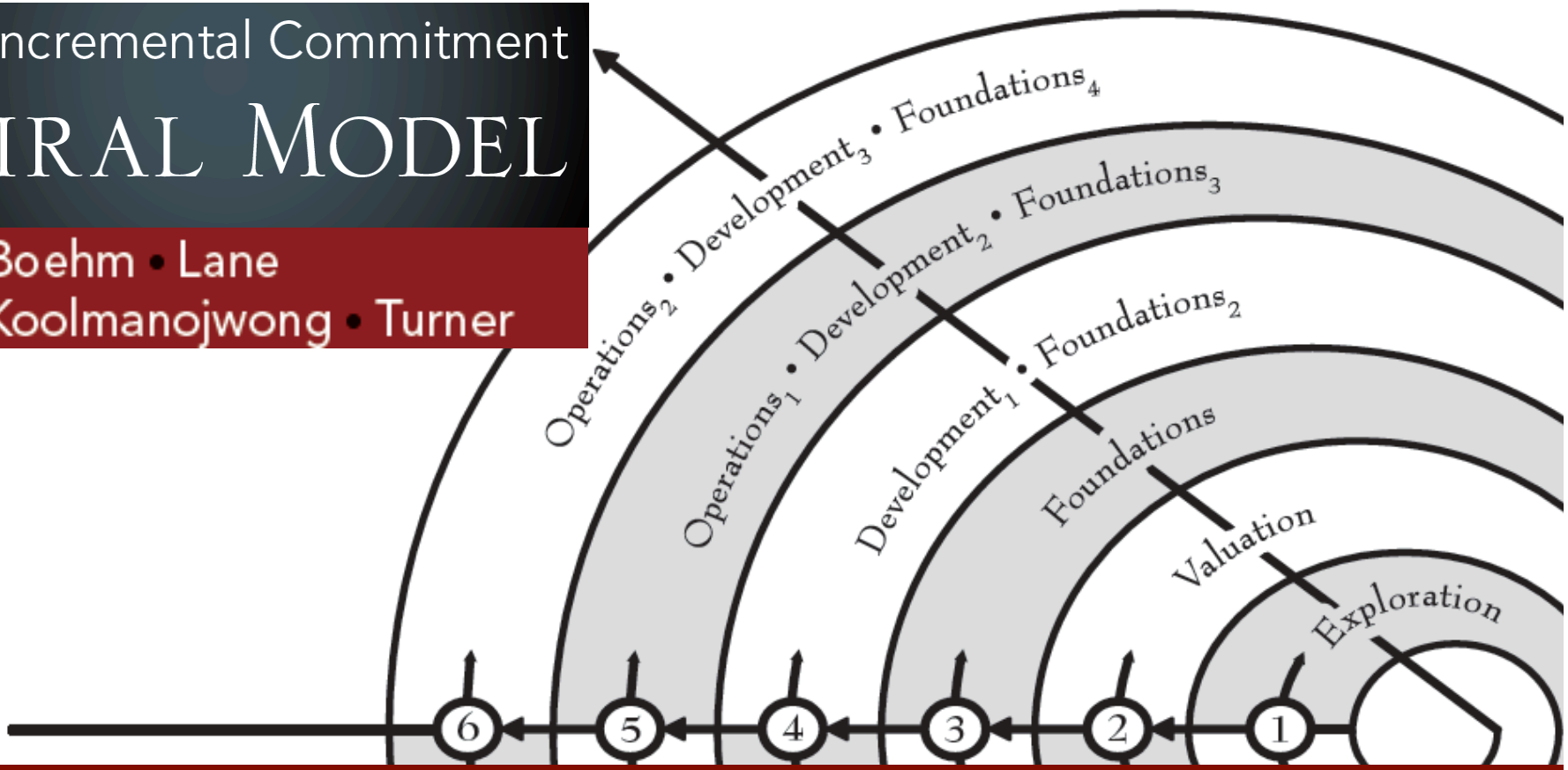
The Incremental Commitment

SPIRAL MODEL

Boehm • Lane
Koolmanojwong • Turner



The Incremental Commitment Spiral Model as Applied to SoS

Jo Ann Lane
(San Diego State University) jalane@mail.sdsu.edu
Richard Turner
(Stevens Institute) rturner@stevens.edu

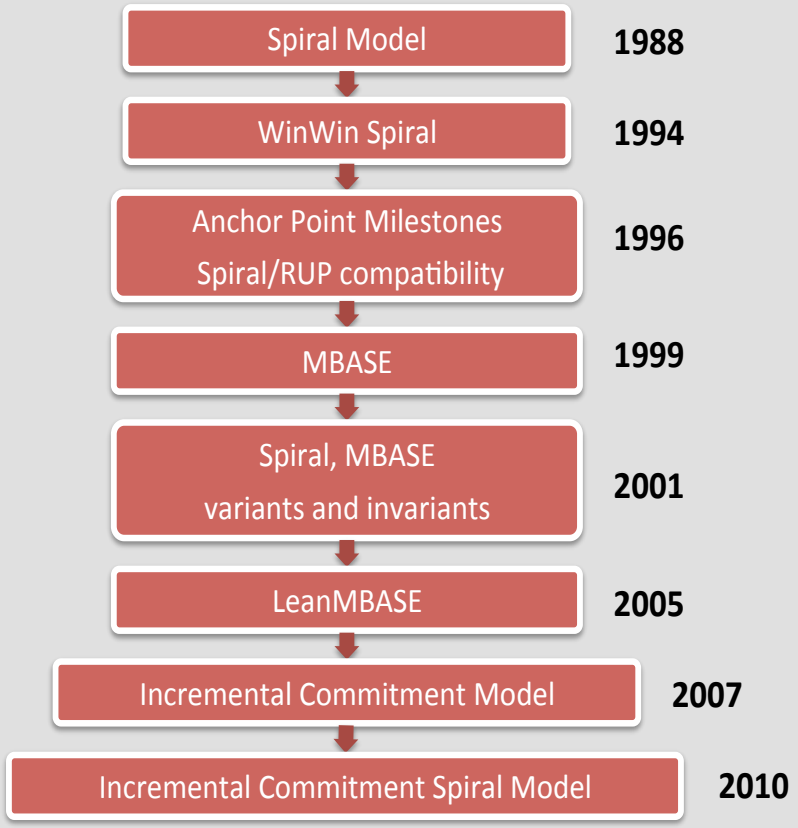
Presented to the SoSECIE – 8 September 2015

Agenda

- ICSM Fundamentals
  - Rationale and Legacy
  - ICSM Principles
  - ICSM General Framework and Views
- ICSM and Systems of Systems
  - ICSM for SoS Context
  - ICSM for SoSE
  - Sources for Additional Information and Related Research

ICSM Nature and Origins

- Integrates hardware, software, and human factors elements of systems life cycle
  - Concurrent exploration of needs and opportunities
  - Concurrent engineering of hardware, software, human aspects
  - Concurrency stabilized via anchor point milestones
- Responds to a variety of issues
  - Clarify "spiral development" usage
  - Provide framework for human-systems integration
- Builds on strengths of current process models, but not their weaknesses
- Facilitates transition from existing practices



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

What is Feasibility Evidence?

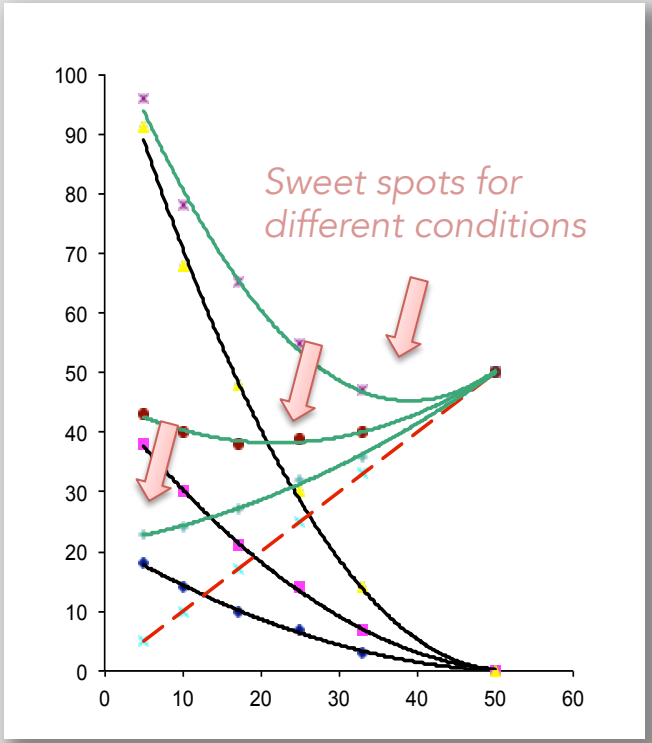
- Evidence provided by developer and validated by independent experts that:
- If the system is built to the specified architecture, it will
  - Satisfy the requirements: capability, interfaces, level of service, and evolution
  - Support the operational concept
  - Be buildable within the budgets and schedules in the plan
  - Generate a viable return on investment
  - Generate satisfactory outcomes for all success-critical stakeholders
- All major risks resolved or covered by risk management plans
- Serves as basis for stakeholder commitment to proceed
- Synchronizes and stabilizes concurrent activities

Can be used to strengthen current schedule- or event-based reviews

Meta-Principle (4+): Risk Balancing

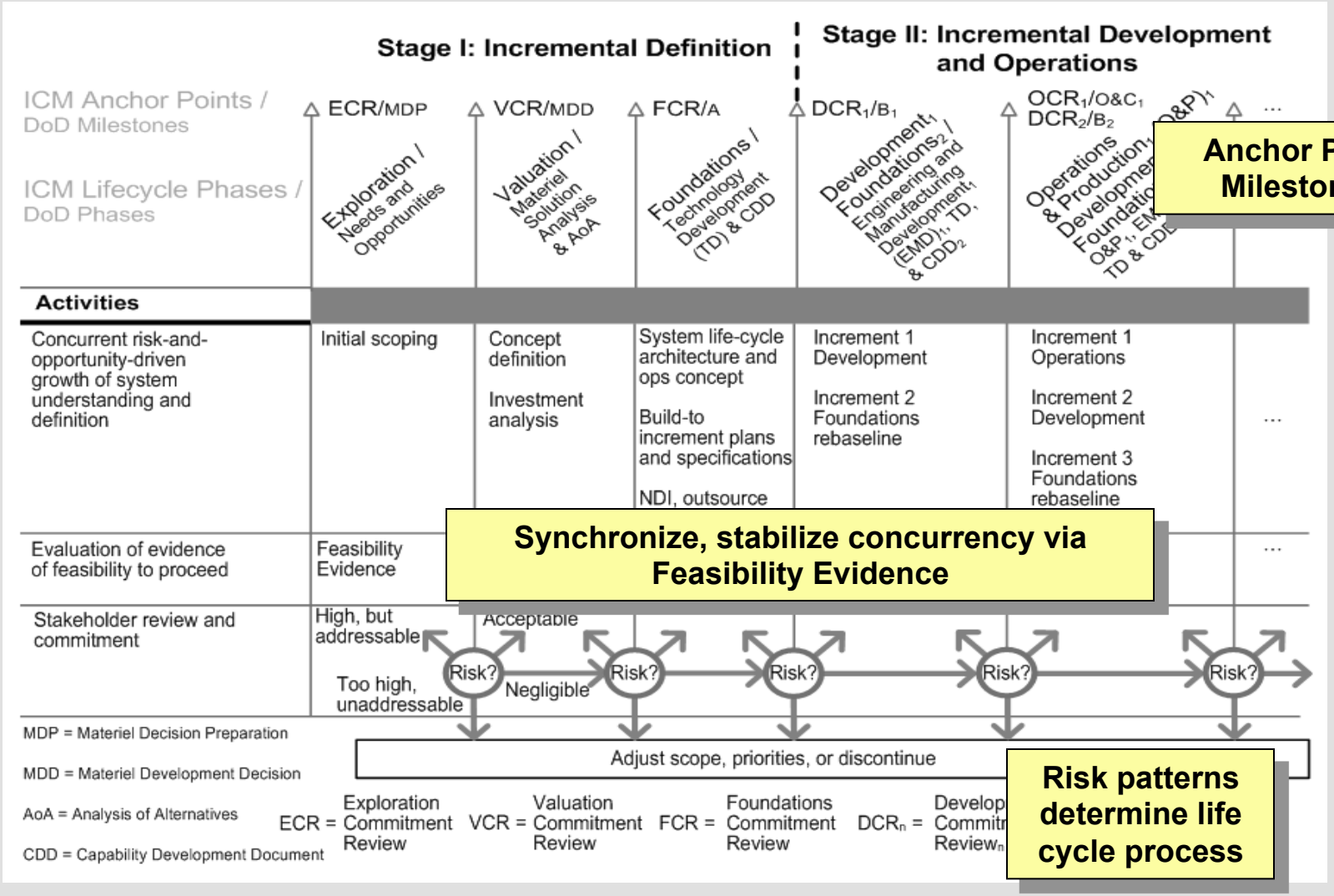
- Question: How much is enough?

• System scoping • Planning • Architecting • Prototyping • COTS evaluation • Requirements detail • Spare capacity • Fault tolerance • Safety • Security	• Environmental protection • Documenting • Configuration management • Quality assurance • Peer reviewing • Testing • Use of formal methods • Feasibility evidence



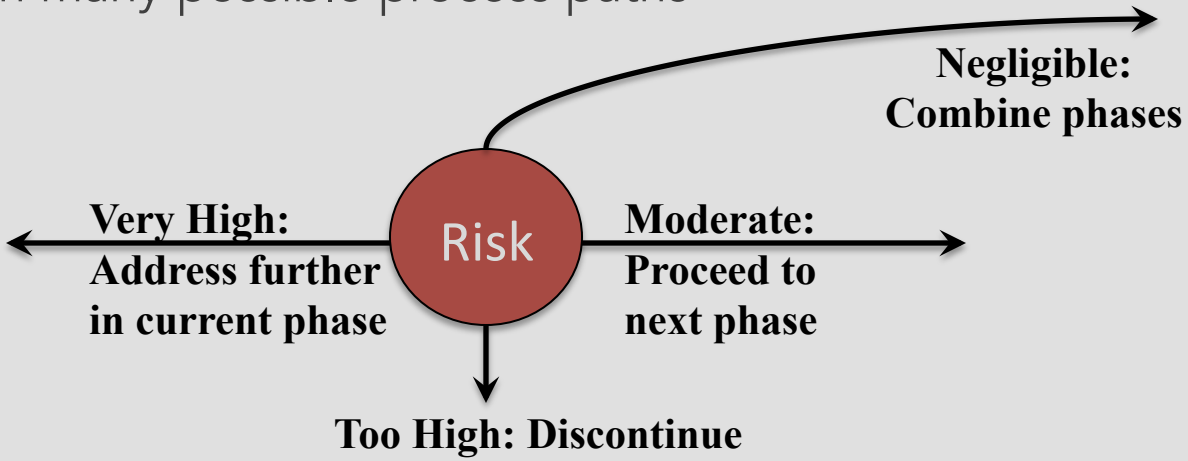
Answer: Balancing the risk of doing too little and the risk of doing too much will generally find a middle-course sweet spot that is about the best you can do.

The ICSM: Phased View



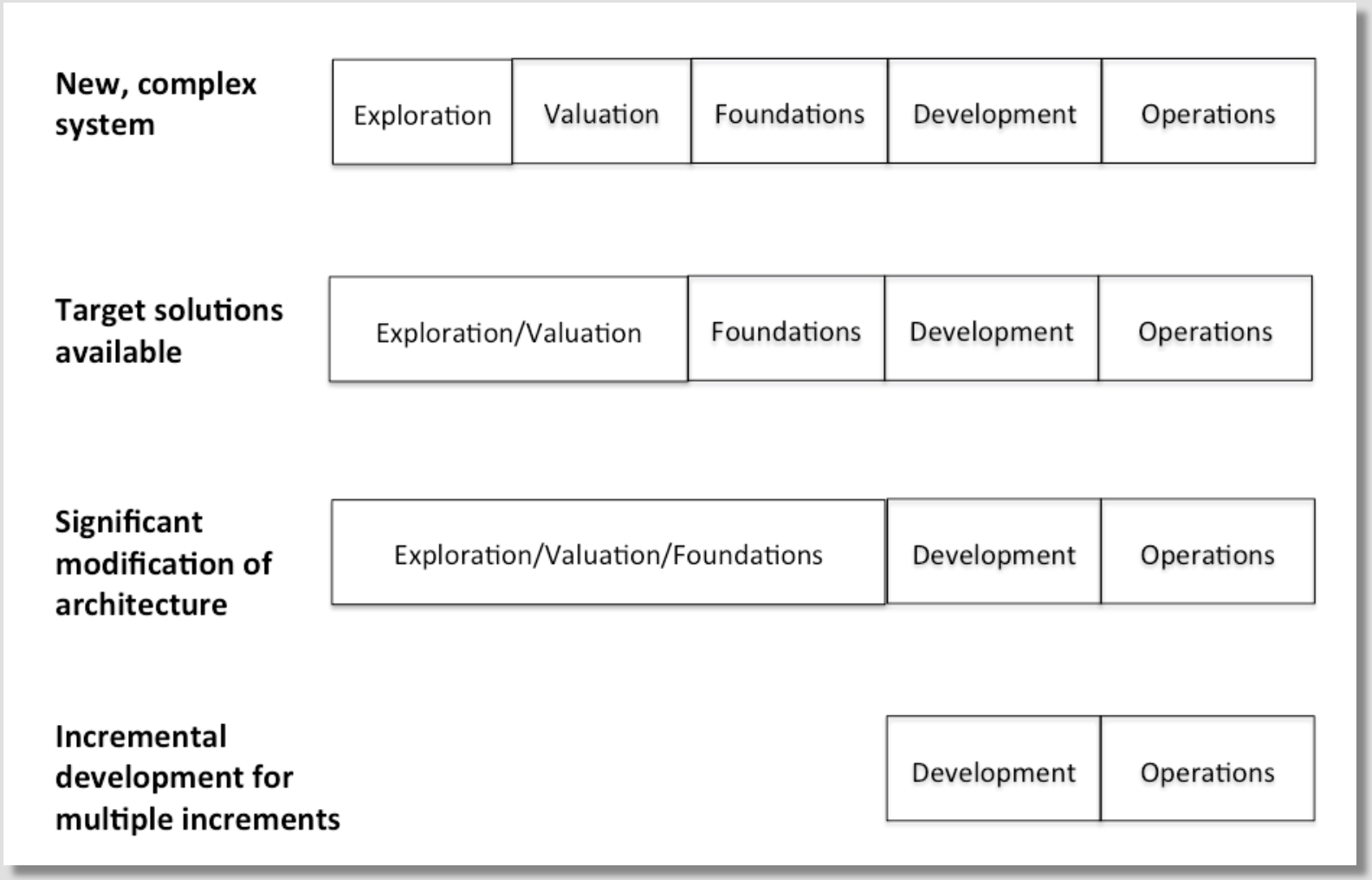
ICSM as Risk-Driven Process Generator

- ICSM has 5 decision anchors, each with 4 options
  - Risk-driven assessment on how to proceed
  - Some options involve go-backs
  - Results in many possible process paths



- Can use ICSM risk patterns to generate frequently-used processes
  - With confidence that they fit the situation
- Can generally determine this in the Valuation phase
  - Develop as proposed plan with risk-based evidence at FCR milestone
  - Adjustable in later phases

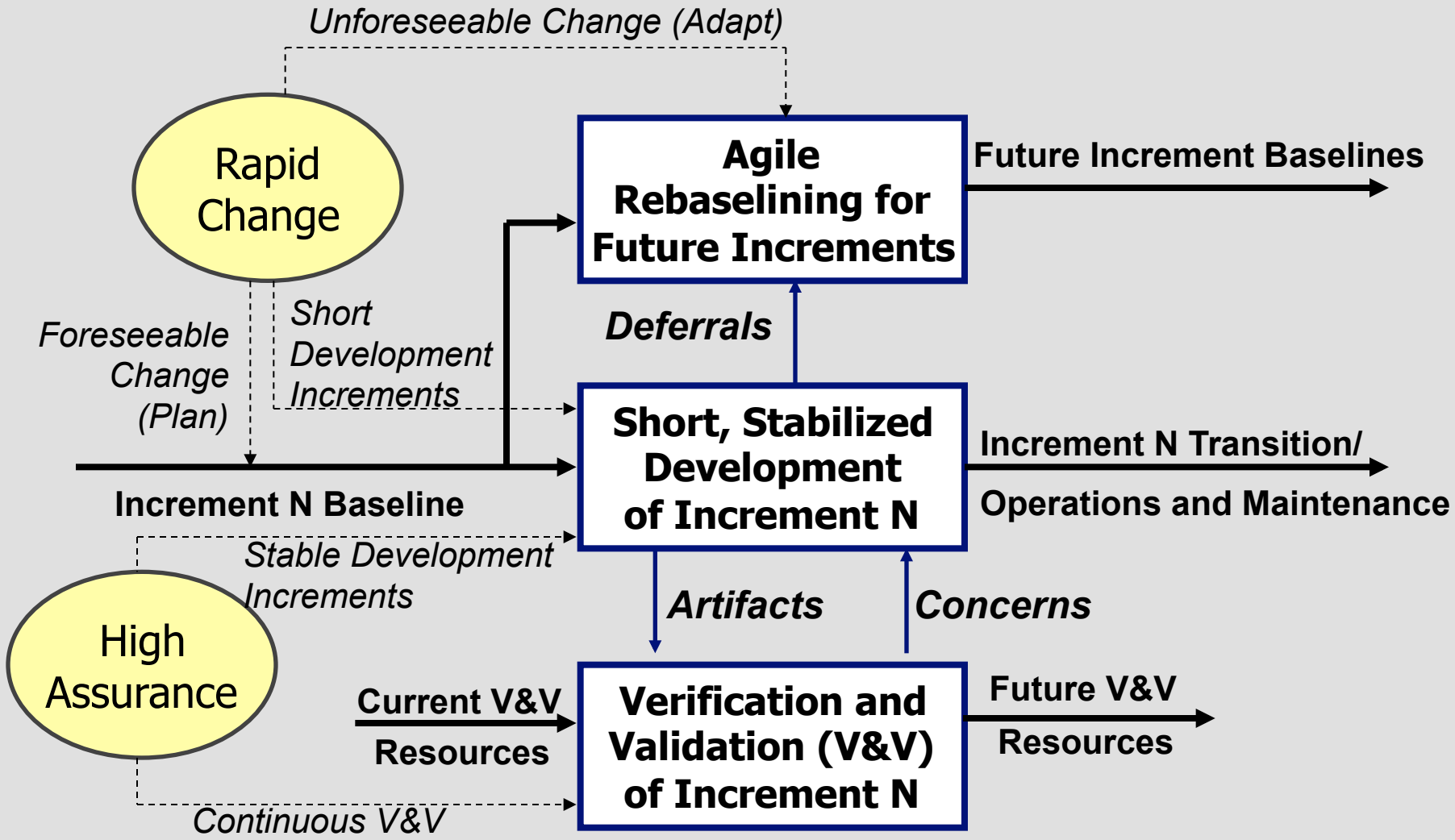
ICSM Patterns: How Phases Can Be Combined



Going slow, going fast: Phase combinations based on scope, risks, and maturity of solution space

ICSM: Increment View

Used for each incremental development of each system element or level of systems-of-interest

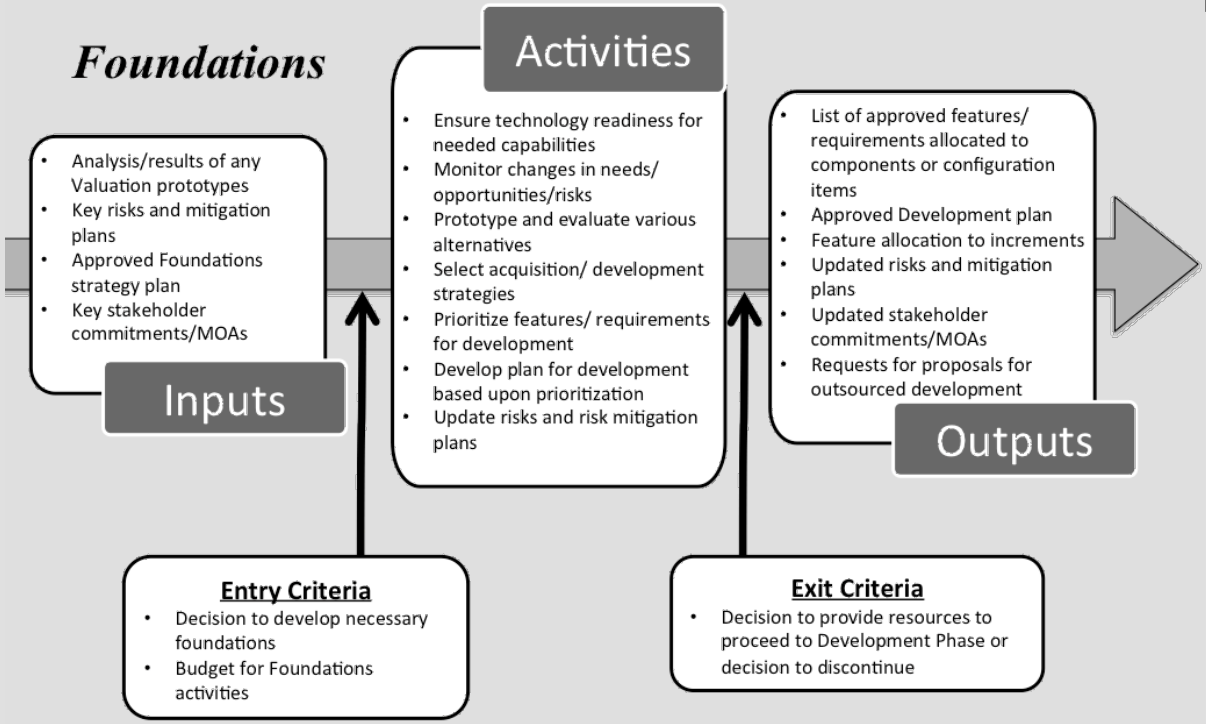


ICSM Common Cases

- Software application or system
- Software-intensive device
- Hardware platform
- Family of systems or product line
- **System of systems (SoS) or enterprise-wide system**
- Brownfield modernization

- Software strategies for software cases
  - Architected agile
  - Agile
  - Plan-driven
  - Formal methods
  - COTS/services

ICSM Guidance for Each Phase

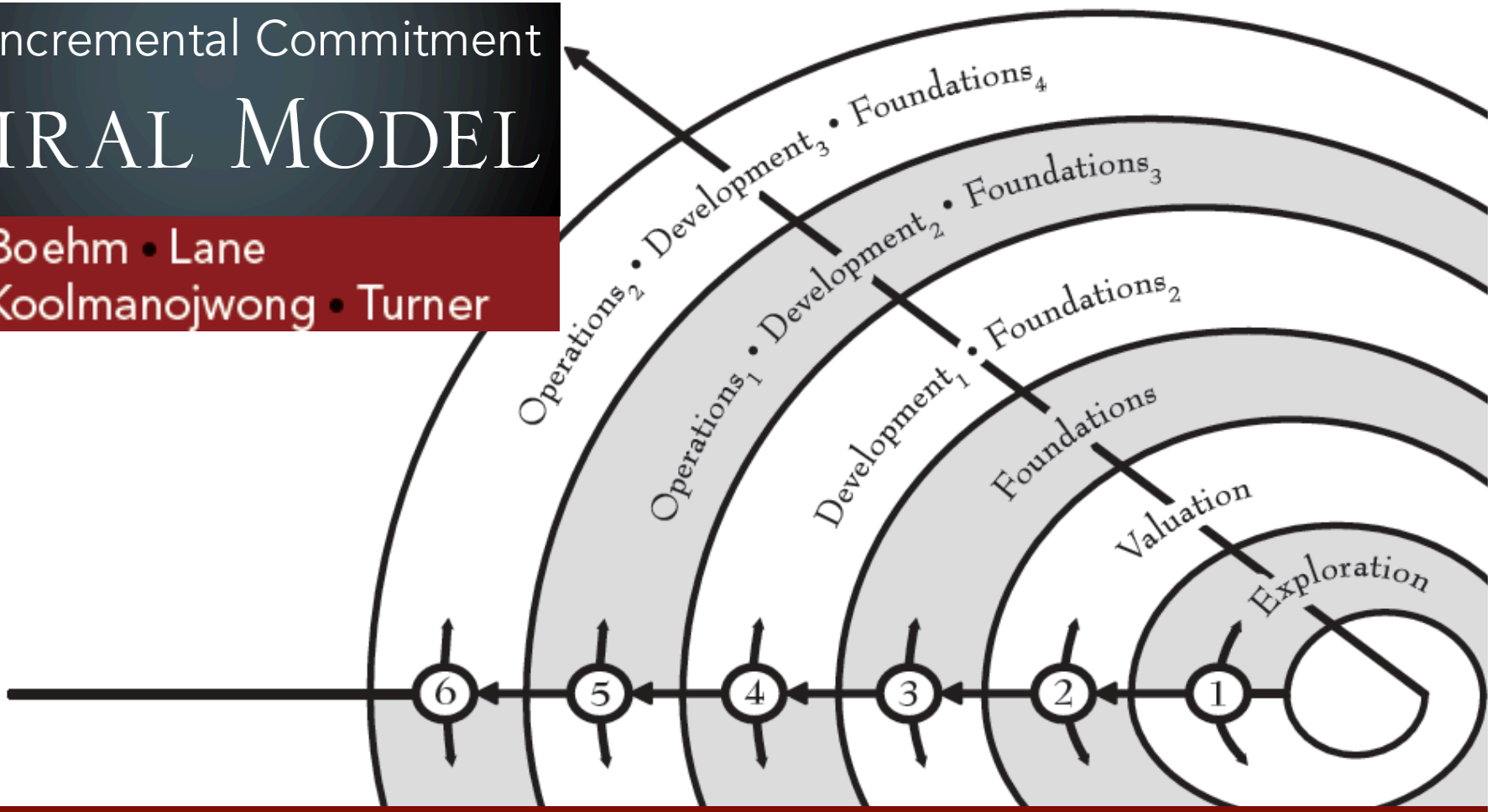


- Process diagrams plus:
  - Questions to guide phase activities
  - Potential pitfalls during phase
  - Likely major risks
  - How phase scales from small to large/ complex
  - Role of ICSM principles in phase

The Incremental Commitment

Spiral Model

Boehm • Lane
Koolmanojwong • Turner

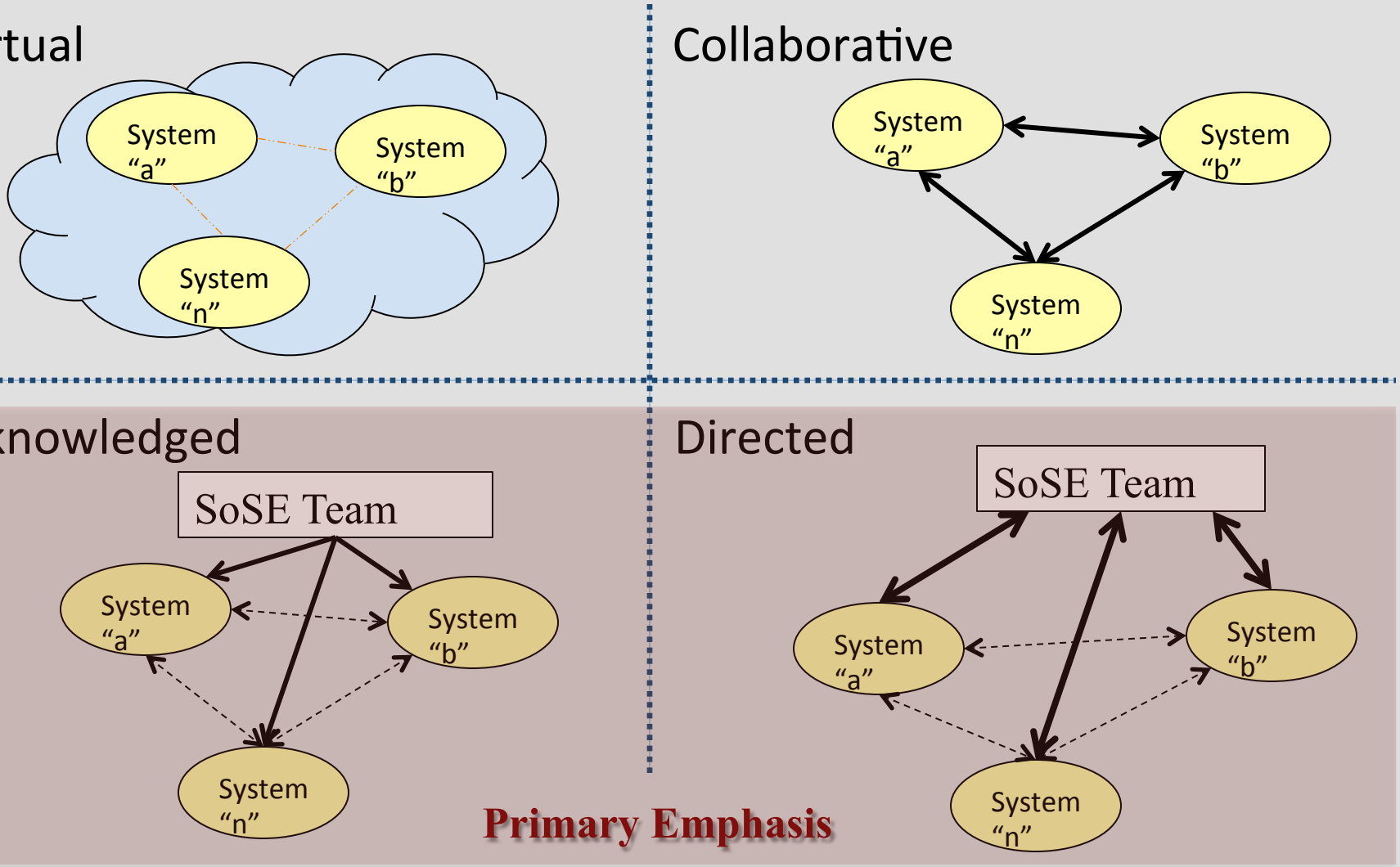


ICSM and Systems of Systems

ICSM Challenge: Multi-owner, multi-mission systems of systems (SoS)

- 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 of
  - Cost
  - Schedule
  - Risk
  - Level of capability
  - Future adaptability/flexibility

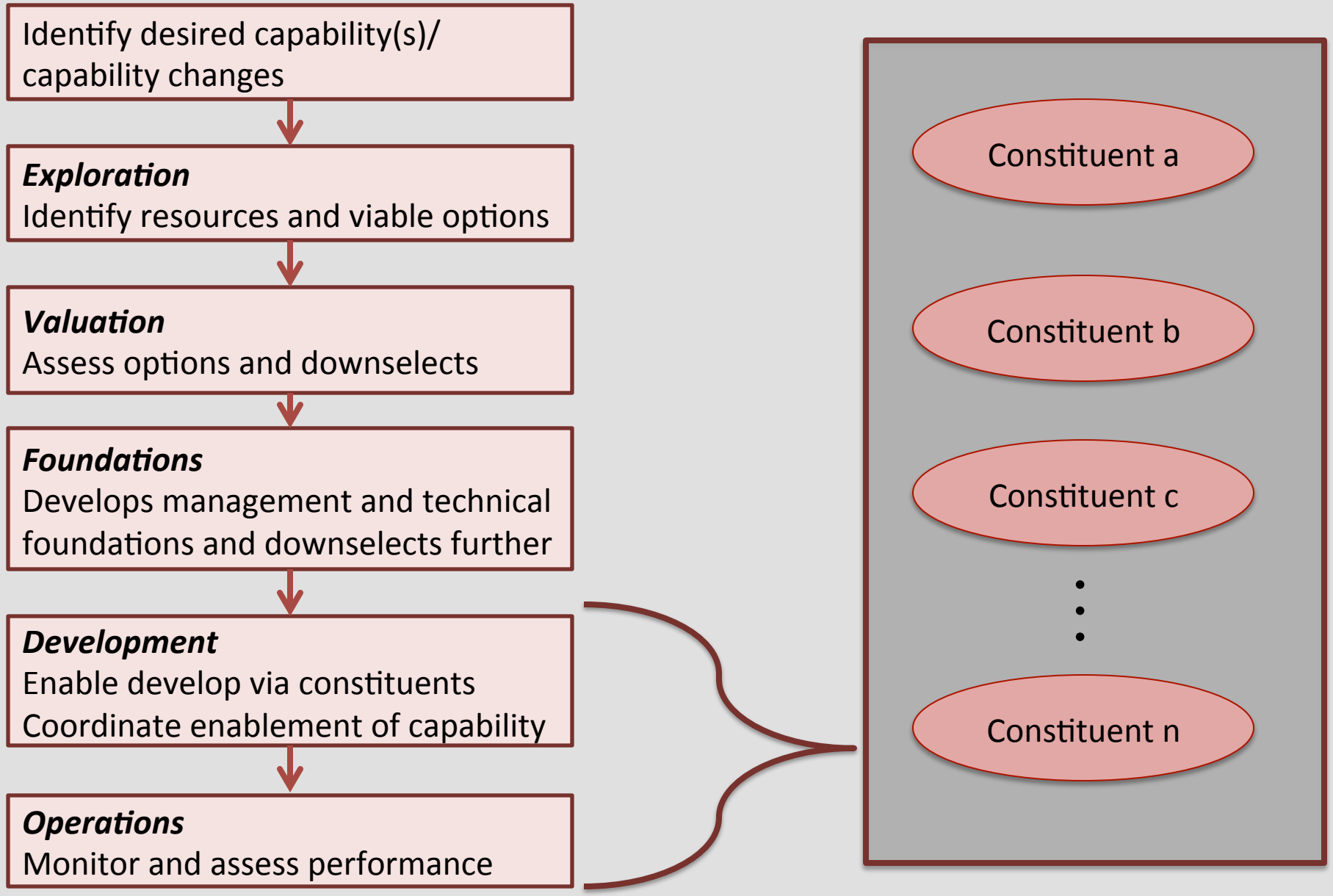
Types of SoS: Organizational Structures



ICSM Guidance for SoSE

- Questions to guide SoSE activities
- Potential pitfalls to avoid
- Major risks to watch for/mitigate
- Focus of principles for SoSE
- Examples of SoS capability feasibility evidence
- Key research contributing to ICSM for SoSE guidance:
  - Capability to Requirements Engineering (IEEE SoSE Conference 2014)
  - Schedule Compliance Risk Assessment Methodology (SCRAM) for SoS (IEEE SoSE Conference 2015)
  - Technical debt (journal paper submitted for publication)
  - Value-based scheduling for SoS (CSER 2015)

ICSM Phases for SoS Common Case



Sample Stage I Questions to Guide SoSE Activities

- What is the current state of the SoS
- What changes/new capabilities are desired
  - Who wants the new capability and why
  - Who are the key proponents and antagonists
  - How strong is the mission requirement/priority
- What are the value-based priorities associated with desired changes/new capabilities
- What are the options associated with each desired change/ new capability
  - Nontechnical options (e.g. operational changes)
  - Changes to existing constituent systems
  - Technical maturity, regulatory, legal, political, cultural issues associated with option
  - "New" system(s)
    - Interface to other existing systems or SoS
    - Commercial Off-the-Shelf (COTS) components
    - Develop new
- What is the expected "probability of success" for each option
- What is the expected value vs. cost for each option

Capability Engineering: Methods, Processes, & Tools

Identify Technical Resources
SysML Objects

Determine Organizational Factors
Responsibility/ dependability modeling

Example Feasibility Assessment Activities
- 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

Anchor Point Commitment Review
to select option

Develop and allocate requirements to constituents

Note: The level of rigor used is always risk-driven

More on Feasibility Evidence for SoSE

- Evidence can include results of
  - Prototypes
    - E.g. networks, robots, algorithms, response times, COTS interoperability
    - 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 and constituent systems
  - Realism of assumptions
  - Representativeness of scenarios
  - Thoroughness of analysis
  - Coverage of key off-nominal conditions

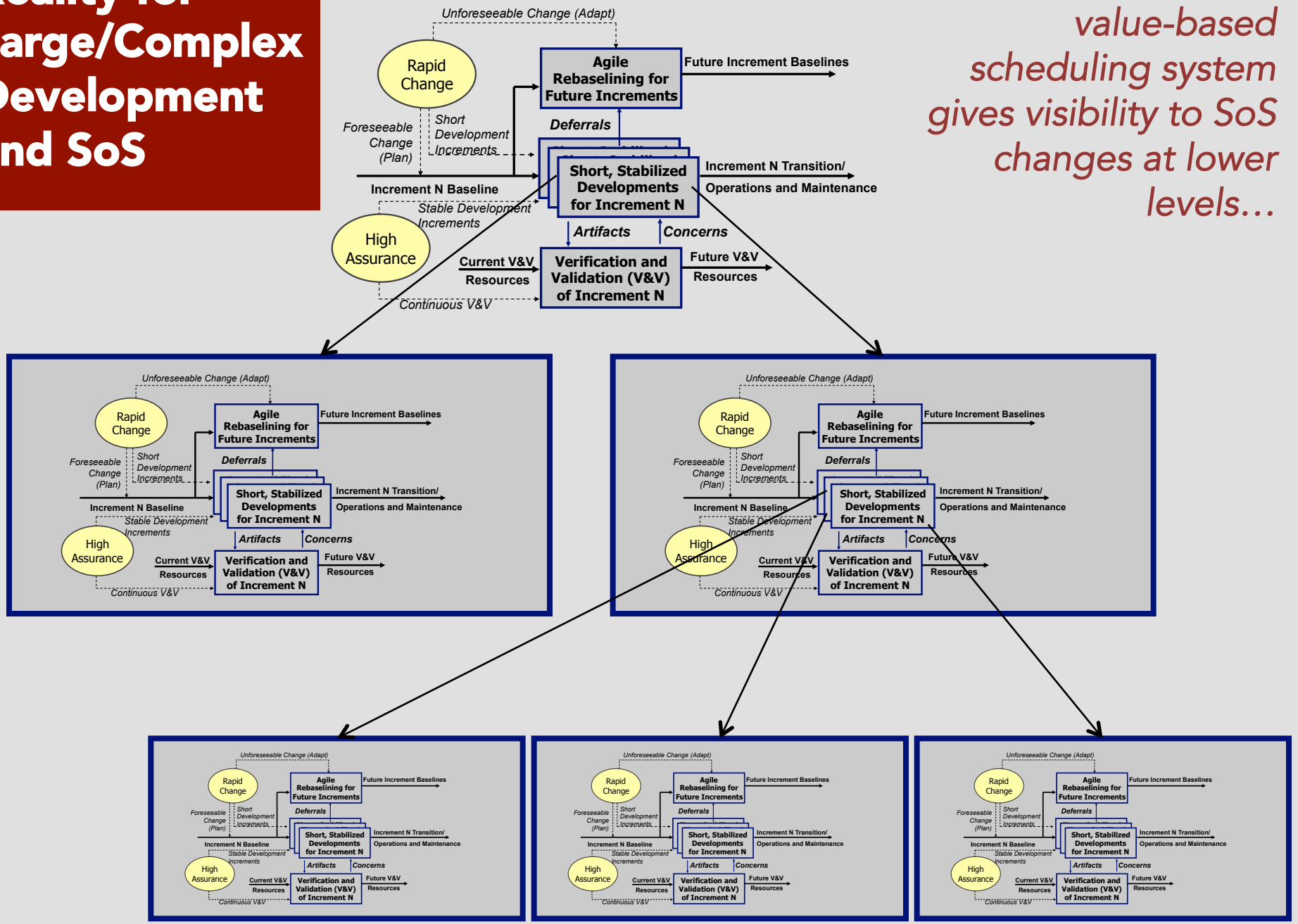
Sample Stage II Questions to Guide SoSE Activities

- 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 work is done by constituent system developers using an appropriate ICSM common case for their system

Reality for Large/Complex Development and SoS

Cross-constituent, value-based scheduling system gives visibility to SoS changes at lower levels…



Common Pitfalls for 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
- No attention to tech refresh coordination issues, especially those that may impact interoperability between systems
- Not 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
- No attention to required SoS level safety or security certifications
- Poor integration and test planning/execution at the SoS level

Capability-Related Risks for SoSE

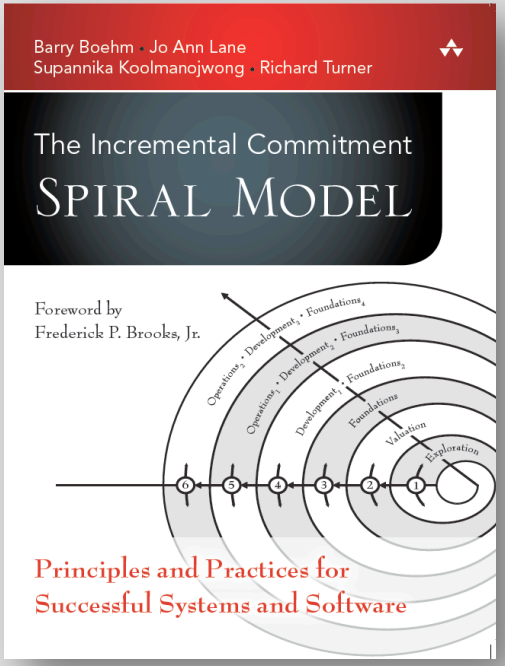
- Changing commitments of stakeholders/proponents/ constituents
- Key technologies that are not yet mature with respect to intended use
- Significant technical debt in constituent system(s) leading to schedule slips or capability gaps
- Reliance on older legacy systems that are close to end of life
- Critical engineering staff shortfalls
  - SoS-level
  - Constituent system level
- Lack of vendor support/weak critical links in candidate supply chains
- Overly optimistic plans, schedules, and estimates for next phase commitment
- Constituent systems do not understand the value of changes associated with SoS capabilities

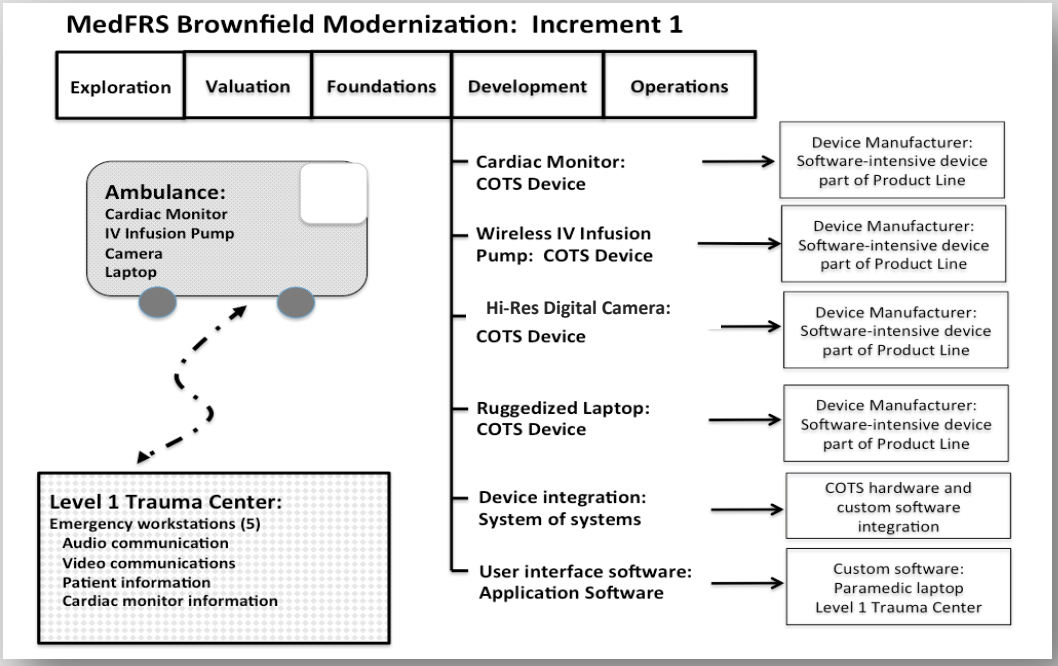
ICSM Principles Apply to SoSE in Spades!

- Stakeholder value-based guidance
  - Need balance between SoS and constituent system success-critical stakeholders
- Incremental commitment and accountability
  - Multi-way commitments and accountability between SoS stakeholders, constituent system stakeholders, and development organizations
- Concurrent system engineering
  - SoSE adds another level of concurrent engineering
  - Successful SoSE continually monitors for opportunities to expand and improve SoS capabilities
- Evidence and risk-driven decisions
  - SoSE level
  - Constituent system level
  - Needs to be compatible

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





On-going or Future SERC Work Related to ICSM for SoS

- Integration of SysML models with cost estimations models
- Agile/Lean SE in SoS environments (DATASEM)
- Assessing and quantifying technical debt to support SoS capability trades
- SERC toolbox for SoSE tools
- SoSE Experiences for the SE Experience Accelerator

Questions and Discussion?

References for Further Information

- B. Boehm, J. Lane, S. Koolmanojwong, and R. Turner (2014); *The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software*, Addison-Wesley, ISBN-13: 978-0-80822-6.
- 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.
- 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.
- Q. Zhanga, L. Huang, N. Jan, J. Lane, and H. Zhang (2015); Detecting and Evaluating Technical Debt in Software Systems: A Systematic Literature Review, submitted to the *Journal of Systems and Software*, June.
- R. Turner; L. Yilmaz; J. Smith; Donghuang Li; S. Chada; A. Smith.; A. Tregubov (2015); "Modeling an organizational view of the SoS towards managing its evolution," System of Systems Engineering Conference (SoSE), 2015 10th , vol., no., pp.480,485, 17-20 May.
- J. Lane (2014); Systems of Systems Capability to Requirements Engineering, Proceedings of the IEEE 9th Annual System of Systems Engineering Conference, Adelaide, Australia.
- R. Turner (2014); "Rediscovering Systems Engineering," INCOSE Insight, Vol.17, No. 2, July.
- B. Boehm; R. Turner; J. Lane; S. Koolmanojwong (2014); "High Maturity Is Not A Procrustean Bed," Crosstalk, Jul/Aug.
- R. Turner (2014); "Value-based Scheduling in System of Systems Evolution," Proceedings of the IEEE 9th International Conference on System of Systems Engineering (SoSE 2014).
- J. Lane and R. Turner (2013) "Improving Development Visibility and Flow in Large Operational Organizations," 4th International Conference on Lean Enterprise Software and Systems, Galway, Ireland, December 1-4, 2013, Proceedings, Lecture Notes in Business Information Processing, Vol. 167, pp 65-80, Springer-Verlag, Heidelberg.
- R. Turner (2013); "A Lean Approach to Scheduling Systems Engineering Resources," CrossTalk, May/June..
- J. Lane (2009); Cost Model Extensions to Support Systems Engineering Cost Estimation for Complex Systems and Systems of Systems, Proceedings of the Seventh Conference on Systems Engineering Research.