Future Challenges for Software Data Collection and Analysis

Barry Boehm, USC-CSSE

PROMISE 2009 Keynote

May 18, 2009

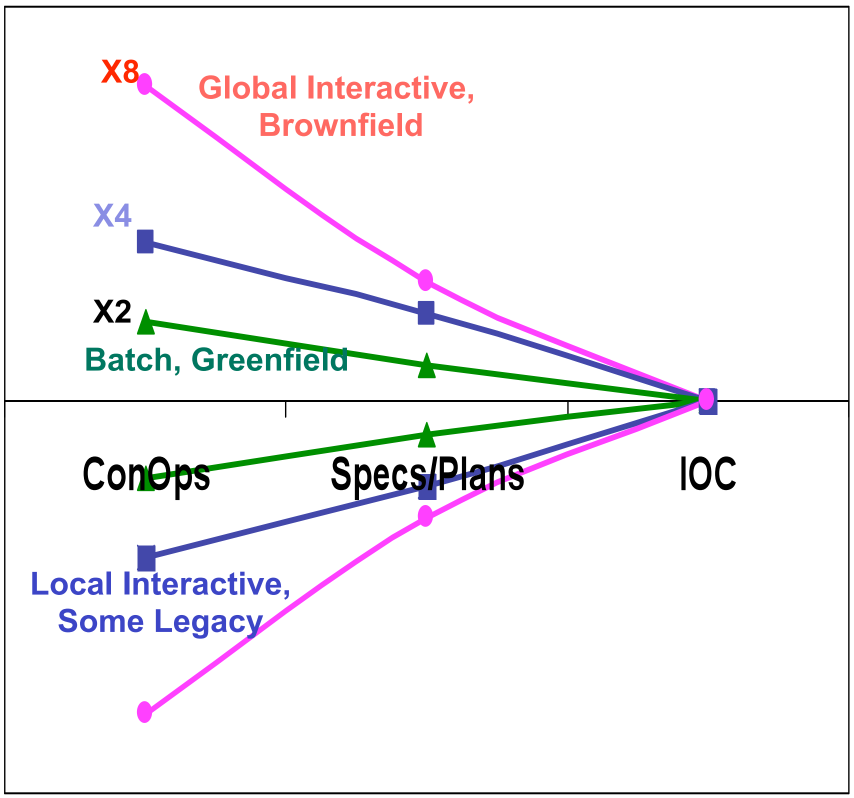
Future Software Measurement Challenges

- Emergent requirements
  - Example: Virtual global collaboration support systems
  - Need to manage early concurrent engineering
- Rapid change
  - In competitive threats, technology, organizations, environment
- Net-centric systems of systems
  - Incomplete visibility and control of elements
- Model-driven, service-oriented, Brownfield systems
  - New phenomenology, counting rules
- Always-on, never-fail systems
  - Need to balance agility and discipline

Emergent Requirements
– Example: Virtual global collaboration support systems

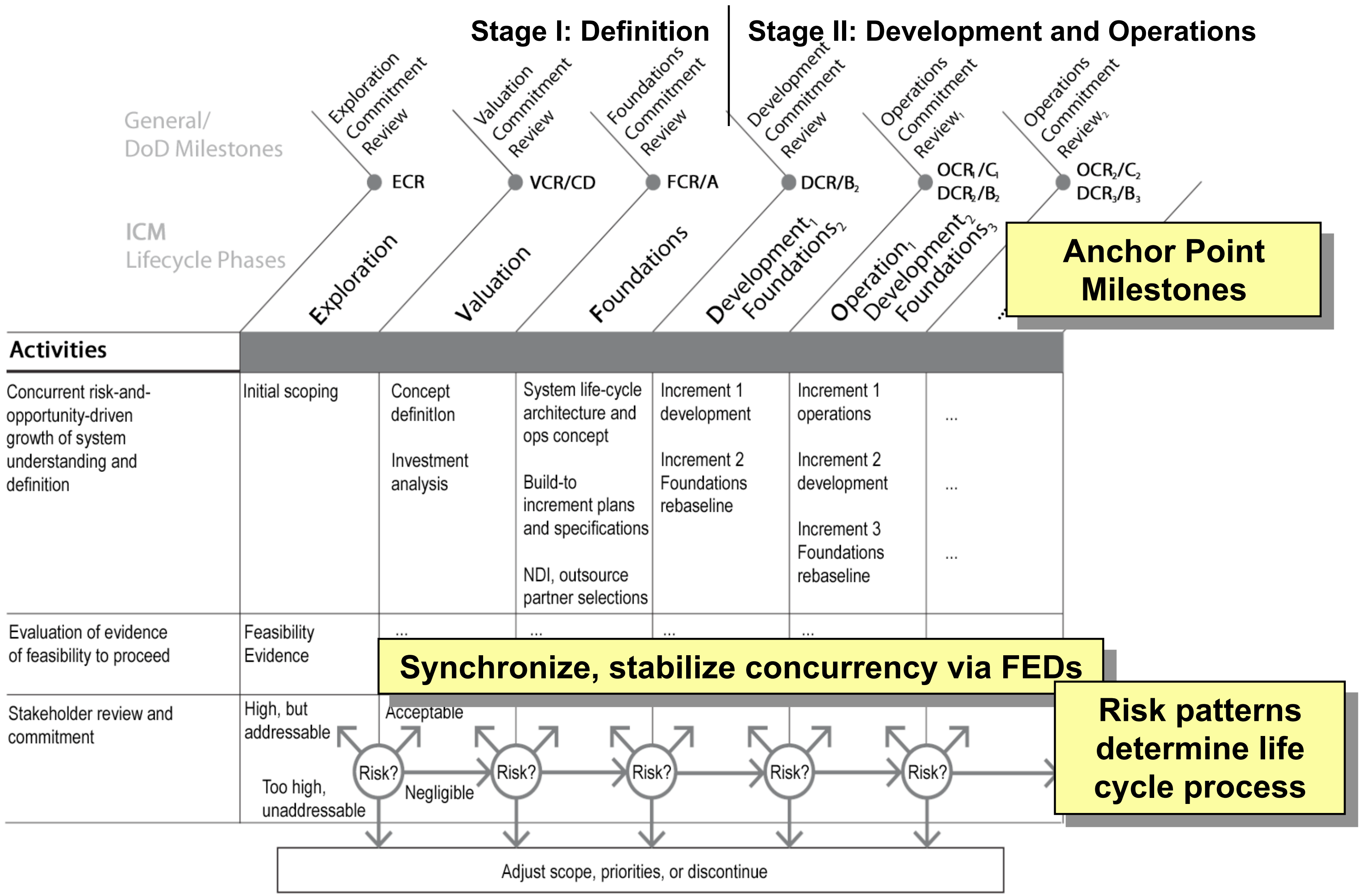
- View sharing, navigation, modification;agenda control; access control
- Mix of synchronous and asynchronous participation
- No way to specify collaboration support requirements in advance
- Need greater investments in concurrent engineering
  - of needs, opportunities, requirements, solutions, plans, resources

The Broadening Early Cone of Uncertainty (CU)

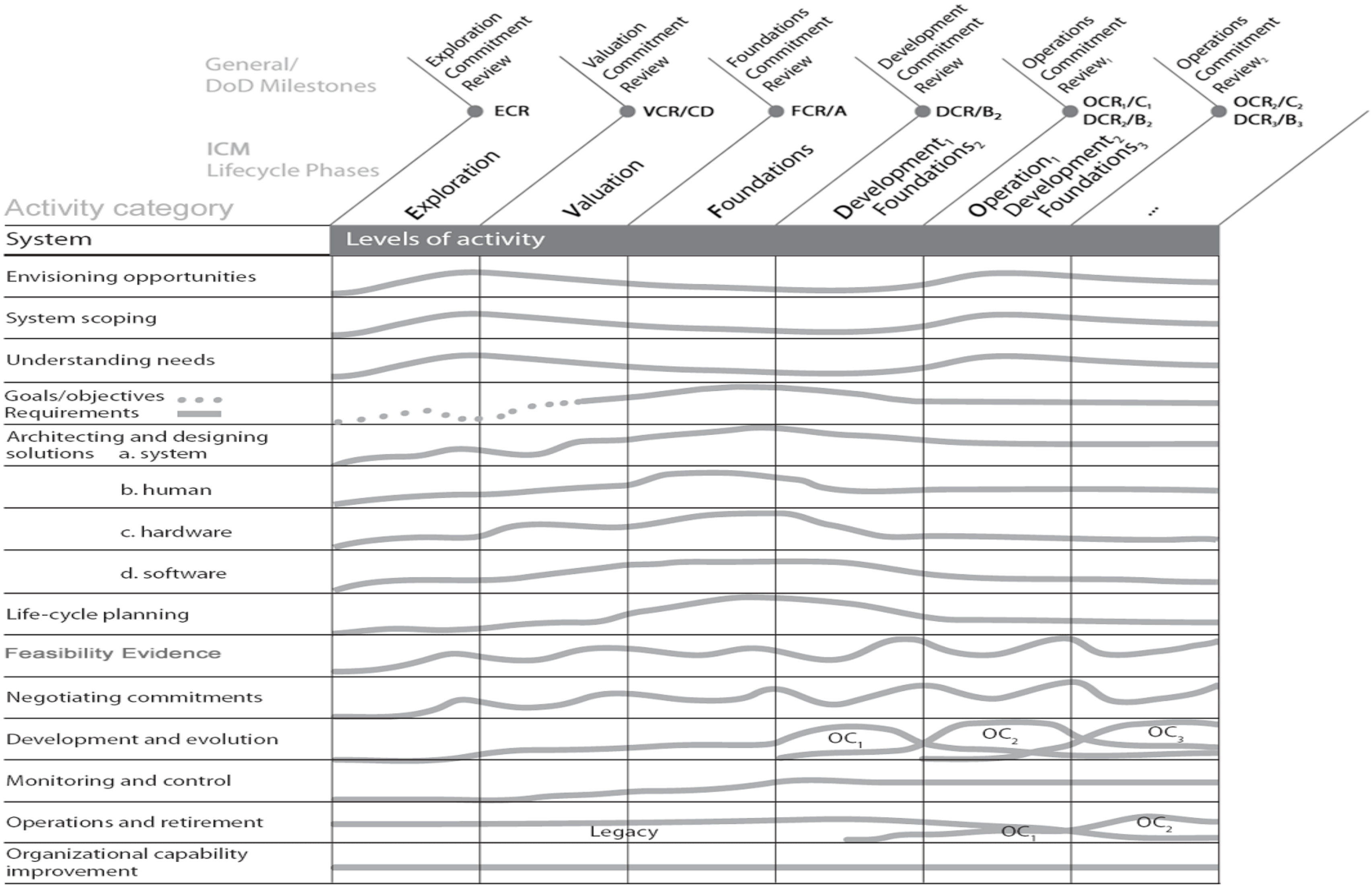


- Need greater investments in narrowing CU
  - Mission, investment, legacy analysis
  - Competitive prototyping
  - Concurrent engineering
  - Associated estimation methods and management metrics
- Larger systems will often have subsystems with narrower CU's

The Incremental Commitment Life Cycle Process: Overview



ICM HSI Levels of Activity for Complex Systems



Nature of FEDs and Anchor Point Milestones

- Evidence provided by developer and validated by independent experts that:

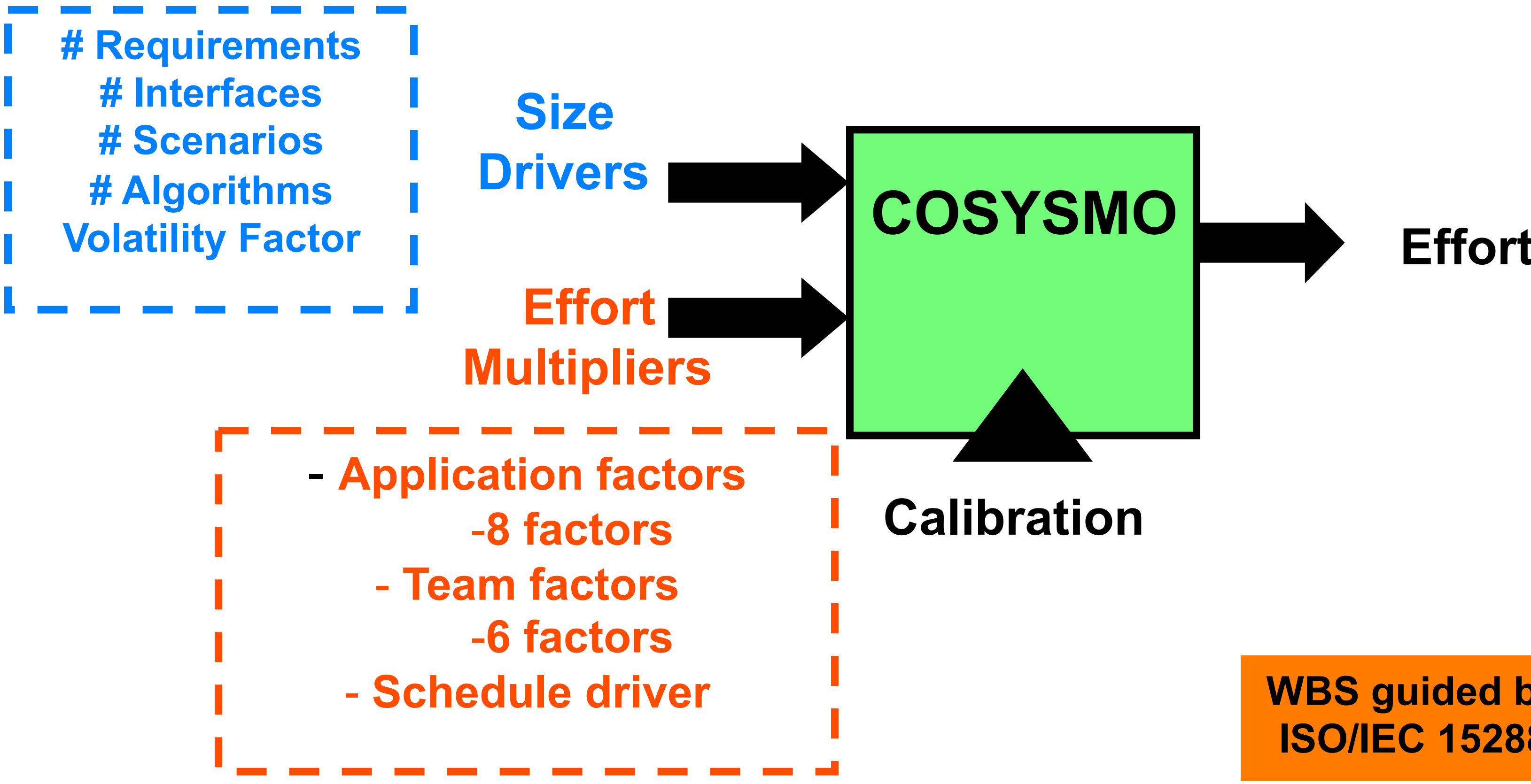
  If the system is built to the specified architecture, it will

  - Satisfy the specified operational concept and requirements
    - Capability, interfaces, level of service, and evolution
  - Be buildable within the budgets and schedules in the plan
  - Generate a viable return on investment
  - Generate satisfactory outcomes for all of the success-critical stakeholders
- Shortfalls in evidence are uncertainties and risks
  - Should be resolved or covered by risk management plans
- Assessed in increasing detail at major anchor point milestones
  - Serves as basis for stakeholders' commitment to proceed
  - Serves to synchronize and stabilize concurrently engineered elements
  - Can be used to strengthen current schedule- or event-based reviews

Key Point: Need to Show Evidence

- Not just traceability matrices and PowerPoint charts
- Evidence can include results of
  - Prototypes: networks, robots, user interfaces, COTS interoperability
  - Benchmarks: performance, scalability, accuracy
  - Exercises: mission performance, interoperability, security
  - Models: cost, schedule, performance, reliability; tradeoffs
  - Simulations: mission scalability, performance, reliability
  - Early working versions: infrastructure, data fusion, legacy compatibility
  - Representative past projects
  - Combinations of the above
- Validated by independent experts
  - Realism of assumptions
  - Representativeness of scenarios
  - Thoroughness of analysis
  - Coverage of key off-nominal conditions
- Much more effort data, product data to collect and analyze

COSYSMO Operational Concept



WBS guided by ISO/IEC 15288

TOC	COSYSMO Application Factor Selection											See Embedded Comments for Descriptions and Selection Criteria
COSYSMO Application Factor Description	Identifier	Current Prod. Range	Suggested Prod. Range	VLOW (VL)	LOW (L)	NOM (N)	HIGH (H)	VHIGH (VH)	XHIGH (XH)	Rating Selected	Resulting Multiplier	Application Factor Rating Selection Comments
Requirements Understanding	RQMT	1.73	1.73	1.40	1.20	1.00	0.90	0.81	****	N	1.00	
Architecture Complexity	ARCH	1.66	1.66	1.28	1.14	1.00	0.88	0.77	****	N	1.00	
Level of Service (KPP) Requirements	LSVC	2.50	2.50	0.66	0.83	1.00	1.33	1.65	****	N	1.00	
Migration Complexity	MIGR	1.50	1.50	****	****	1.00	1.25	1.50	****	N	1.00	
No. and Diversity of Installations/Platforms	INST	1.50	1.50	****	****	1.00	1.25	1.50	****	N	1.00	
No. of Recursive Levels in the Design	RECU	1.50	1.50	0.82	0.91	1.00	1.12	1.23	****	N	1.00	
Documentation to Match Lifecycle Needs	DOCU	0.67	0.67	0.82	0.91	1.00	1.12	1.23	****	N	1.00	
Technology Maturity	TMAT	2.50	2.50	1.75	1.37	1.00	0.85	0.70	****	N	1.00	

Productivity Range (PR) is the Highest Number / Lowest Number and is an indication of the "Relative Degree of Influence" of this parameter on SE effort as currently

The "Suggested" column has no immediate impact in the COSYSMO SE Costing Mode. However, for the COSYSMO SE Data Collection Mode, it serves as a means of collecting your inputs as to what you think the "Relative Degree of Influence" of this parameter should be based upon your overall experience (not specific to the past program being characterized). If you agree with the "Current" number, do nothing. If you disagree, simply overwrite the current number with a new number n (n>1.0) in the appropriate cell.

Select the Rating from the pulld that best represents the Rating program being estimated in the Mode or in the SE Data Collectio Rating that best characterizes t program for which you are provi

4. Parameters I / 5. Parameters II / 6a. Staffing Table / 6b. Staffing Chart / 7. Labor Distribution / Local SE Data Repository / 8a. Application Factors / 8t

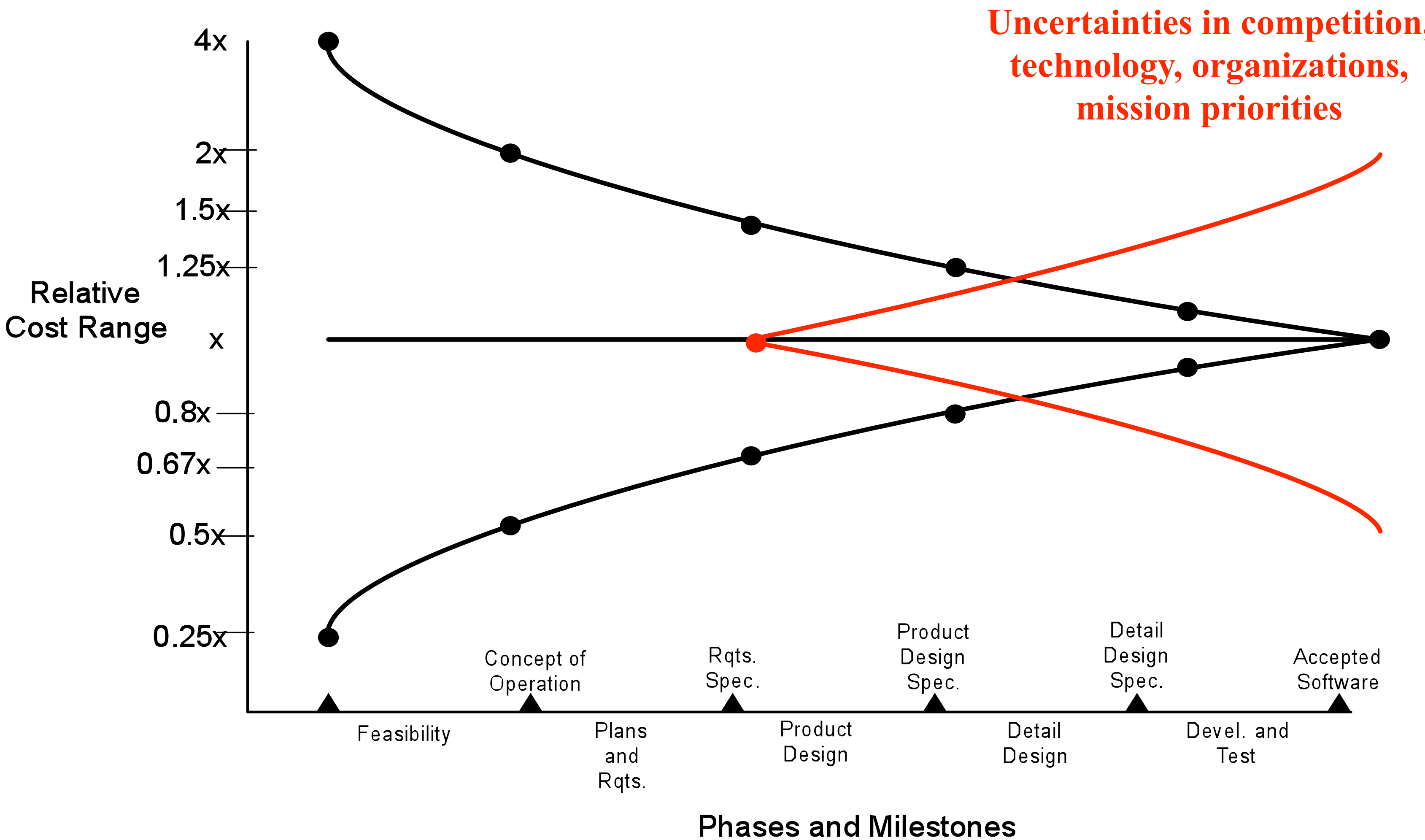
Next-Generation Systems Challenges

- Emergent requirements
  - Example: Virtual global collaboration support systems
  - Need to manage early concurrent engineering
- Rapid change
  - In competitive threats, technology, organizations, environment
- Net-centric systems of systems
  - Incomplete visibility and control of elements
- Model-driven, service-oriented, Brownfield systems
  - New phenomenology, counting rules
- Always-on, never-fail systems
  - Need to balance agility and discipline

Rapid Change Creates a Late Cone of Uncertainty

– Need evolutionary/incremental vs. one-shot development

- - No simple boundary between development and maintenance



Incremental Development Productivity Decline (IDPD)

- Example: Site Defense BMD Software
  - 5 builds, 7 years, $100M
  - Build 1 productivity over 300 SLOC/person month
  - Build 5 productivity under 150 SLOC/PM
    - Including Build 1-4 breakage, integration, rework
    - 318% change in requirements across all builds
    - IDPD factor = 20% productivity decrease per build
  - Similar trends in later unprecedented systems
  - Not unique to DoD: key source of Windows Vista delays
- Maintenance of full non-COTS SLOC, not ESLOC
  - Build 1: 200 KSLOC new; 200K reused@20% = 240K ESLOC
  - Build 2: 400 KSLOC of Build 1 software to maintain, integrate

"Equivalent SLOC" Paradoxes

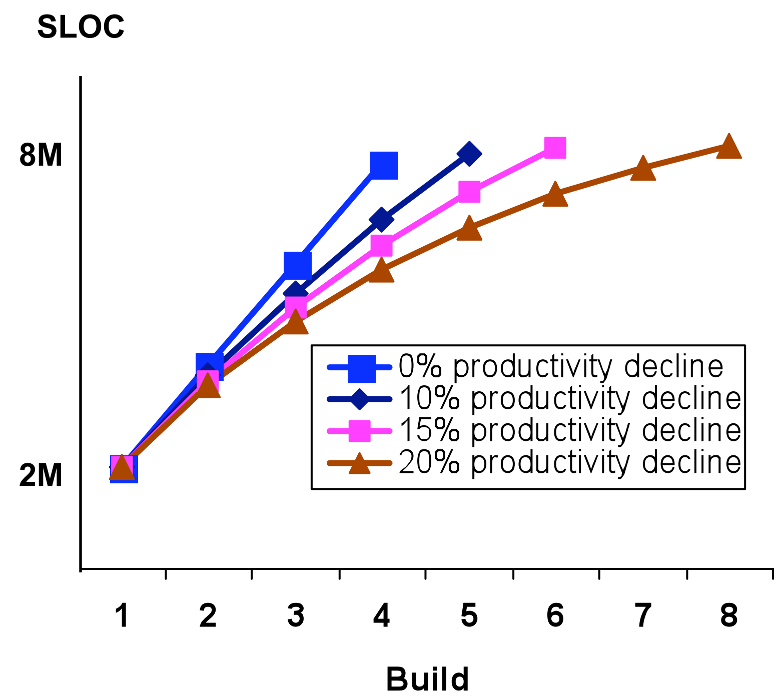
- Not a measure of software size
- Not a measure of software effort
- Not a measure of delivered software capability
- A quantity derived from software component sizes and reuse factors that helps estimate effort
- Once a product or increment is developed, its ESLOC loses its identity
  - Its size expands into full SLOC
  - Some people apply reuse factors to this to determine an ESLOC quantity for the next increment
    - But this has no relation to the product's size

IDPD Cost Drivers: Conservative 4-Increment Example

- Some savings: more experienced personnel (5-20%)
  - Depending on personnel turnover rates
- Some increases: code base growth, diseconomies of scale, requirements volatility, user requests
  - Breakage, maintenance of full code base (20-40%)
  - Diseconomies of scale in development, integration (10-25%)
  - Requirements volatility; user requests (10-25%)
- Best case: 20% more effort (IDPD=6%)
- Worst case: 85% (IDPD=23%)

Effects of IDPD on Number of Increments

- Model relating productivity decline to number of builds needed to reach 8M SLOC Full Operational Capability
- Assumes Build 1 production of 2M SLOC @ 100 SLOC/PM
  - 20000 PM/ 24 mo. = 833 developers
  - Constant staff size for all builds
- Analysis varies the productivity decline per build
  - Extremely important to determine the incremental development productivity decline (IDPD) factor per build



Choosing and Costing Incremental Development Forms

Type	Examples	Pros	Cons	Cost Estimation
Evolutionary Sequential	Small: Agile Large: Evolutionary Development	Adaptability to change	Easiest-first; late, costly breakage	Small: Planning-poker-type Large: Parametric with IDPD
Prespecified Sequential	Platform base plus PPPIs	Prespecifiable full-capability requirements	Emergent requirements or rapid change	COINCOMO with no increment overlap
Overlapped Evolutionary	Product lines with ultrafast change	Modular product line	Cross-increment breakage	Parametric with IDPD and Requirements Volatility
Rebaselining Evolutionary	Mainstream product lines; Systems of systems	High assurance with rapid change	Highly coupled systems with very rapid change	COINCOMO, IDPD for development; COSYSMO for rebaselining

IDPD: Incremental Development Productivity Decline, due to earlier increments breakage, increasing code base to integrate

PPPIs: Pre-Planned Product Improvements

COINCOMO: COCOMO Incremental Development Model (COCOMO II book, Appendix B)

COSYSMO: Systems Engineering Cost Model (in-process COSYSMO book)

All Cost Estimation approaches also include expert-judgment cross-check.

Next-Generation Systems Challenges

- **Emergent requirements**
  - Example: Virtual global collaboration support systems
  - Need to manage early concurrent engineering
- **Rapid change**
  - In competitive threats, technology, organizations, environment
- **Net-centric systems of systems**
  - Incomplete visibility and control of elements
- **Model-driven, service-oriented, Brownfield systems**
  - New phenomenology, counting rules
- **Always-on, never-fail systems**
  - Need to balance agility and discipline

Further Attributes of Future Challenges

Type	Examples	Pros	Cons	Cost Estimation
Systems of Systems	•Directed: Future Combat Systems •Acknowledged: Missile Defense Agency	•Interoperability •Rapid Observe-Orient-Decide-Act (OODA) loop	•Often-conflicting partner priorities •Change processing very complex	•Staged hybrid models • Systems engineering: COSYSMO • Multi-organization development costing • Lead Systems integrator costing • Requirements volatility effects •Integration&test: new cost drivers
Model-Driven Development	•Business 4th-generation languages (4GLs) •Vehicle-model driven development	•Cost savings •User-development advantages •Fewer error sources	•Multi-model composition incapabilities •Model extensions for special cases (platform-payload) •Brownfield complexities •User-development V&V	•Models directives as 4GL source code •Multi-model composition similar to COTS integration, Brownfield integration
Brownfield	•Legacy C4ISR System •Net-Centric weapons platform •Multicore-CPU upgrades	•Continuity of service •Modernization of infrastructure •Ease of maintenance	•Legacy re-engineering often complex •Mega-refactoring often complex	•Models for legacy re-engineering, mega-refactoring •Reuse model for refactored legacy

Further Attributes of Future Challenges (Continued)

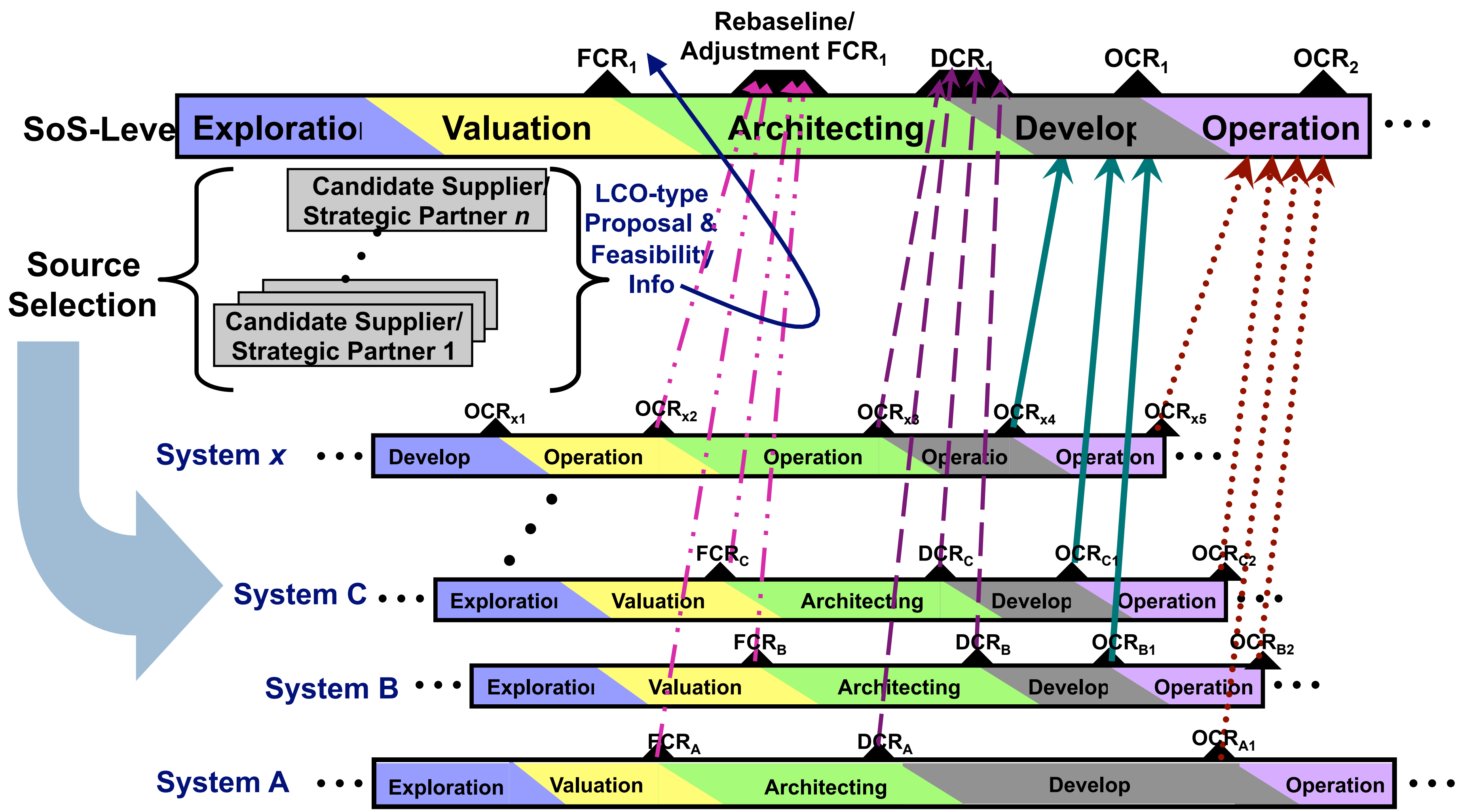
Type	Examples	Pros	Cons	Cost Estimation
Ultrareliable Systems	• Safety-critical systems • Security-critical systems • High-performance real-time systems	• System resilence, survivability • Service-oriented usage opportunities	• Conflicts among attribute objectives • Compatibility with rapid change	• Cost model extensions for added assurance levels • Change impact analysis models

Competitive Prototyping	• Stealth vehicle fly-offs
	• Agent-based RPV control
	• Combinations of challenges
		• Risk buy-down
		• Innovation modification
		• In-depth exploration of alternatives
			• Competitor evaluation often complex
			• Higher up-front cost
			  • But generally good ROI
			• Tech-leveling avoidance often complex
				• Competition preparation, management costing
				  • Evaluation criteria, scenarios, testbeds
				• Competitor budget estimation
				  • Virtual, proof-of-principle, robust prototypes

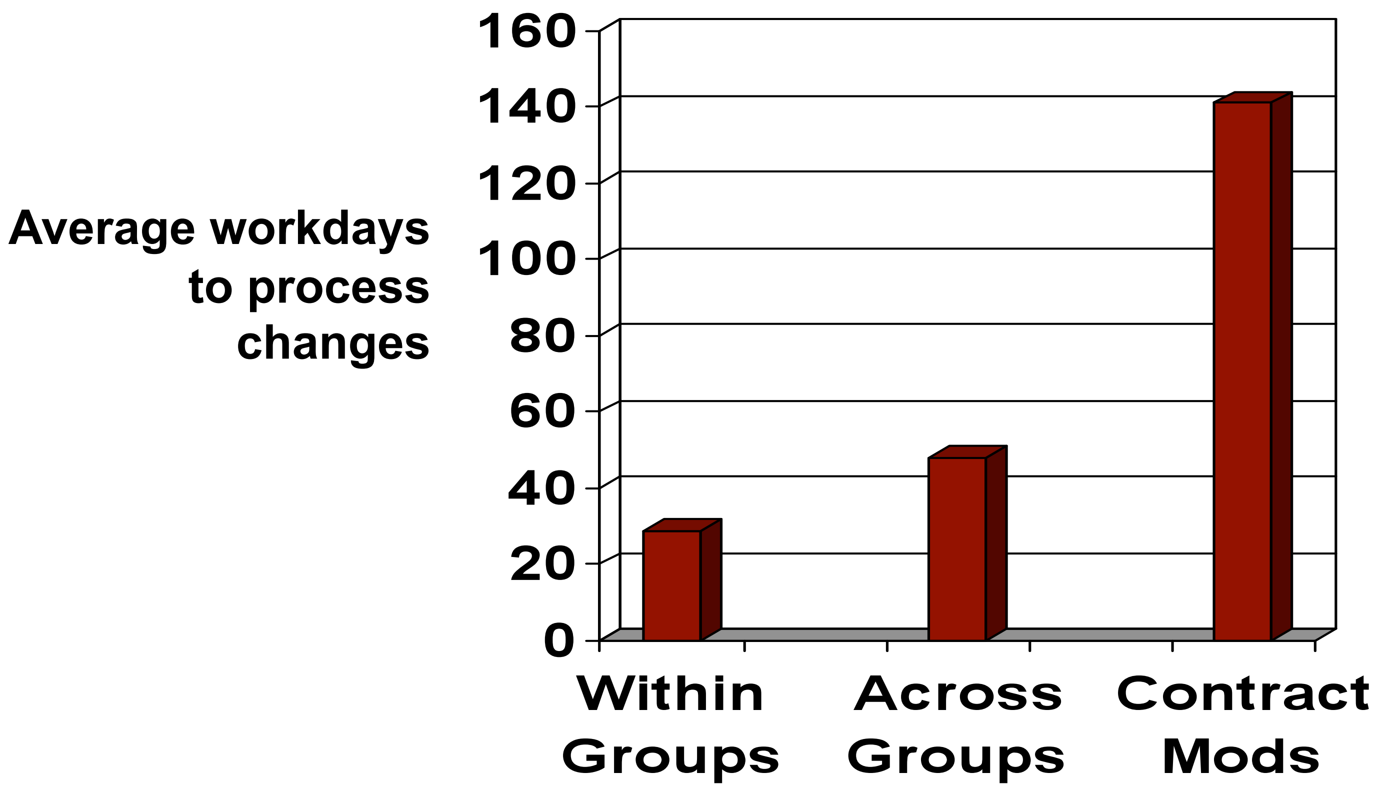
Net-Centric Systems of Systems Challenges

- Need for rapid adaptation to change
  - See first, understand first, act first, finish decisively
- Built-in authority-responsibility mismatches
  - Increasing as authority decreases through Directed, Acknowledged, Collaborative, and Virtual SoS classes
    - Incompatible element management chains, legacy constraints, architectures, service priorities, data, operational controls, standards, change priorities...
- High priority on leadership skills, collaboration incentives, negotiation support such as cost models
  - SoS variety and complexity makes compositional cost models more helpful than one-size-fits-all models

Example: SoSE Synchronization Points

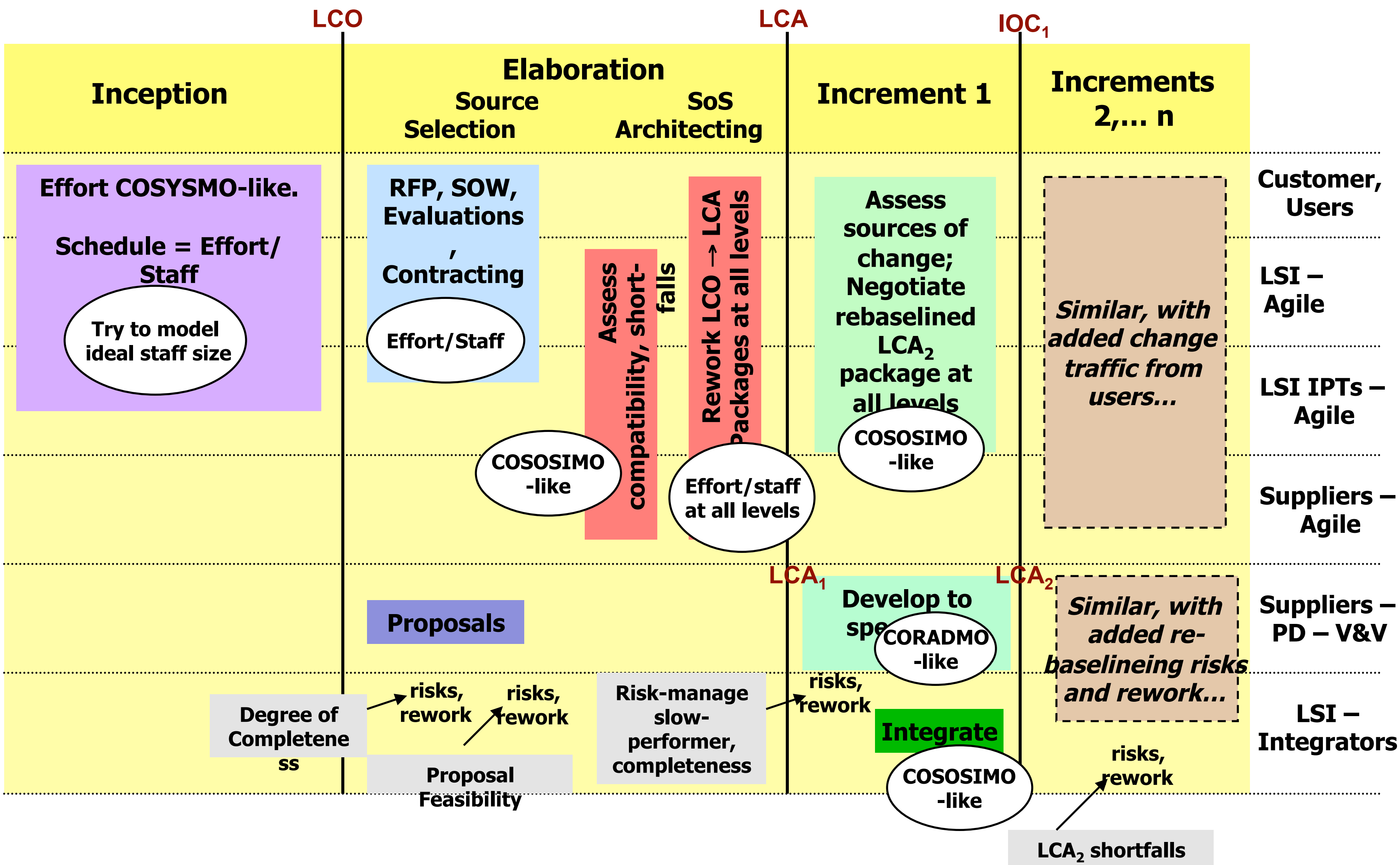


Average Change Processing Time: Two Complex Systems of Systems

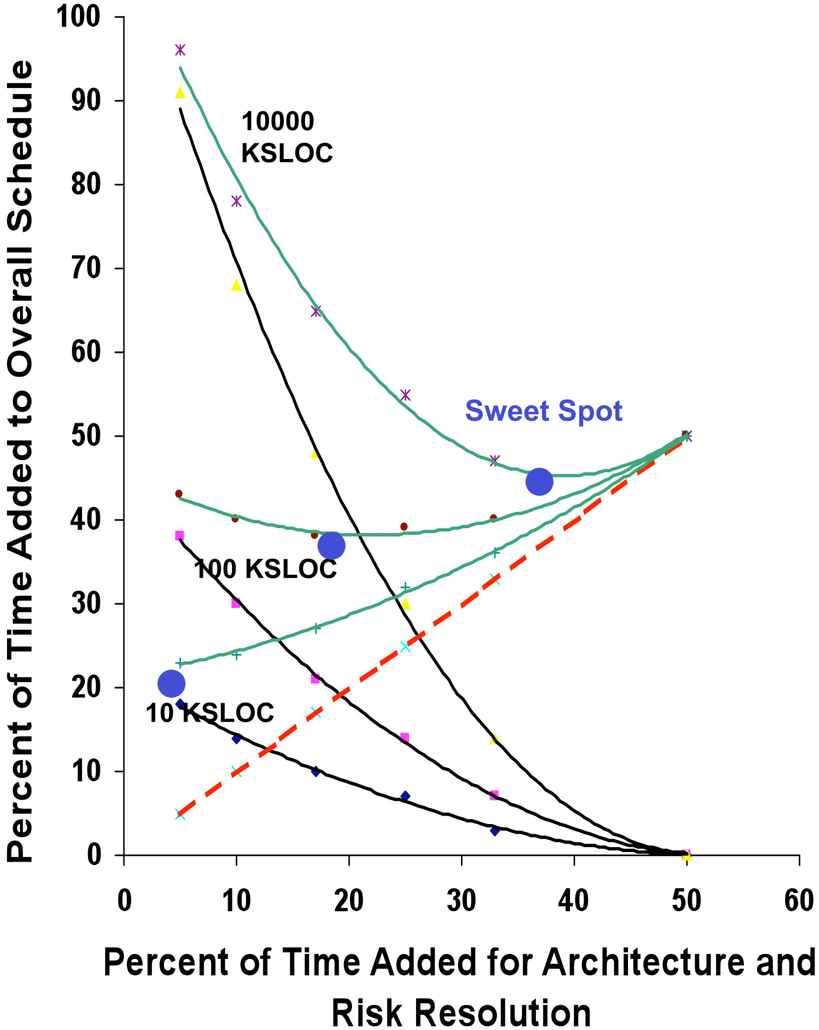


Incompatible with turning within adversary's OODA loop

Compositional approaches: Directed systems of systems



How Much Architecting is Enough?
- Larger projects need more



Sweet Spot Drivers:

Rapid Change: leftward

High Assurance: rightward

Comparison of Cost Model Parameters

Parameter Aspects	COSYSMO	COSOSIMO
Size drivers	# of system requirements # of system interfaces # operational scenarios *# algorithms*	# of SoS requirements # of SoS interface protocols *# of constituent systems* *# of constituent system organizations* # operational scenarios
"Product" characteristics	Size/complexity Requirements understanding Architecture understanding Level of service requirements *# of recursive levels in design* *Migration complexity* Technology risk *#/ diversity of platforms/installations* *Level of documentation*	Size/complexity Requirements understanding Architecture understanding Level of service requirements *Component system maturity and stability* *Component system readiness*
Process characteristics	Process capability *Multi-site coordination* Tool support	Maturity of processes Tool support *Cost/schedule compatibility* SoS risk resolution
People characteristics	Stakeholder team cohesion Personnel/team capability Personnel experience/continuity	Stakeholder team cohesion SoS team capability

Model-Driven, Service-Oriented, Brownfield Systems

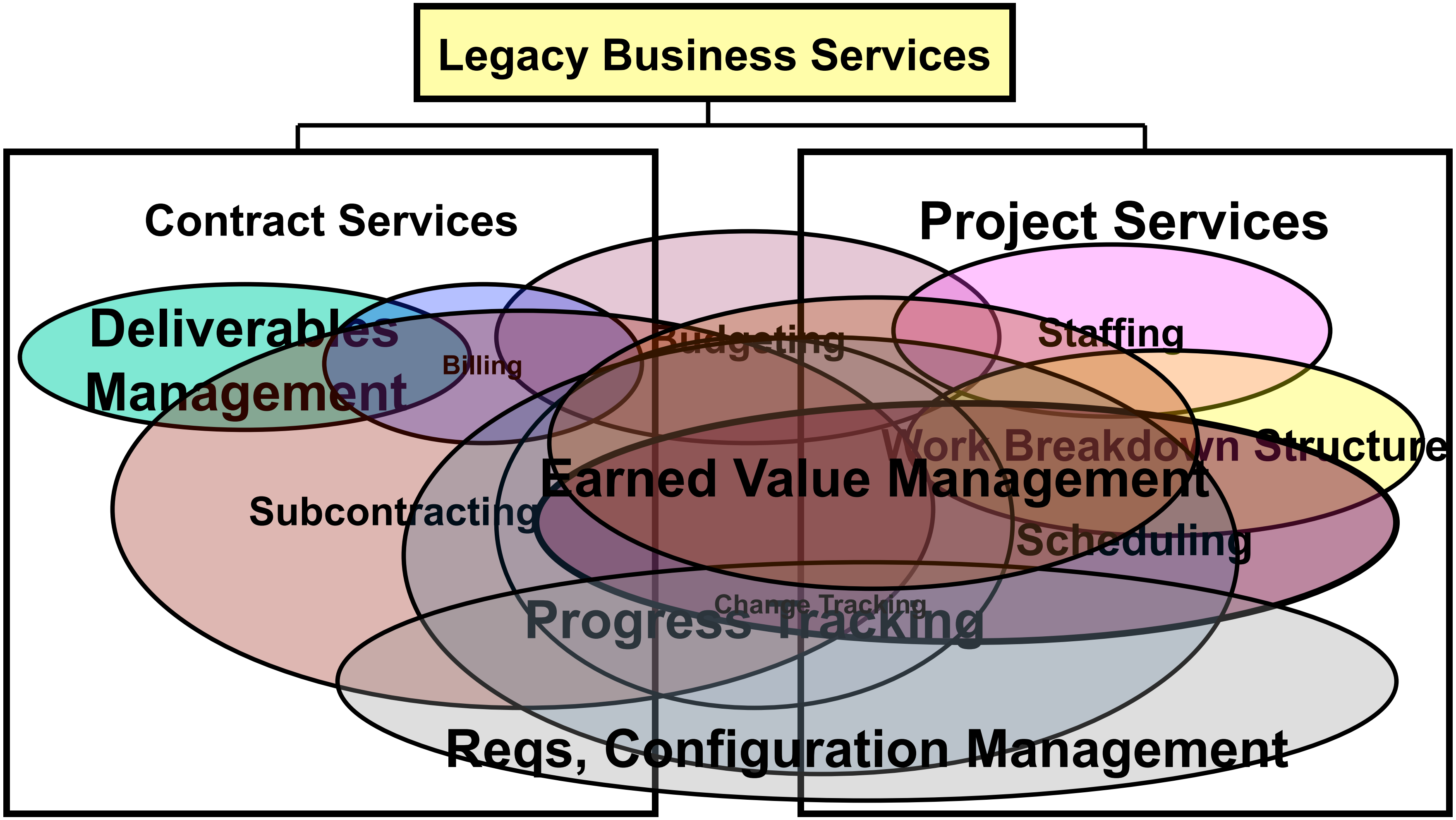
New phenomenology, counting rules

- Product generation from model directives
  - Treat as very high level language: count directives
- Model reuse feasibility, multi-model incompatibilities
  - Use Feasibility Evidence progress tracking measures
- Functional vs. service-oriented architecture mismatches
  - Part-of (one-many) vs. served-by (many-many)
- Brownfield legacy constraints, reverse engineering
  - Reverse-engineer legacy code to fit new architecture
  - Elaborate COSYSMO Migration Complexity cost driver
  - Elaborate COCOMO II reuse model for reverse engineering

Failed Greenfield Corporate Financial System

- Used waterfall approach
  - Gathered requirements
  - Chose best-fit ERP system
  - Provided remaining enhancements
- Needed to ensure continuity of service
  - Planned incremental phase-in of new services
- Failed due to inability to selectively phase out legacy services
  - Dropped after 2 failed tries at cost of $40M

Legacy Systems Patched, Highly Coupled Financial and Non-Financial Services



ICM Approach to Brownfield Engineering

- Understanding needs
  - Analysis of legacy system difficulties
- Envisioning opportunities
  - Concurrently decouple legacy financial and non-financial services, explore new system phase-in and architecture options
- System scoping and architecting
  - Extract legacy financial, non-financial services
  - Prioritize, plan for incremental financial services phase-in/out
- Feasibility evidence development
  - Successful examples of representative service extractions
  - Evidence of cost, schedule, performance feasibility

Result of Legacy Re-engineering

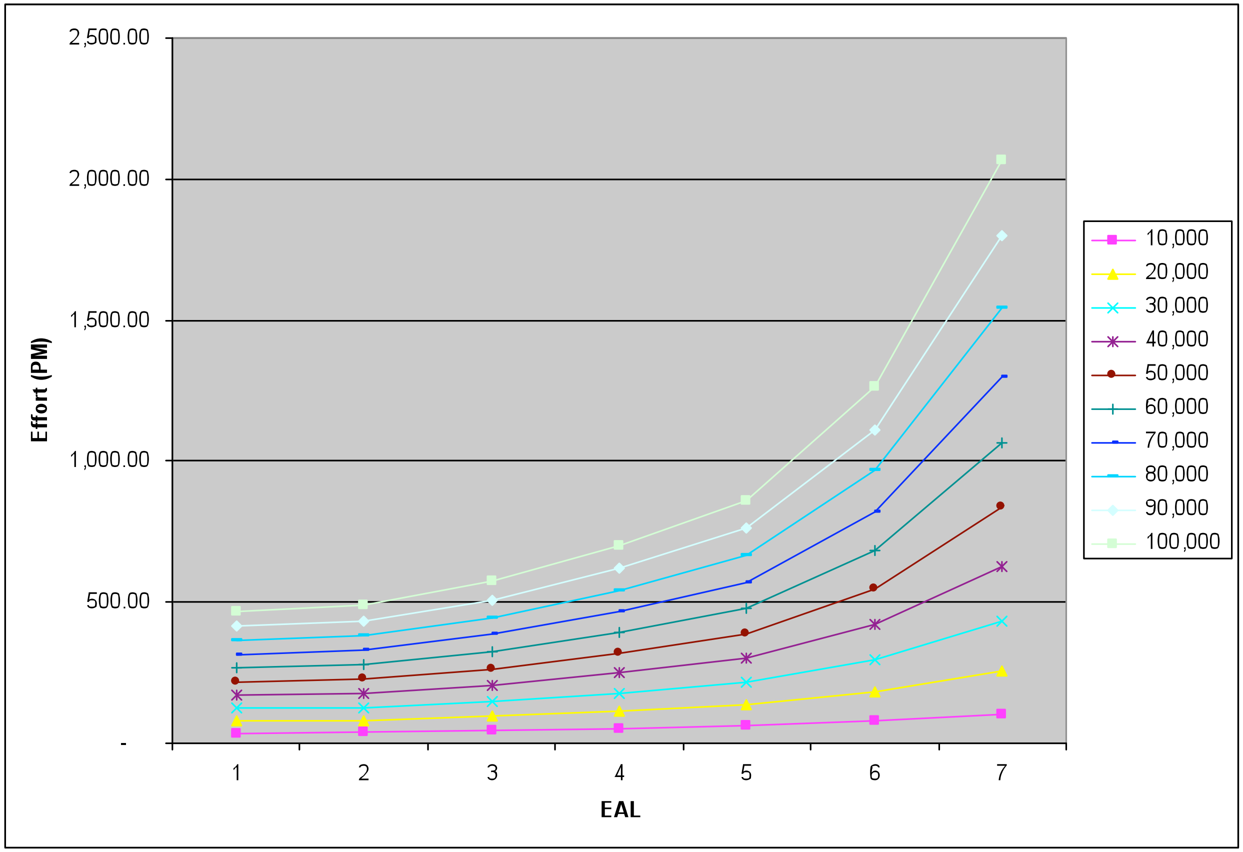
Legacy Business Services

- **Contract Services**
  - **Contract Financial Services**
    - Billing
    - Subcontract payments
    - …
  - **Contract Non-Financial Services**
    - Deliverables mgmt.
    - Terms compliance
    - …
- **General Financial Services**
  - Accounting
  - Budgeting
  - Earned value
  - Payroll
  - …
- **General Non-Financial Services**
  - Progress tracking
  - Change tracking
  - …
- **Project Services**
  - **Project Financial Services**
    - WBS
    - Expenditure categories
    - …
  - **Project Non-Financial Services**
    - Scheduling
    - Staffing
    - Reqs CM
    - …

Always-on, never-fail systems

- Consider using "weighted SLOC" as a productivity metric
- Some SLOC are "heavier to move into place" than others
  - And largely management uncontrollables
  - Examples: high values of COCOMO II cost drivers
    - RELY: Required Software Reliability
    - DATA: Database Size
    - CPLX: Software Complexity
    - DOCU: Required Documentation
    - RUSE: Required Development for Future Reuse
    - TIME: Execution Time Constraint
    - STOR: Main Storage Constraint
    - SCED: Required Schedule Compression
- Provides way to compare productivities across projects
  - And to develop profiles of project classes

COSECMO Estimation Trends
Effort by Assurance Levels for Different Size Projects



- Plot of projects where only SECU & effort increasing drivers
- Efforts seem a little low based on values from Orange Book projects

Balancing Agility and Assurance

- No one-size-fits-all estimation and metrics approach
  - Need compositional approach for both phases and components
- ICM decision table provides criteria for component processes, estimation methods, management metrics
  - Agile: Planning poker/ Wideband Delphi; story burndown
  - Architected agile: planning and implementation sprints: agile plus FED preparation estimation and progress monitoring
  - Mission platforms: hardware, software cost models plus FED preparation estimation and progress monitoring; Leading Indicators and Macro Risk Tool
  - Systems of Systems: composite estimation models; FED estimation and monitoring; extended Macro Risk Tool

Common Risk-Driven Special Cases of the ICM

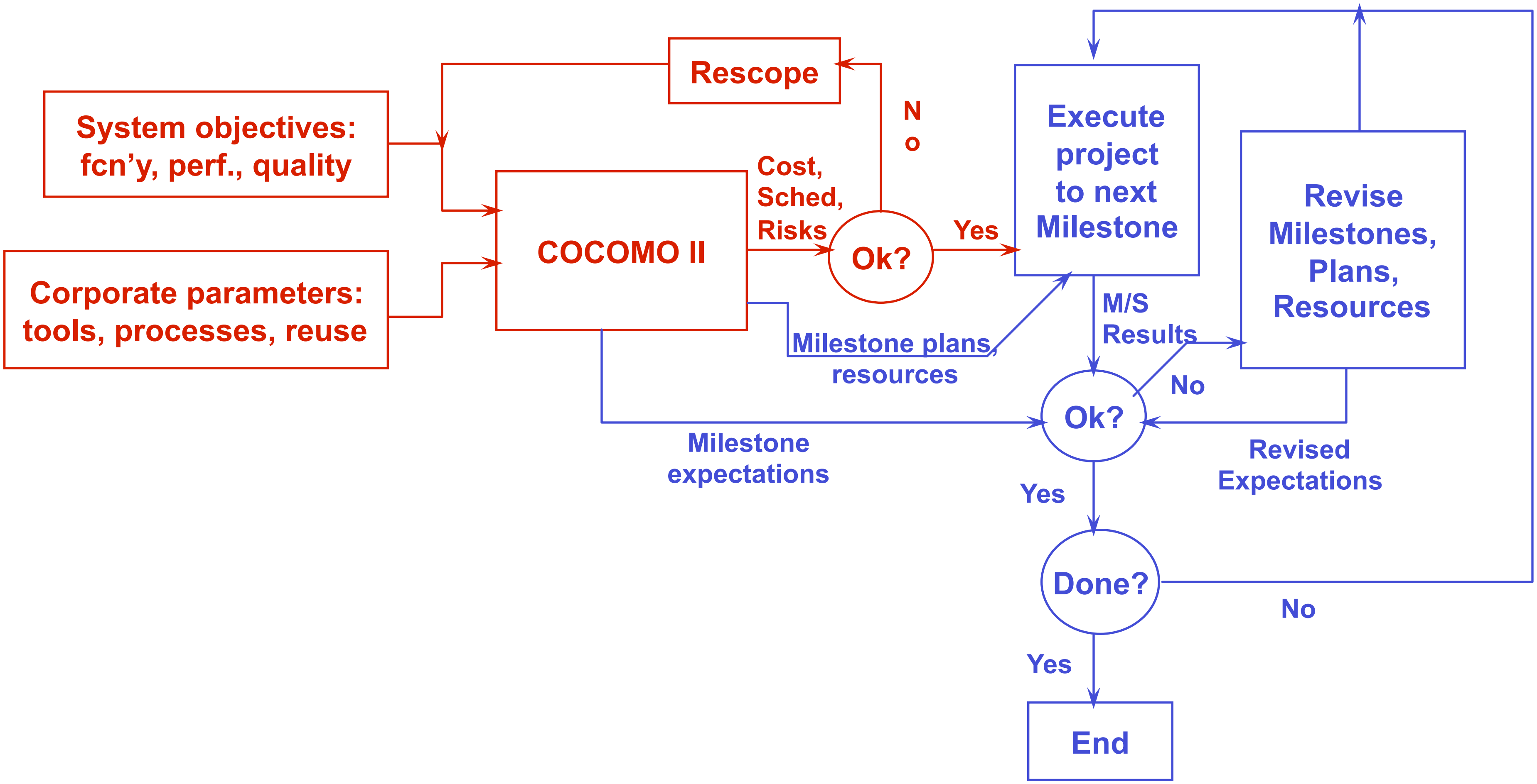
	Special Case	Example	Size, Complexity	Change Rate % /Month	Criticality	NDI Support	Org, Personnel Capability	Key Stage I Activities : Incremental Definition	Key Stage II Activities: Incremental Development, Operations	Time per Build; per Increment
1	Use NDI	Small Accounting				Complete		Acquire NDI	Use NDI	
2	Agile	E-services	Low	1 – 30	Low-Med	Good; in place	Agile-ready Med-high	Skip Valuation , Architecting phases	Scrum plus agile methods of choice	<= 1 day; 2-6 weeks
3	Architected Agile	Business data processing	Med	1 – 10	Med-High	Good; most in place	Agile-ready Med-high	Combine Valuation, Architecting phases. Complete NDI preparation	Architecture-based Scrum of Scrums	2-4 weeks; 2-6 months
4	Formal Methods	Security kernel or safety-critical LSI chip	Low	0.3 – 1	Extra high	None	Strong formal methods experience	Precise formal specification	Formally-based programming language; formal verification	1-5 days; 1-4 weeks
5	HW component with embedded SW	Multi-sensor control device	Low	0.3 – 1	Med-Very High	Good; In place	Experienced; med-high	Concurrent HW/SW engineering. CDR-level ICM DCR	IOC Development, LRIP, FRP. Concurrent Version N+1 engineering	SW: 1-5 days; Market-driven
6	Indivisible IOC	Complete vehicle platform	Med – High	0.3 – 1	High-Very High	Some in place	Experienced; med-high	Determine minimum-IOC likely, conservative cost. Add deferrable SW features as risk reserve	Drop deferrable features to meet conservative cost. Strong award fee for features not dropped	SW: 2-6 weeks; Platform: 6-18 months
7	NDI- Intensive	Supply Chain Management	Med – High	0.3 – 3	Med- Very High	NDI-driven architecture	NDI-experienced; Med-high	Thorough NDI-suite life cycle cost-benefit analysis, selection, concurrent requirements/ architecture definition	Pro-active NDI evolution influencing, NDI upgrade synchronization	SW: 1-4 weeks; System: 6-18 months
8	Hybrid agile / plan-driven system	C4ISR	Med – Very High	Mixed parts: 1 – 10	Mixed parts; Med-Very High	Mixed parts	Mixed parts	Full ICM; encapsulated agile in high change, low-medium criticality parts (Often HMI, external interfaces)	Full ICM ,three-team incremental development, concurrent V&V, next-increment rebaselining	1-2 months; 9-18 months
9	Multi-owner system of systems	Net-centric military operations	Very High	Mixed parts: 1 – 10	Very High	Many NDIs; some in place	Related experience, med-high	Full ICM; extensive multi-owner team building, negotiation	Full ICM; large ongoing system/software engineering effort	2-4 months; 18-24 months
10	Family of systems	Medical Device Product Line	Med – Very High	1 – 3	Med – Very High	Some in place	Related experience, med – high	Full ICM; Full stakeholder participation in product line scoping. Strong business case	Full ICM. Extra resources for first system, version control, multi-stakeholder support	1-2 months; 9-18 months

C4ISR: Command, Control, Computing, Communications, Intelligence, Surveillance, Reconnaissance. **CDR:** Critical Design Review.
DCR: Development Commitment Review. **FRP:** Full-Rate Production. **HMI:** Human-Machine Interface. **HW:** Hard ware.
IOC: Initial Operational Capability. **LRIP:** Low-Rate Initial Production. **NDI:** Non-Development Item. **SW:** Software

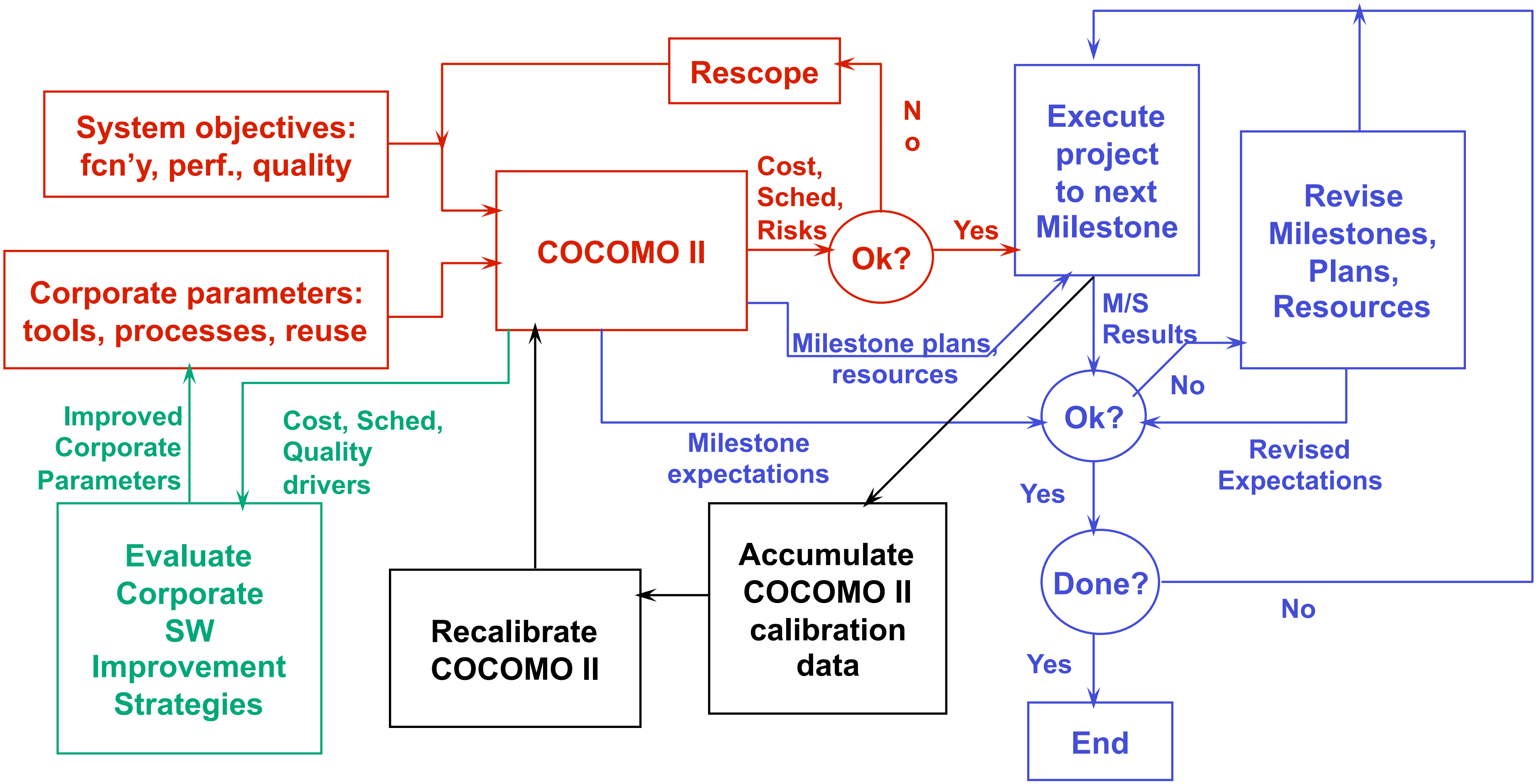
Conclusions

- Future trends imply need to concurrently address new estimation and management metrics challenges
  - Emergent requirements, rapid change, net-centric systems of systems, MDD/SOA/Brownfield, ultrahigh assurance
- Need to work out cost drivers, estimating relationships for new phenomena
  - Incremental Development Productivity Decline (IDPD)
  - ESLOC and milestone definitions
  - Compositional approach for systems of systems
  - NDI, model, and service composability
  - Re-engineering, migration of legacy systems
  - Ultra-reliable systems development
  - Cost/schedule tradeoffs
- Need adaptive data collection & analysis feedback cycle

TRW/COCOMO II Experience Factory: II



TRW/COCOMO II Experience Factory: IV



References

Boehm, B., "Some Future Trends and Implications for Systems and Software Engineering Processes", *Systems Engineering* 9(1), pp. 1-19, 2006.

Boehm, B. and Lane J., "21st Century Processes for Acquiring 21st Century Software-Intensive Systems of Systems." *CrossTalk*: Vol. 19, No. 5, pp.4-9, 2006.

Boehm, B., and Lane, J., "Using the ICM to Integrate System Acquisition, Systems Engineering, and Software Engineering," *CrossTalk*, October 2007, pp. 4-9.

Boehm, B., Brown, A.W.. Clark, B., Madachy, R., Reifer, D., et al., *Software Cost Estimation with COCOMO II*, Prentice Hall, 2000.

Dahmann, J. (2007); "Systems of Systems Challenges for Systems Engineering", Systems and Software Technology Conference, June 2007.

Department of Defense (DoD), *Defense Acquisition Guidebook, version 1.6, http://akss.dau.mil/dag/, 2006*.

Department of Defense (DoD), *Instruction 5000.2, Operation of the Defense Acquisition System,* May 2003.

Department of Defense (DoD), *Systems Engineering Plan Preparation Guide, USD(AT&L),* 2004.

Galorath, D., and Evans, M., Software Sizing, Estimation, and Risk Management, Auerbach, 2006.

Lane, J. and Boehm, B., "Modern Tools to Support DoD Software-Intensive System of Systems Cost Estimation, DACS State of the Art Report, also Tech Report USC-CSSE-2007-716

Lane, J., Valerdi, R., "Synthesizing System-of-Systems Concepts for Use in Cost Modeling," *Systems Engineering*, Vol. 10, No. 4, December 2007.

Madachy, R., "Cost Model Comparison," Proceedings 21st, COCOMO/SCM Forum, November, 2006, http://csse.usc.edu/events/2006/CIIForum/pages/program.html

Maier, M., "Architecting Principles for Systems-of-Systems"; *Systems Engineering*, Vol. 1, No. 4 (pp 267-284).

Northrop, L., et al., *Ultra-Large-Scale Systems: The Software Challenge of the Future*, Software Engineering Institute, 2006.

Reifer, D., "Let the Numbers Do the Talking," *CrossTalk*, March 2002, pp. 4-8.

Valerdi, R, *Systems Engineering Cost Estimation with COSYSMO*, Wiley, 2009 (to appear)

List of Acronyms

AA	Assessment and Assimilation
AAF	Adaptation Adjustment Factor
AAM	Adaptation Adjustment Modifier
COCOMO	Constructive Cost Model
COSOSIMO	Constructive System of Systems Integration Cost Model
COSYSMO	Constructive Systems Engineering Cost Model
COTS	Commercial Off-The-Shelf
CU	Cone of Uncertainty
DCR	Development Commitment Review
DoD	Department of Defense
ECR	Exploration Commitment Review
ESLOC	Equivalent Source Lines of Code
EVMS	Earned Value Management System
FCR	Foundations Commitment Review
FDN	Foundations, as in FDN Package
FED	Feasibility Evidence Description
GD	General Dynamics
GOTS	Government Off-The-Shelf

List of Acronyms *(continued)*

ICM	Incremental Commitment Model
IDPD	Incremental Development Productivity Decline
IOC	Initial Operational Capability
LCA	Life Cycle Architecture
LCO	Life Cycle Objectives
LMCO	Lockheed Martin Corporation
LSI	Lead System Integrator
MDA	Model-Driven Architecture
NDA	Non-Disclosure Agreement
NDI	Non-Developmental Item
NGC	Northrop Grumman Corporation
OC	Operational Capability
OCR	Operations Commitment Review
OO	Object-Oriented
OODA	Observe, Orient, Decide, Act
O&M	Operations and Maintenance
PDR	Preliminary Design Review
PM	Program Manager

List of Acronyms *(continued)*

RFP	Request for Proposal
SAIC	Science Applications international Corporation
SLOC	Source Lines of Code
SoS	System of Systems
SoSE	System of Systems Engineering
SRDR	Software Resources Data Report
SSCM	Systems and Software Cost Modeling
SU	Software Understanding
SW	Software
SwE	Software Engineering
SysE	Systems Engineering
Sys Engr	Systems Engineer
S&SE	Systems and Software Engineering
ToC	Table of Contents
USD (AT&L)	Under Secretary of Defense for Acquisition, Technology, and Logistics
VCR	Validation Commitment Review
V&V	Verification and Validation
WBS	Work Breakdown Structure