The Incremental Commitment Model (ICM), with Ground Systems Applications

Barry Boehm, USC-CSSE

http://csse.usc.edu (tech report 2009-500)

Presented at

GSAW 2009, March 25, 2009

Outline

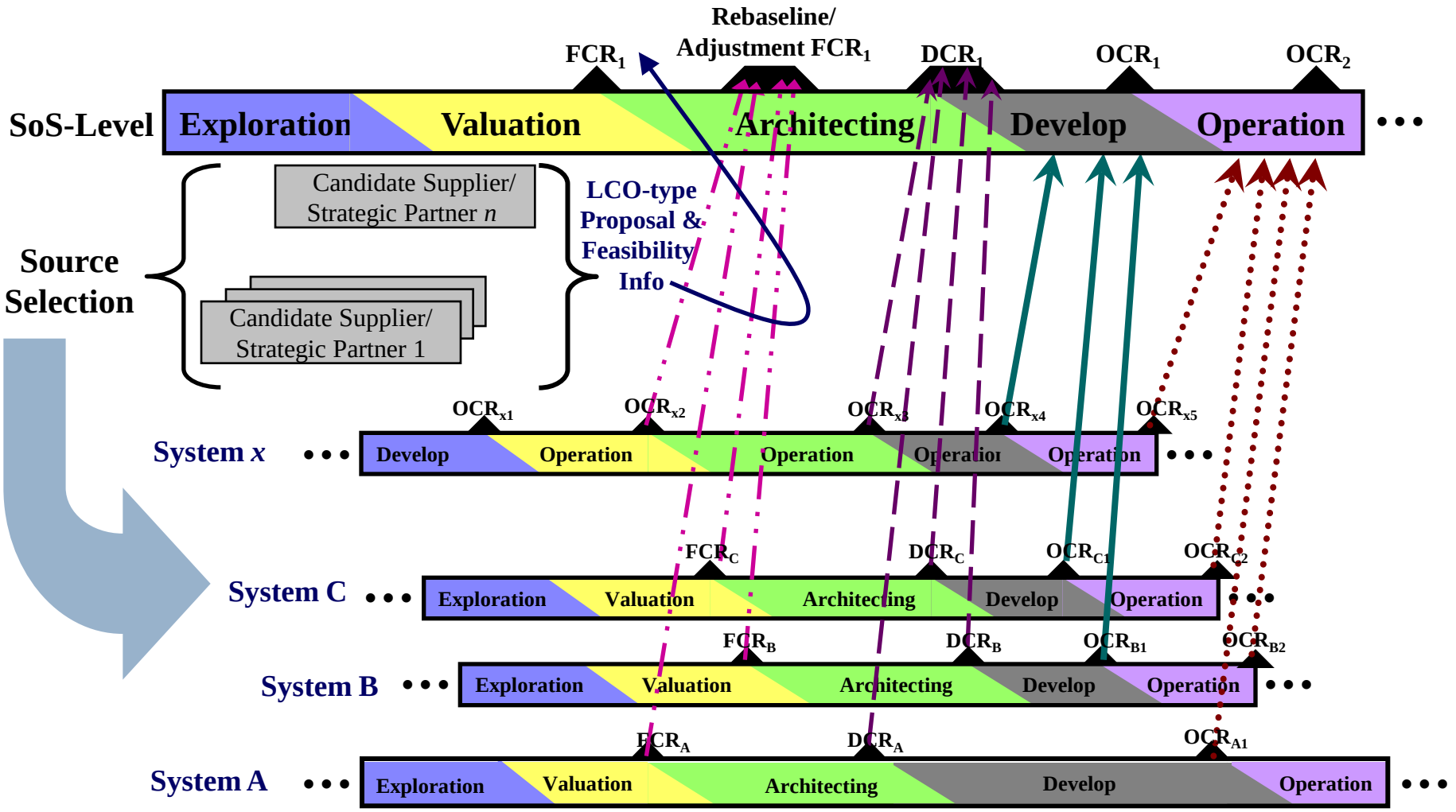
- ***Challenges for developing next-generation ground systems***
- **Overview of ICM**
  - **How ICM implements the new DoDI 5000.02**
- **Risk-based balance of agility and assurance**
- **ICM process decision table**
- **Ground system example for using the ICM**
- **Conclusions**

Future Ground Systems Challenges-I

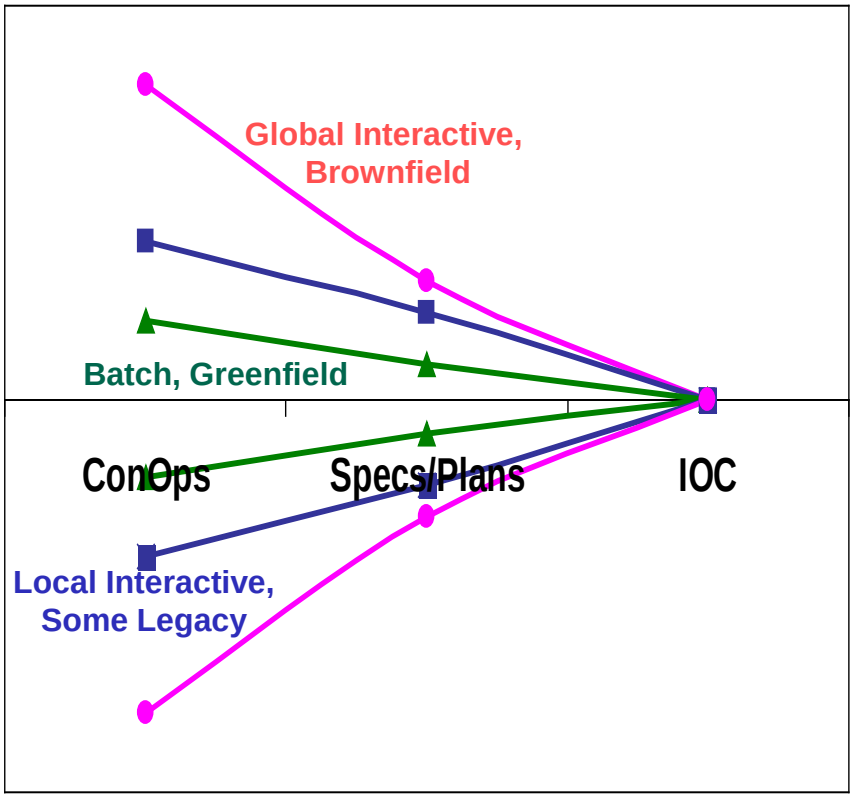
- **Multi-owner, multi-mission systems of systems**
  - **Ground system must simultaneously interoperate with a wide variety of independently evolving Service, joint, interagency, and commercial systems of systems**
  - **Need to satisfice among multiple stakeholders**
  - **No one-size-fits-all solutions or processes**

- **Emergence and human-intensiveness**
  - **Requirements not pre-specifiable**
  - **Budgets and schedules not pre-specifiable**
  - **Need for evolutionary growth**
  - **Need to manage uncertainty and risk**

Example: SoSE Synchronization Points



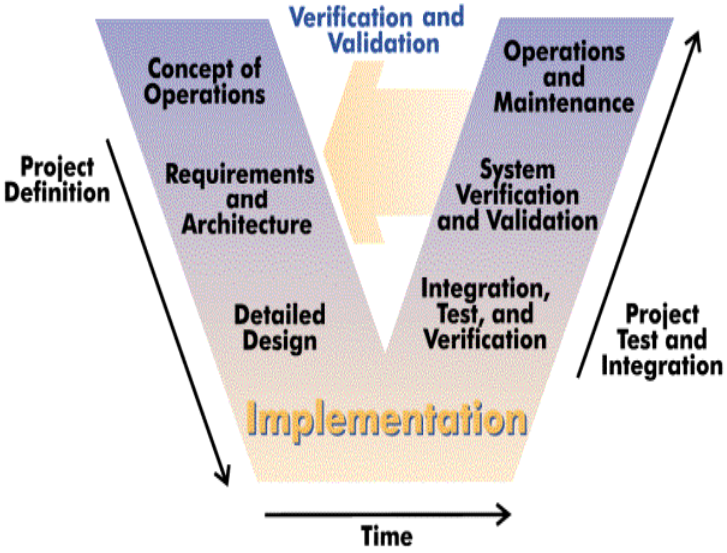
The Broadening Early Cone of Uncertainty (CU)



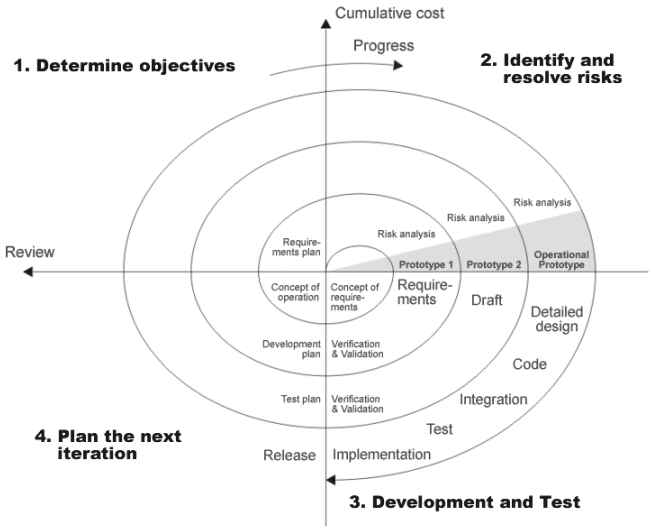
- Need greater investments in narrowing CU
  - Mission, investment, legacy analysis
  - Competitive prototyping
  - Concurrent engineering
  - Associated estimation methods and management metrics
- Larger systems will often have subsystems with narrower CU's

Current System Acquisition Methods
Easy to misinterpret as one-size-fits-all

- **V-Model**[1]



- **Spiral Model**[2]



High level guidance assumes that acquirers have extensive acquisition experience...
Without experience, too easy to misinterpret and auger in with disastrous results...

[1] http://en.wikipedia.org/wiki/V-Model

[2] http://en.wikipedia.org/wiki/Spiral_model

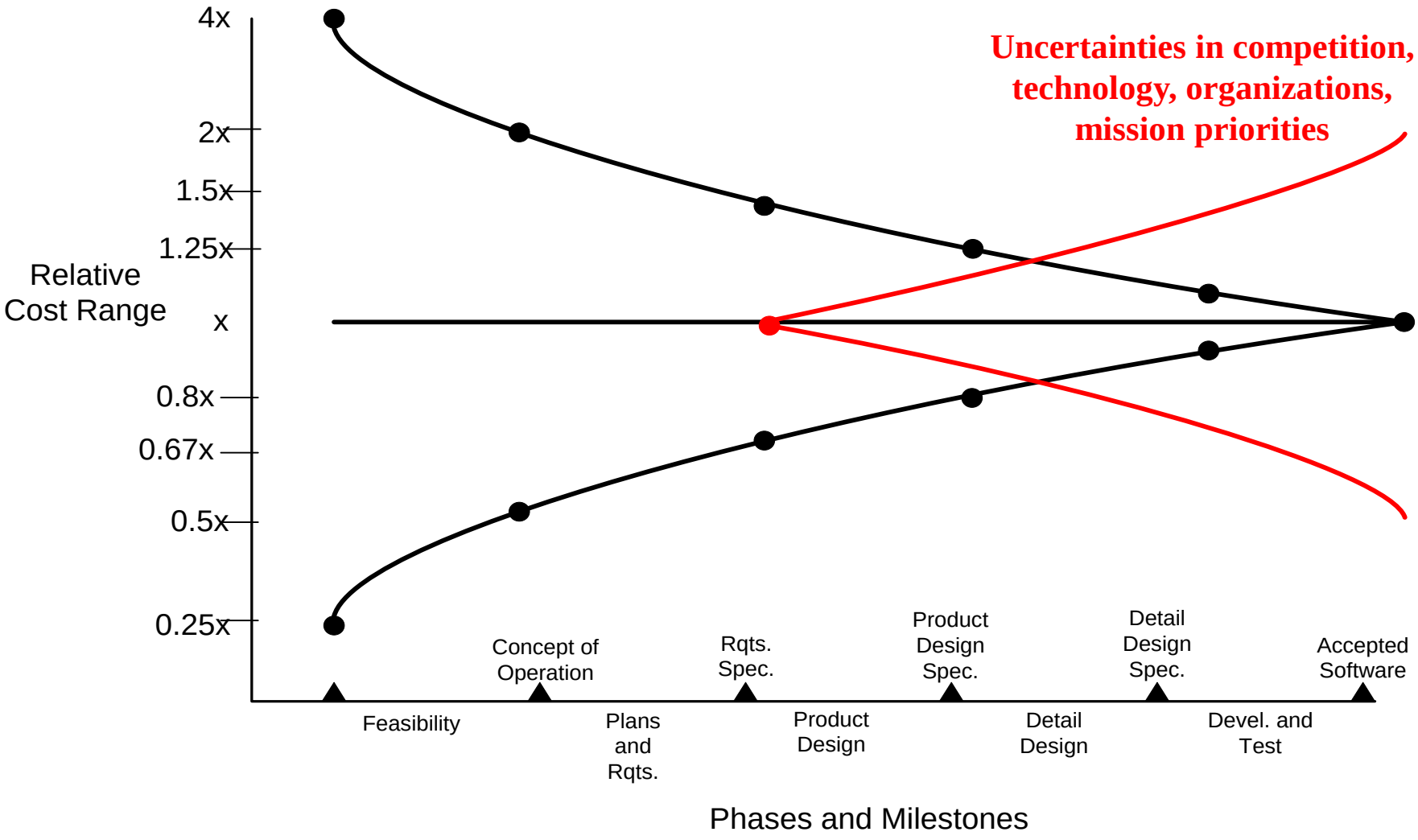
Typical Acquisition Process

- **Military pilot coming off a fighter plane is assigned to manage the acquisition of a new satellite ground system**
  - **Excellent understanding of aircraft operator needs**
  - **No experience with ground system/software development**
  - **Conditioned to plan the flight and fly the plan**
  - **Will interpret V-model diagram sequentially**
  - **Will interpret spiral diagram as one-size-fits-all**

Future Ground System Challenges-II

- Rapid pace of change
  - In competition, mission priorities, technology, Commercial Off-the-Shelf (COTS), environment
  - Need incremental development to avoid obsolescence
  - Need concurrent vs. sequential processes
  - Need both prescience and rapid adaptability
    - Software important; humans more important
- Brownfield vs. Greenfield development
  - Need to provide legacy continuity of service
  - Need to accommodate legacy, OTS constraints
- Always-on, never-fail systems
  - Need well-controlled, high-assurance processes
  - Need to synchronize and stabilize concurrency
  - Need to balance assurance and agility

Rapid Change Creates a Late Cone of Uncertainty
– Need incremental vs. one-shot development



Outline

- Challenges for developing next-generation ground systems
- *Overview of ICM*
  - *How ICM implements the new DoDI 5000.02*
- Risk-based balance of agility and assurance
- ICM process decision table
- Ground system example for using the ICM
- Conclusions

What is the ICM?

- **Risk-driven framework for determining and evolving best-fit system life-cycle process**
- **Integrates the strengths of phased and risk-driven spiral process models**
- **Synthesizes together principles critical to successful system development**
  - **Commitment and accountability of system sponsors**
  - **Success-critical stakeholder satisficing**
  - **Incremental growth of system definition and stakeholder commitment**
  - **Concurrent engineering**
  - **Iterative development cycles**
  - **Risk-based activity levels and anchor point milestones**

Principles trump diagrams…

Principles used by 60-80% of CrossTalk Top-5 projects, 2002-2005

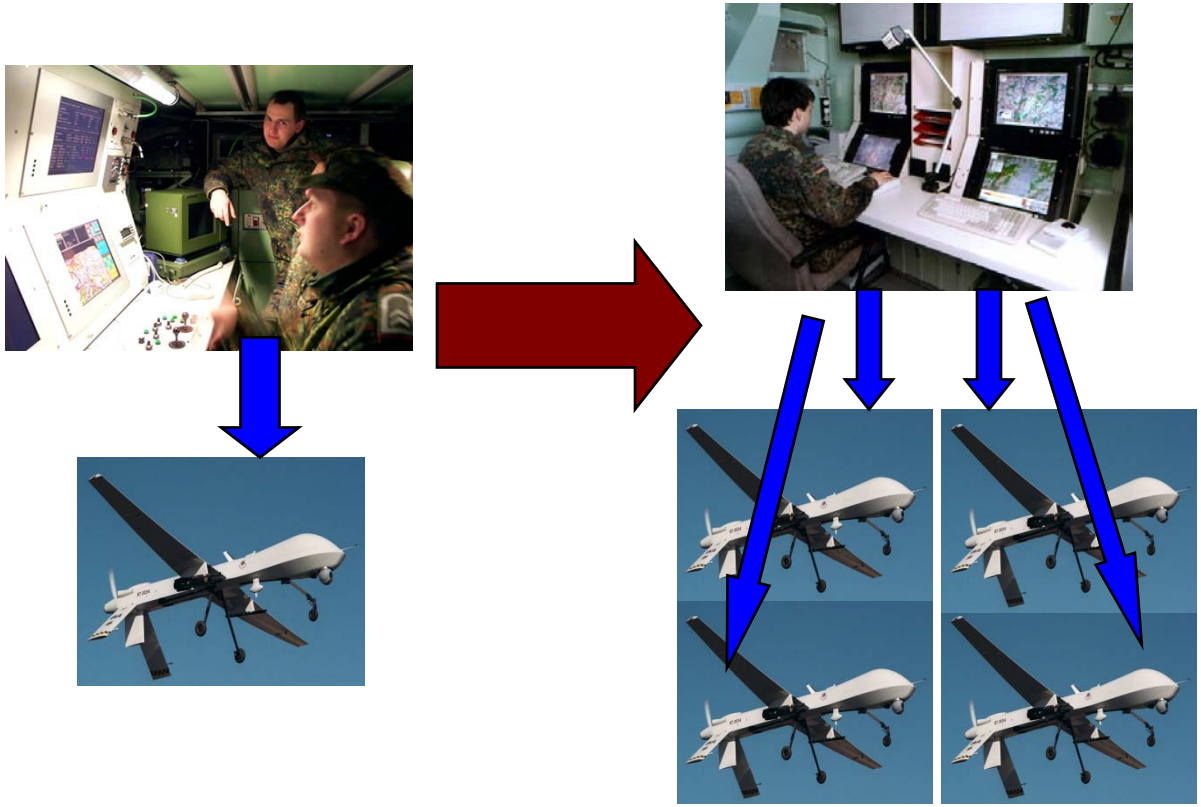
ICM Nature and Origins

- Integrates hardware, software, and human factors elements of systems engineering
  - Concurrent exploration of needs and opportunities
  - Concurrent engineering of hardware, software, human aspects
  - Concurrency stabilized via anchor point milestones
- Developed in response to DoD-related issues
  - Clarify "spiral development" usage in DoD Instruction 5000.2
    - Initial phased version (2005)
  - Explain Future Combat System of systems spiral usage to GAO
    - Underlying process principles (2006)
  - Provide framework for human-systems integration
    - National Research Council report (2007)
- Integrates strengths of current process models
  - But not their weaknesses

Incremental Commitment in Gambling

- Total Commitment: Roulette
  - Put your chips on a number
    - E.g., a value of a key performance parameter
  - Wait and see if you win or lose
- Incremental Commitment: Poker, Blackjack
  - Put some chips in
  - See your cards, some of others' cards
  - Decide whether, how much to commit to proceed

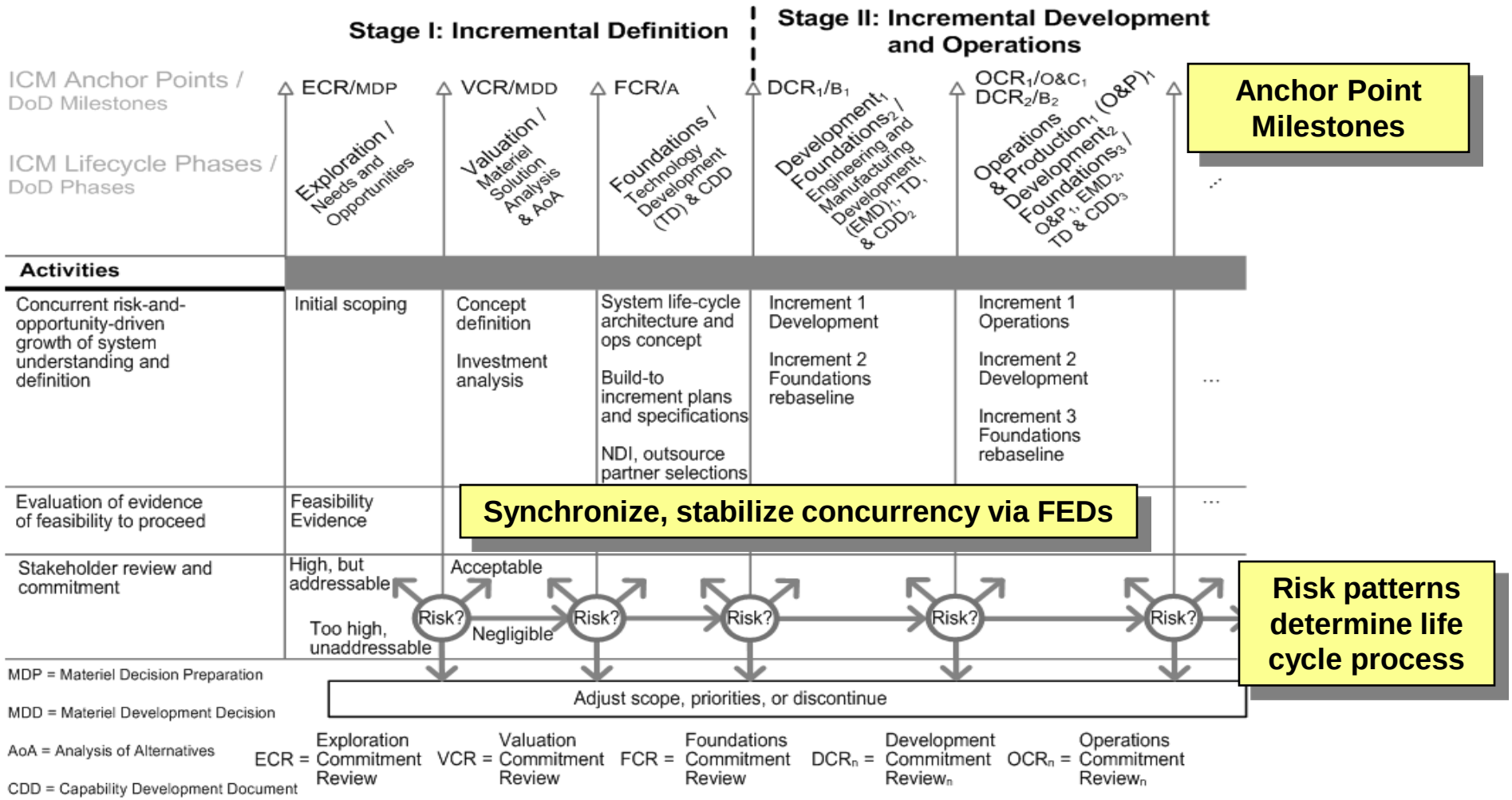
Scalable Remotely Controlled Operations



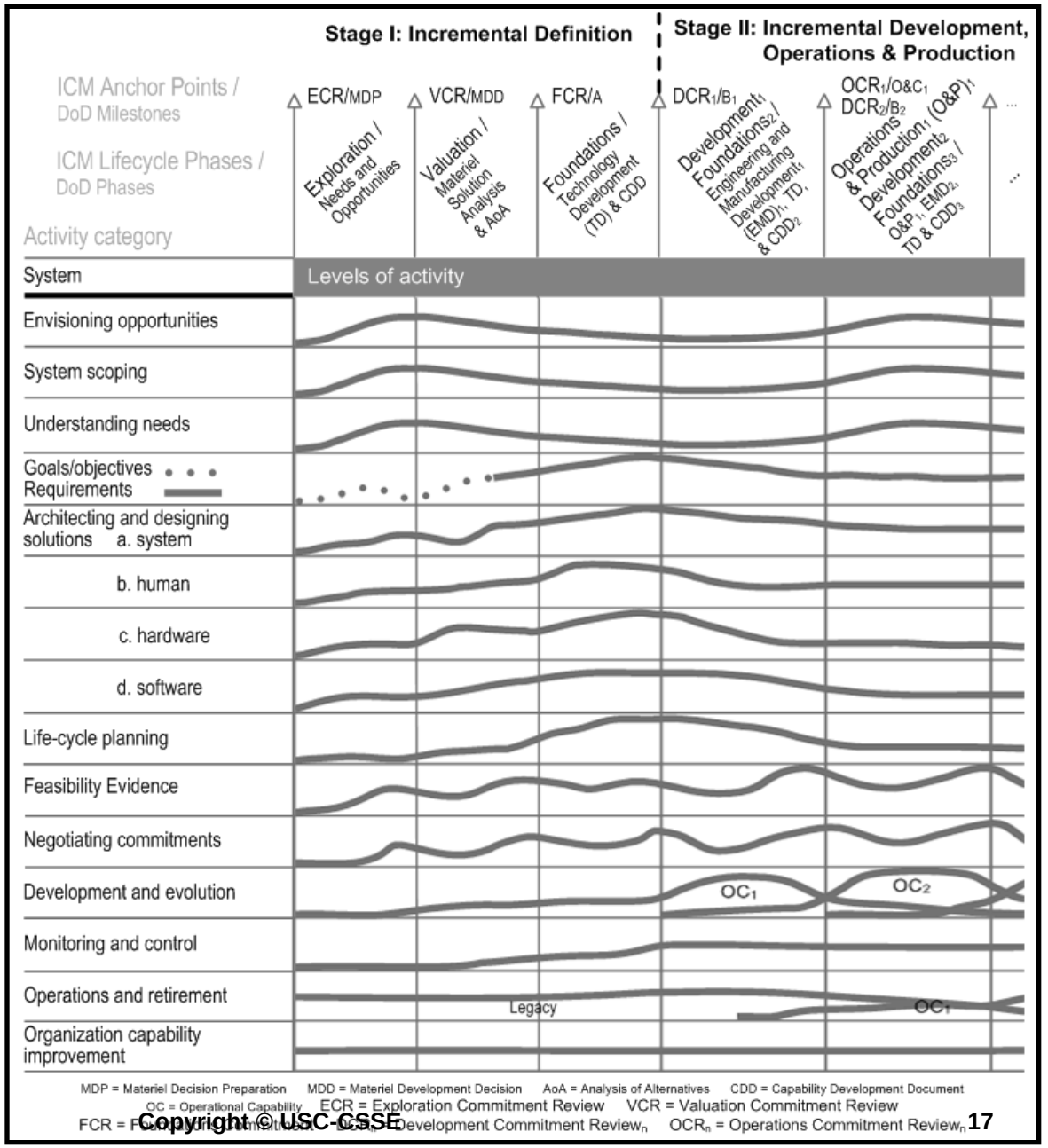
Total vs. Incremental Commitment – 4:1 RPV

- **Total Commitment**
  - **Agent technology demo and PR: Can do 4:1 for $1B**
  - **Winning bidder: $800M; PDR in 120 days; 4:1 capability in 40 months**
  - **PDR: many outstanding risks, undefined interfaces**
  - **$800M, 40 months: "halfway" through integration and test**
  - **1:1 IOC after $3B, 80 months**
- **Incremental Commitment [with a number of competing teams]**
  - **$25M, 6 mo. to VCR [4]: may beat 1:2 with agent technology, but not 4:1**
  - **$75M, 8 mo. to FCR [3]: agent technology may do 1:1; some risks**
  - **$225M, 10 mo. to DCR [2]: validated architecture, high-risk elements**
  - **$675M, 18 mo. to IOC [1]: viable 1:1 capability**
  - **1:1 IOC after $1B, 42 months**

The Incremental Commitment Life Cycle Process: Overview



ICM Activity Levels for Complex Systems



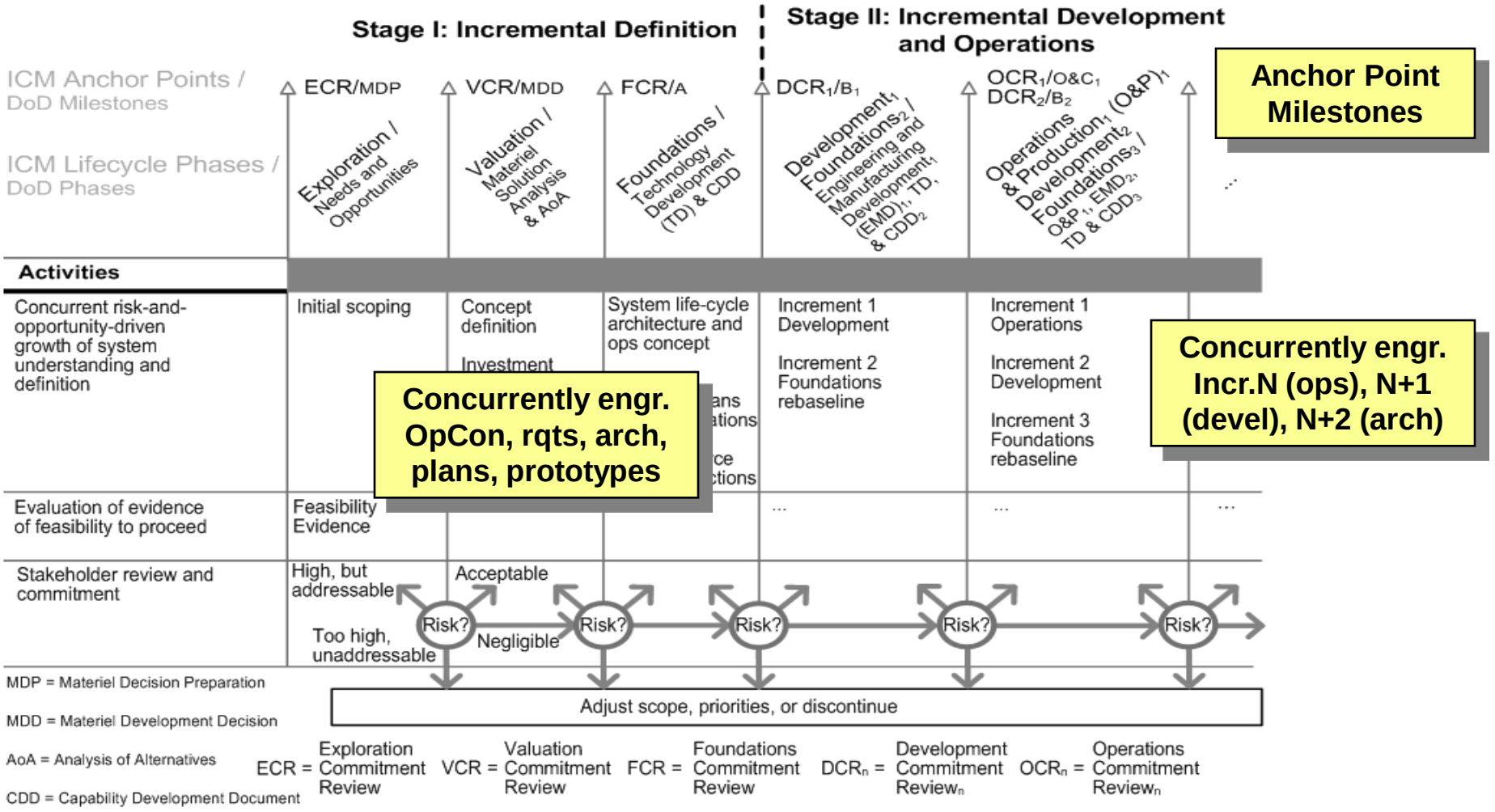
Anchor Point Feasibility Evidence Descriptions

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

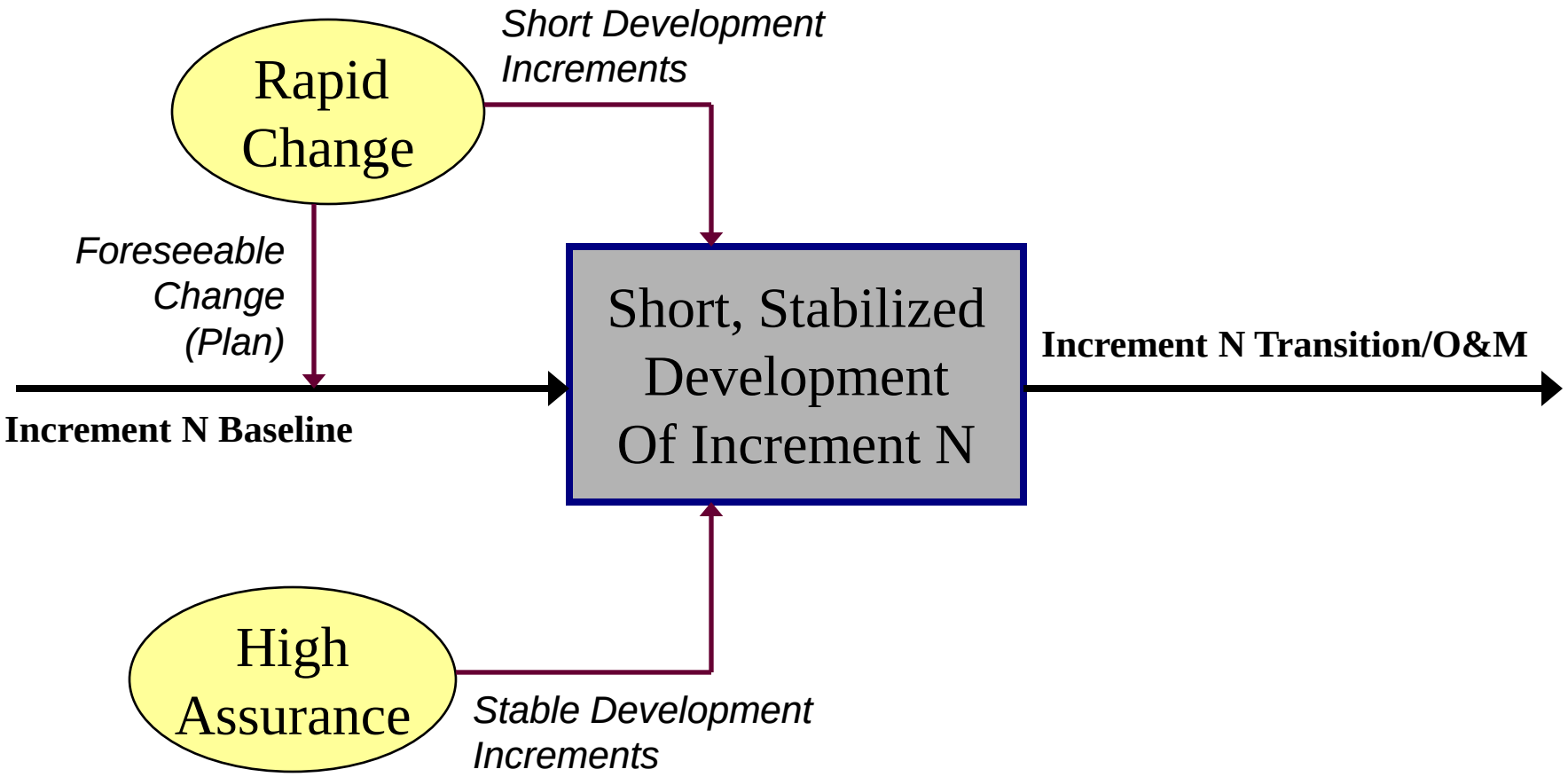
  If the system is built to the specified architecture, it will
  - **Satisfy the requirements: capability, interfaces, level of service, and evolution**
  - **Support the operational concept**
  - **Be buildable within the budgets and schedules in the plan**
  - **Generate a viable return on investment**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

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

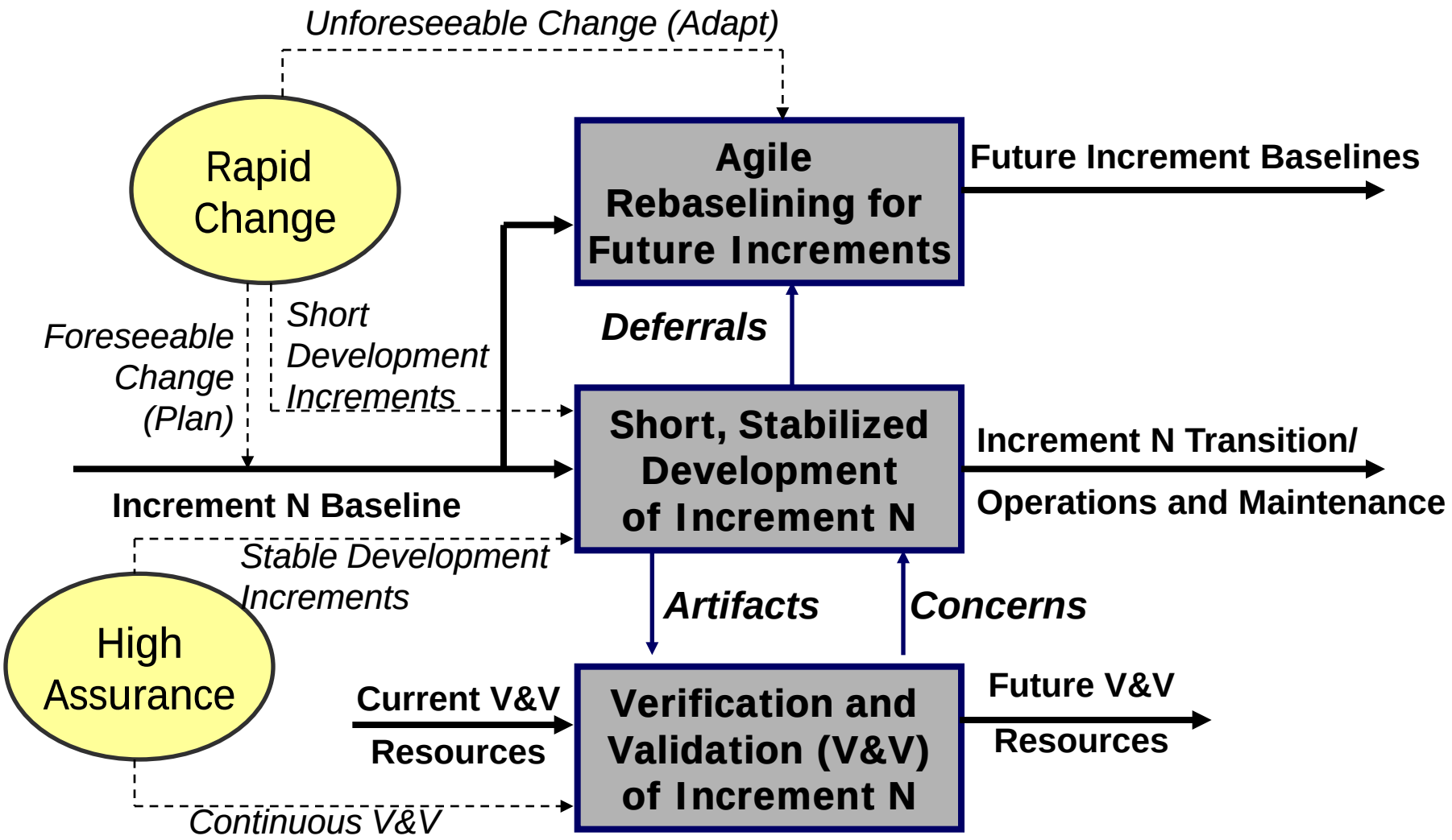
The Incremental Commitment Life Cycle Process: Overview



Risk-Driven Scalable Spiral Model: Increment View



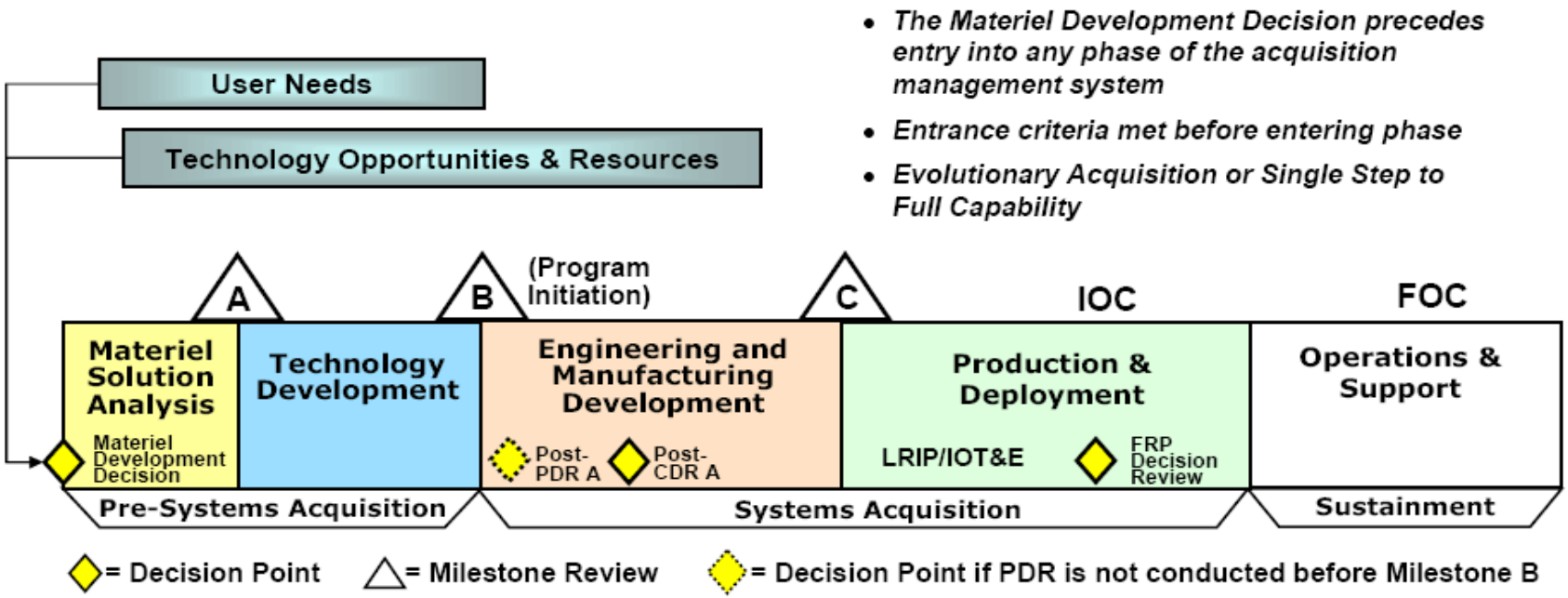
Risk-Driven Scalable Spiral Model: Increment View



ICM Compatibility with New DoDI 5000.02

- Both begin with Needs and Opportunities
- Both emphasize need for Preliminary Design Review before commitment to development
- Both emphasize evolutionary development

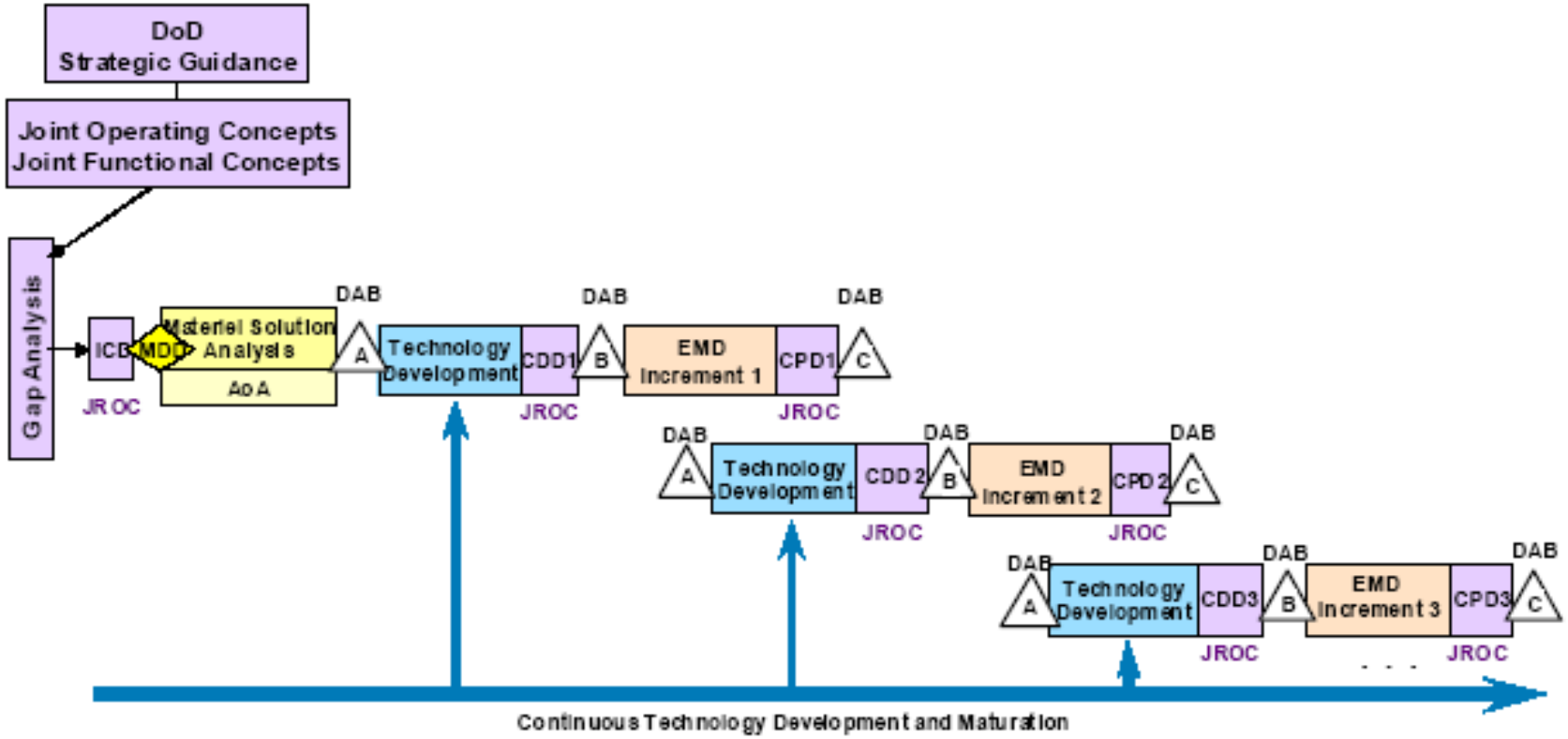
The DoDI 5000.02 Acquisition Life Cycle



- *The Materiel Development Decision precedes entry into any phase of the acquisition management system*
- *Entrance criteria met before entering phase*
- *Evolutionary Acquisition or Single Step to Full Capability*

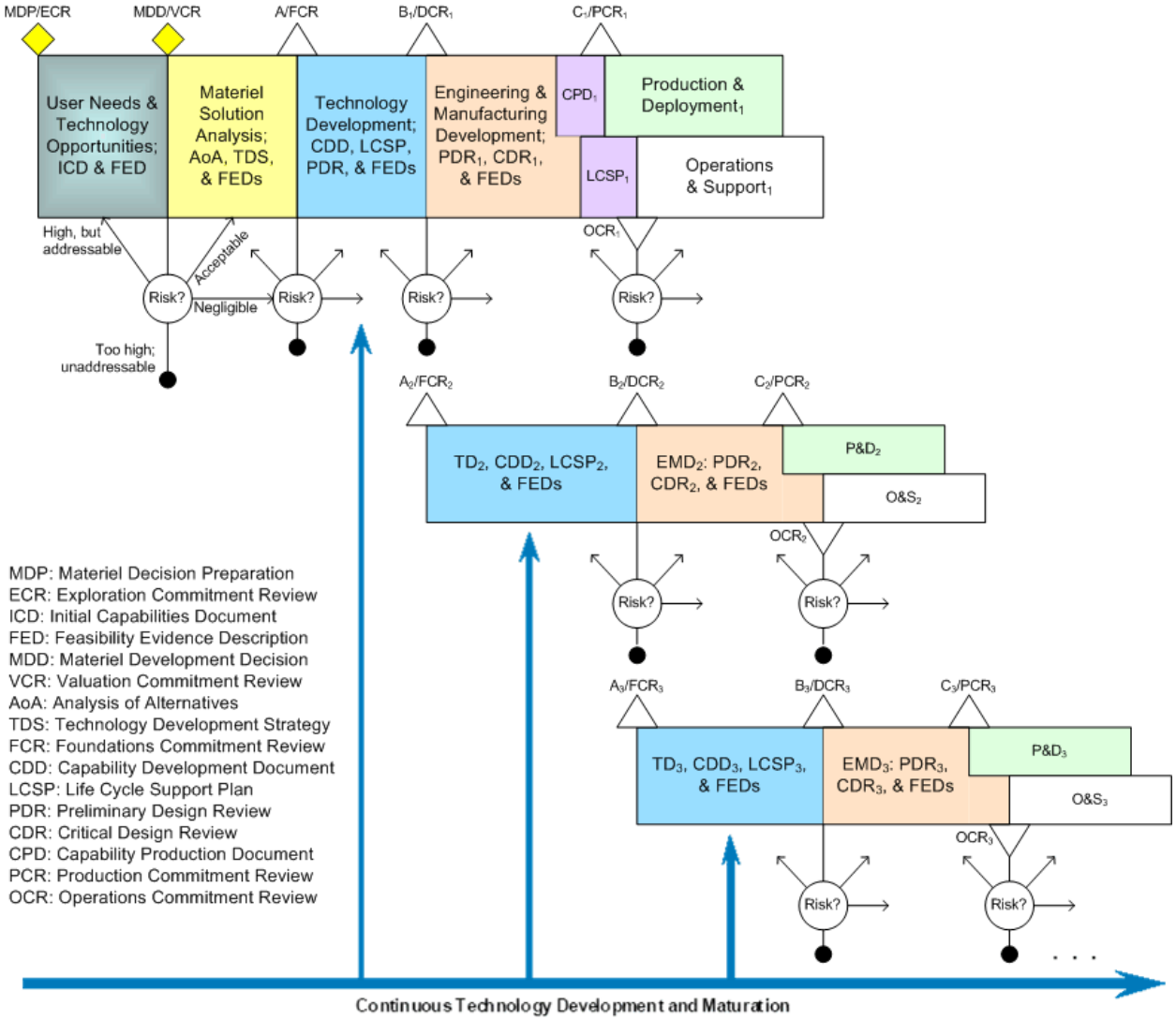
Evolutionary Acquisition per New DoDI 5000.02

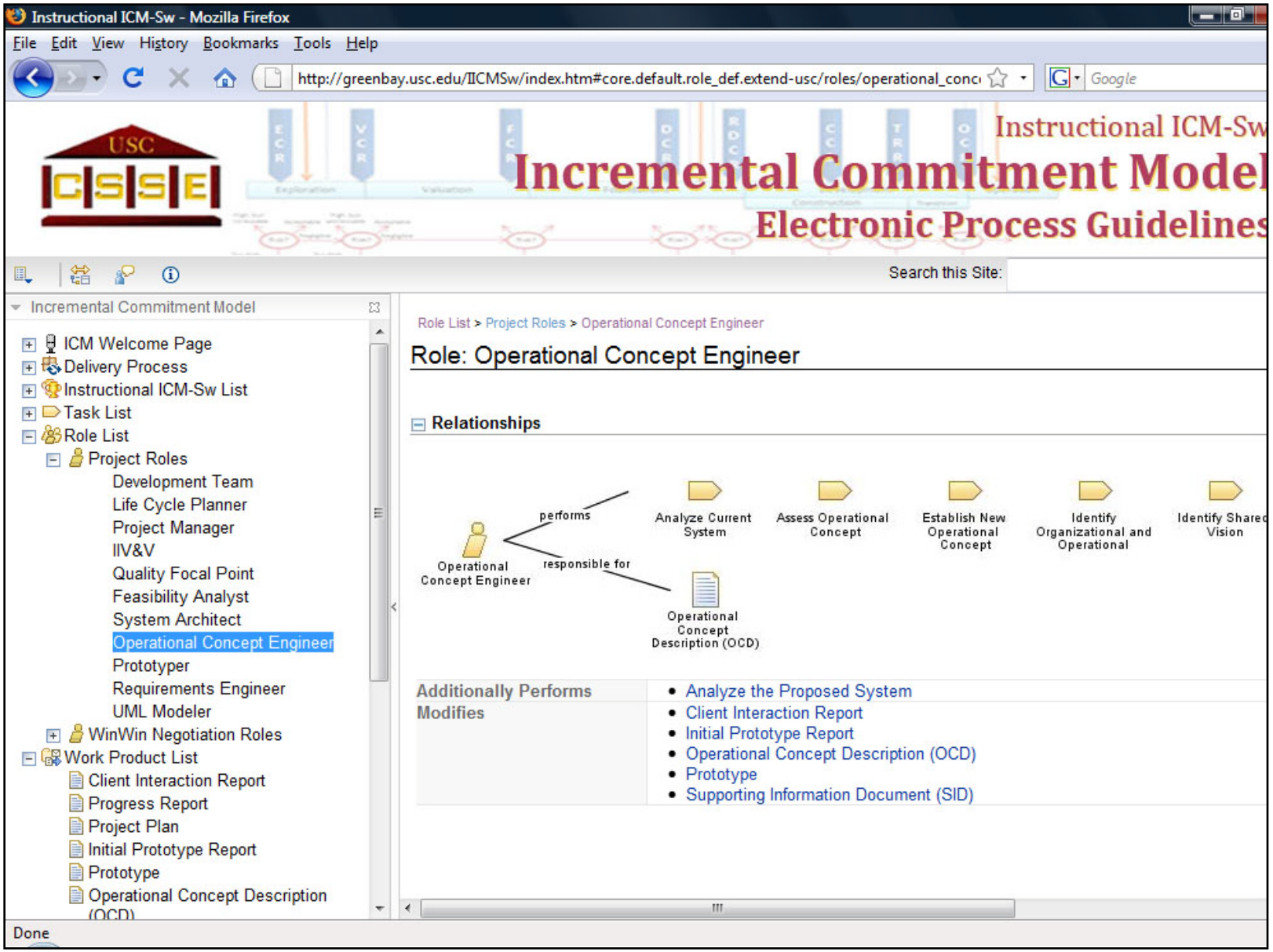
Overlapped Evolutionary

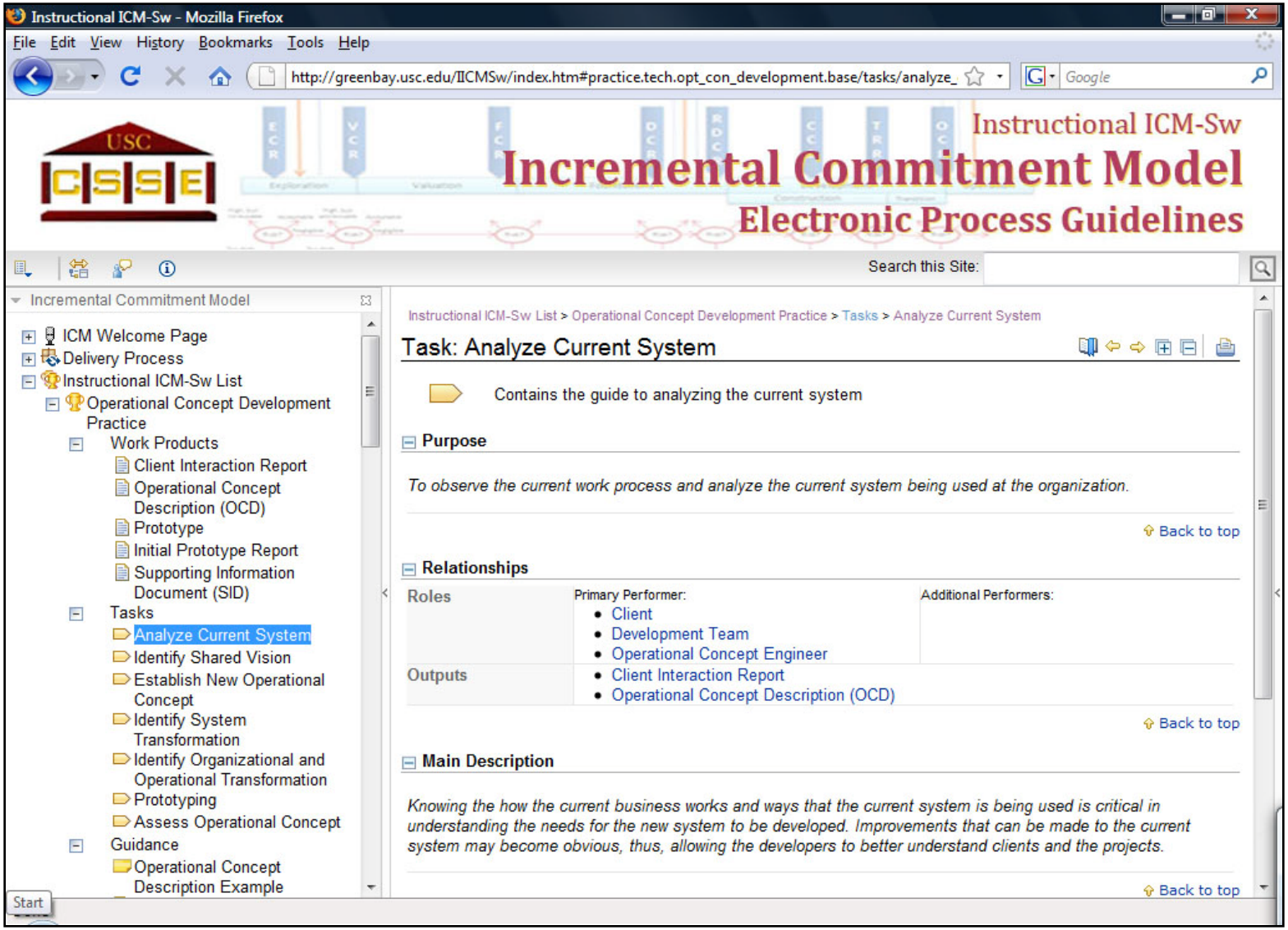


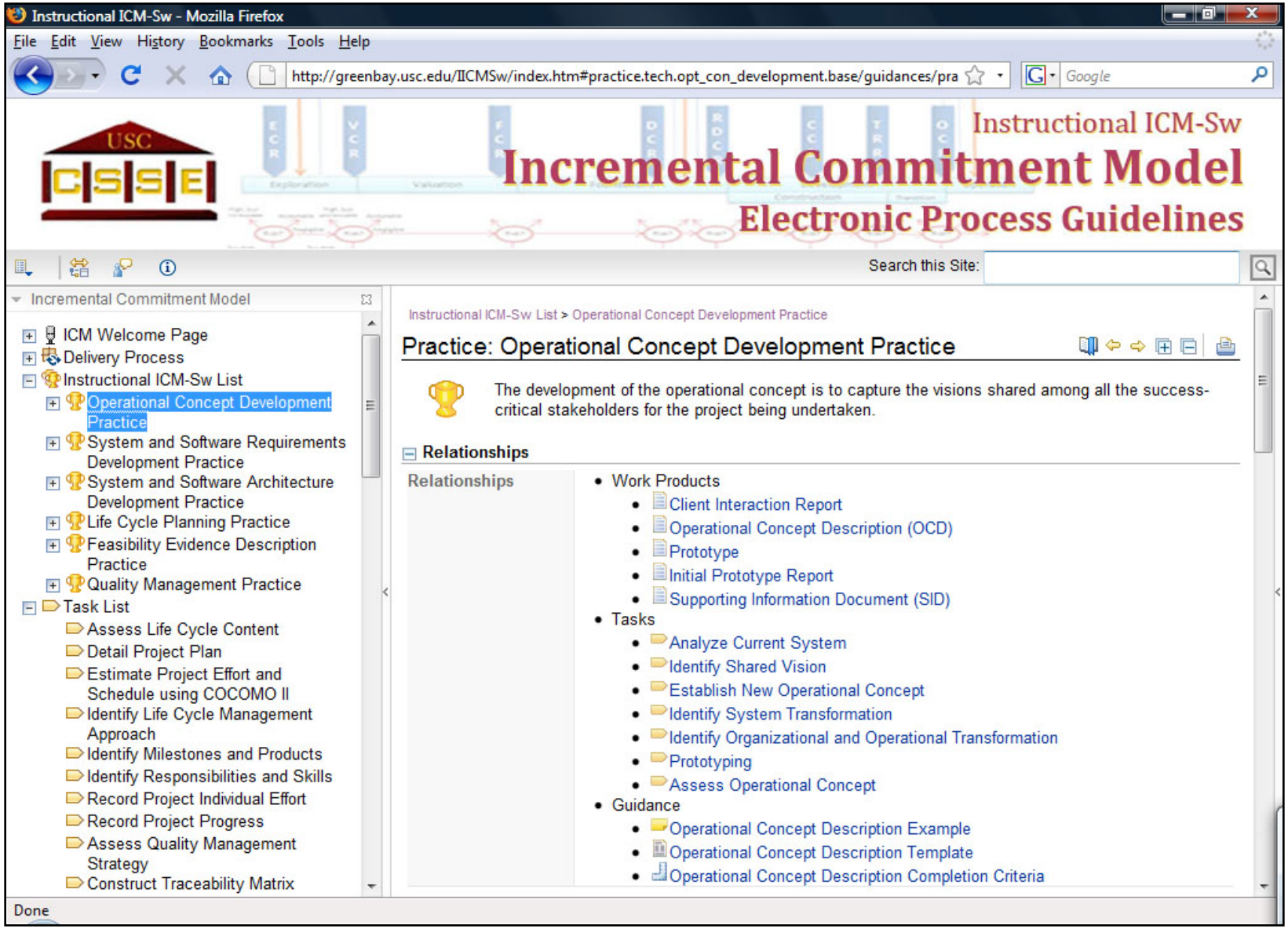
ICM Addresses Both Acquisition and Operations

And concurrent development and next-increment rebaselining




Instructional ICM-Sw - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://greenbay.usc.edu/IICMSw/index.htm#core.default.role_def.extend-usc/roles/operational_conc
Google
Instructional ICM-Sw
Incremental Commitment Model
Electronic Process Guidelines
Search this Site:
Incremental Commitment Model
ICM Welcome Page
Delivery Process
Instructional ICM-Sw List
Task List
Role List
Project Roles
Development Team
Life Cycle Planner
Project Manager
IIV&V
Quality Focal Point
Feasibility Analyst
System Architect
Operational Concept Engineer
Prototyper
Requirements Engineer
UML Modeler
WinWin Negotiation Roles
Work Product List
Client Interaction Report
Progress Report
Project Plan
Initial Prototype Report
Prototype
Operational Concept Description (OCD)
Role List > Project Roles > Operational Concept Engineer
Role: Operational Concept Engineer
Relationships
performs
responsible for
Operational Concept Engineer
Analyze Current System
Assess Operational Concept
Establish New Operational Concept
Identify Organizational and Operational
Operational Concept Description (OCD)
Additionally Performs
Modifies
Analyze the Proposed System
Client Interaction Report
Initial Prototype Report
Operational Concept Description (OCD)
Prototype
Supporting Information Document (SID)
Done

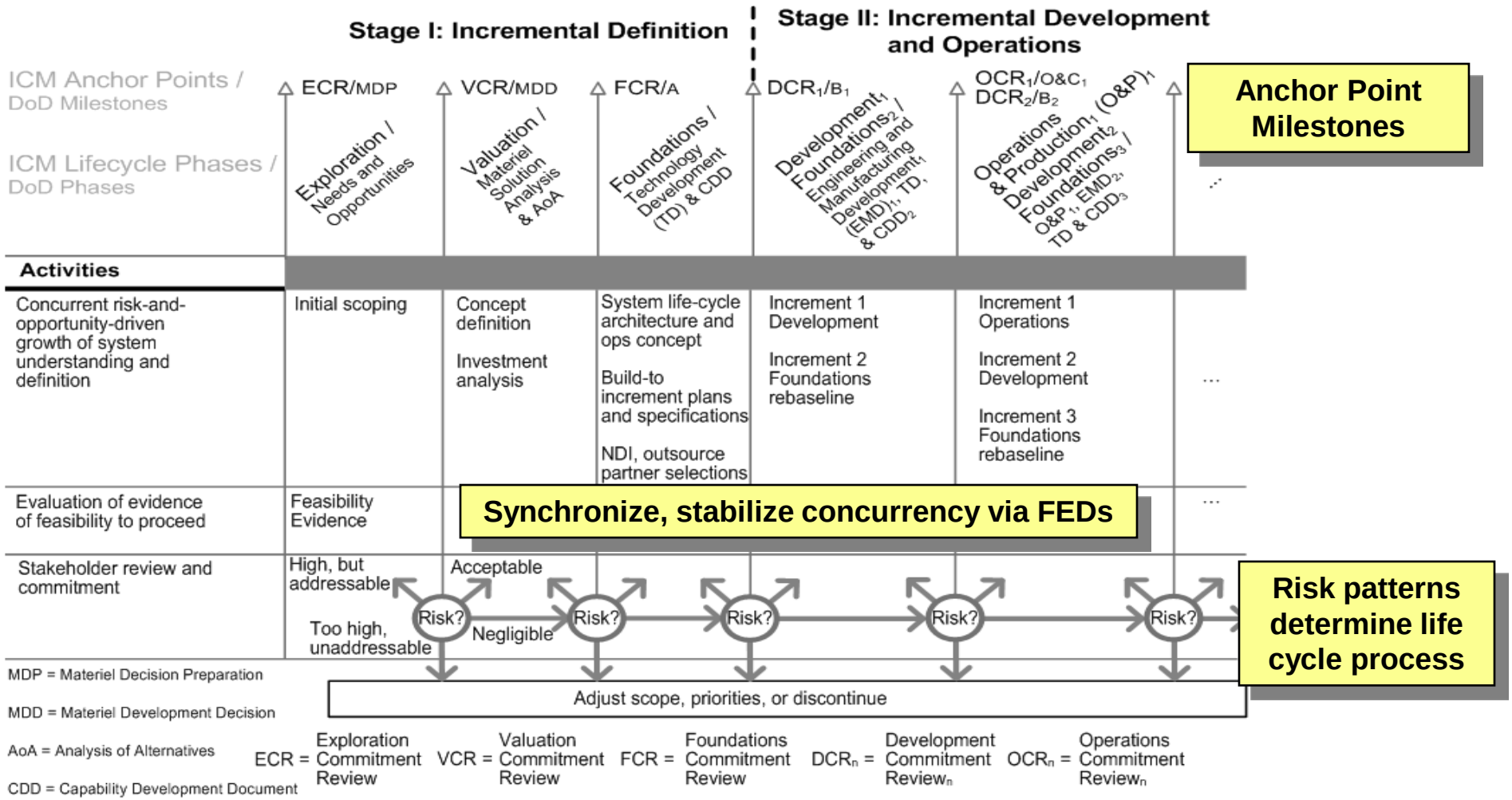

Instructional ICM-Sw - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://greenbay.usc.edu/IICMSw/index.htm#practice.tech.opt_con_development.base/tasks/analyze_
Google
Instructional ICM-Sw
Incremental Commitment Model
Electronic Process Guidelines
Search this Site:
Incremental Commitment Model
ICM Welcome Page
Delivery Process
Instructional ICM-Sw List
Operational Concept Development Practice
Work Products
Client Interaction Report
Operational Concept Description (OCD)
Prototype
Initial Prototype Report
Supporting Information Document (SID)
Tasks
Analyze Current System
Identify Shared Vision
Establish New Operational Concept
Identify System Transformation
Identify Organizational and Operational Transformation
Prototyping
Assess Operational Concept
Guidance
Operational Concept Description Example
Start
Instructional ICM-Sw List > Operational Concept Development Practice > Tasks > Analyze Current System
Task: Analyze Current System
Contains the guide to analyzing the current system
Purpose
To observe the current work process and analyze the current system being used at the organization.
Back to top
Relationships
Roles
Primary Performer:
Client
Development Team
Operational Concept Engineer
Additional Performers:
Outputs
Client Interaction Report
Operational Concept Description (OCD)
Back to top
Main Description
Knowing the how the current business works and ways that the current system is being used is critical in understanding the needs for the new system to be developed. Improvements that can be made to the current system may become obvious, thus, allowing the developers to better understand clients and the projects.
Back to top


Instructional ICM-Sw - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://greenbay.usc.edu/IICMSw/index.htm#practice.tech.opt_con_development.base/guidances/pra
Google
Instructional ICM-Sw
Incremental Commitment Model
Electronic Process Guidelines
Search this Site:
Incremental Commitment Model
ICM Welcome Page
Delivery Process
Instructional ICM-Sw List
Operational Concept Development Practice
System and Software Requirements Development Practice
System and Software Architecture Development Practice
Life Cycle Planning Practice
Feasibility Evidence Description Practice
Quality Management Practice
Task List
Assess Life Cycle Content
Detail Project Plan
Estimate Project Effort and Schedule using COCOMO II
Identify Life Cycle Management Approach
Identify Milestones and Products
Identify Responsibilities and Skills
Record Project Individual Effort
Record Project Progress
Assess Quality Management Strategy
Construct Traceability Matrix
Instructional ICM-Sw List > Operational Concept Development Practice
Practice: Operational Concept Development Practice
The development of the operational concept is to capture the visions shared among all the success-critical stakeholders for the project being undertaken.
Relationships
Relationships
Work Products
Client Interaction Report
Operational Concept Description (OCD)
Prototype
Initial Prototype Report
Supporting Information Document (SID)
Tasks
Analyze Current System
Identify Shared Vision
Establish New Operational Concept
Identify System Transformation
Identify Organizational and Operational Transformation
Prototyping
Assess Operational Concept
Guidance
Operational Concept Description Example
Operational Concept Description Template
Operational Concept Description Completion Criteria
Done

Outline

- Challenges for developing next-generation ground systems
- Overview of ICM
  - How ICM implements the new DoDI 5000.02
- Risk-based balance of agility and assurance
- *ICM process decision table*
- Ground system example for using the ICM
- Conclusions

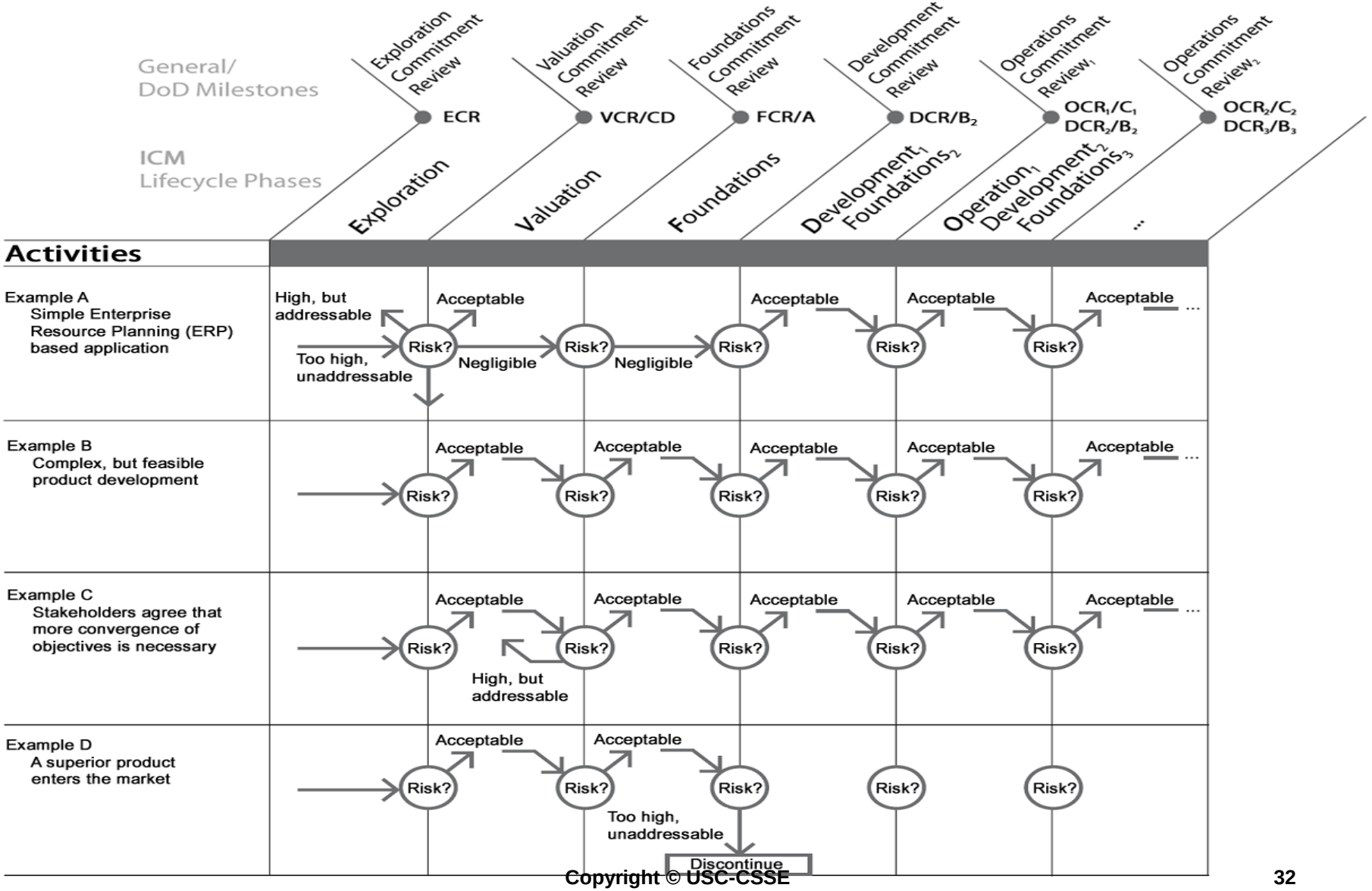
The Incremental Commitment Life Cycle Process: Overview



The ICM as Risk-Driven Process Generator

- Stage I of the ICM has 3 decision nodes with 4 options/node
  - Culminating with incremental development in Stage II
  - Some options involve go-backs
  - Results in many possible process paths
- Can use ICM risk patterns to generate frequently-used processes
  - With confidence that they fit the situation
- Can generally determine this in the Exploration phase
  - Develop as proposed plan with risk-based evidence at VCR milestone
  - Adjustable in later phases

Different Risk Patterns Yield Different Processes



The ICM Process Decision Table: Key Decision Inputs

- Product and project size and complexity
- Requirements volatility
- Mission criticality
- Nature of Non-Developmental Item (NDI)* support
  - Commercial, open-source, reused components
- Organizational and Personnel Capability

*** NDI Definition [DFARS]:** a) any product that is available in the commercial marketplace; b) any previously developed product in use by a U.S. agency (federal, state, or local) or a foreign government that has a mutual defense agreement with the U.S.; c) any product described in the first two points above that requires only modifications to meet requirements; d) any product that is being produced, but not yet in the commercial marketplace, that satisfies the above criteria.

The ICM Process Decision Table: Key Decision Outputs

- Key Stage I activities: incremental definition
- Key Stage II activities: incremental development and operations
- Suggested calendar time per build, per deliverable increment

Common Risk-Driven Special Cases of the ICM (Cases 1-4)

Case 1: Use NDI

Example: Small accounting system
Size, Complexity: Size variable, complexity low
Typical Change Rate/Month: Negligible
Criticality: n/a
NDI Support: Complete
Organizational Personnel Capability: NDI-experienced (medium)
Key Stage I Activities (Incremental Definition): Acquire NDI
Key Stage II Activities (Incremental Development/Operations): Use NDI
Time/Build: n/a
Time/Increment: Vendor-driven

Case 2: Agile

Example: E-services
Size, Complexity: Low
Typical Change Rate/Month: 1-30%
Criticality: Low to medium
NDI Support: Good, in place
Organizational Personnel Capability: Agile-ready, medium-high experience
Key Stage I Activities (Incremental Definition): Skip Valuation and Architecting phases
Key Stage II Activities (Incremental Development/Operations): Scrum plus agile methods of choice
Time/Build: <= 1 day
Time/Increment: 2-6 weeks

Case 3: Architected Agile

Example: Business data processing
Size, Complexity: Medium
Typical Change Rate/Month: 1-10 %
Criticality: Medium to high
NDI Support: Good, most in place
Organizational Personnel Capability: Agile-ready, medium to high experience
Key Stage I Activities (Incremental Definition): Combine Valuation, Architecting phases. Complete NDI preparation.
Key Stage II Activities (Incremental Development/Operations): Architecture-based Scrum of Scrums
Time/Build: 2-4 weeks
Time/Increment: 2-6 months

Case 4: Formal Methods

Example: Security kernel; Safety-critical LSI chip
Size, Complexity: Low
Typical Change Rate/Month: 0.3%
Criticality: Extra high
NDI Support: None
Organizational Personnel Capability: Strong formal methods experience
Key Stage I Activities (Incremental Definition): Precise formal specification
Key Stage II Activities (Incremental Development/Operations): Formally-based programming language; formal verification
Time/Build: 1-5 days
Time/Increment: 1-4 weeks

Common Risk-Driven Special Cases of the ICM (Cases 5-8)

Case 5: Hardware with Embedded Software Component

Example: Multi-sensor control device
Size, Complexity: Low
Typical Change Rate/Month: 0.3 - 1 %
Criticality: Medium to very high
NDI Support: Good, in place
Organizational Personnel Capability: Experienced, medium-high
Key Stage I Activities (Incremental Definition): Concurrent hardware/software engineering. CDR-level ICM DCR
Key Stage II Activities (Incremental Development/Operations): IOC development, LRIP, FRP. Concurrent version N+1 engineering
Time/Build: Software 1-5 days
Time/Increment: Market-driven

Case 6: Indivisible IOC

Example: Complete vehicle platform
Size, Complexity: Medium to high
Typical Change Rate/Month: 0.3 – 1%
Criticality: High to very high
NDI Support: Some in place
Organizational Personnel Capability: Experienced, medium to high
Key Stage I Activities (Incremental Definition): Determine minimum-IOC likely, conservative cost. Add deferrable software features as risk reserve
Key Stage II Activities (Incremental Development/Operations): Drop deferrable features to meet conservative cost. Strong award free for features not dropped.
Time/Build: Software: 2-6 weeks
Time/Increment: Platform: 6-18 months

Case 7: NDI-Intensive

Example: Supply chain management
Size, Complexity: Medium to high
Typical Change Rate/Month: 0.3 – 3%
Criticality: Medium to very high
NDI Support: NDI-driven architecture
Organizational Personnel Capability: NDI-experienced, medium to high
Key Stage I Activities (Incremental Definition): Thorough NDI-suite life cycle cost-benefit analysis, selection, concurrent requirements/architecture definition
Key Stage II Activities (Incremental Development/Operations): Pro-active NDI evolution influencing, NDI upgrade synchronization
Time/Build: Software: 1-4 weeks
Time/Increment: Systems: 6-18 months

Case 8: Hybrid Agile/Plan-Driven System

Example: C4ISR system
Size, Complexity: Medium to very high
Typical Change Rate/Month: Mixed parts; 1-10%
Criticality: Mixed parts; Medium to very high
NDI Support: Mixed parts
Organizational Personnel Capability: Mixed parts
Key Stage I Activities (Incremental Definition): Full ICM, encapsulated agile in high change, low-medium criticality parts (Often HMI, external interfaces)
Key Stage II Activities (Incremental Development/Operations): Full ICM, three-team incremental development, concurrent V&V, next-increment rebaselining
Time/Build: 1-2 months
Time/Increment: 9-18 months

Common Risk-Driven Special Cases of the ICM (Cases 9-11)

Case 9: Multi-Owner Directed System of Systems

Example: Net-centric military operations
Size, Complexity: Very high
Typical Change Rate/Month: Mixed parts; 1-10 %
Criticality: Very high
NDI Support: Many NDIs, some in place
Organizational Personnel Capability: Related experience, medium to high
Key Stage I Activities (Incremental Definition): Full ICM; extensive multi-owner team building, negotiation
Key Stage II Activities (Incremental Development/Operations): Full ICM; large ongoing system/software engineering effort
Time/Build: 2-4 months
Time/Increment: 18-24 months

Case 10: Family of Systems

Example: Medical device product line
Size, Complexity: Medium to very high
Typical Change Rate/Month: 1-3%
Criticality: Medium to very high
NDI Support: Some in place
Organizational Personnel Capability: Related experience, medium to high
Key Stage I Activities (Incremental Definition): Skip Valuation and Architecting phases
Key Stage II Activities (Incremental Development/Operations): Scrum plus agile methods of choice
Time/Build: 1-2 months
Time/Increment: 9-18 months

Case 11: Brownfield

Example: Incremental legacy phaseout
Size, Complexity: High to very high
Typical Change Rate/Month: 0.3-3%
Criticality: Medium-high
NDI Support: NDI as legacy replacement
Organizational Personnel Capability: Legacy re-engineering
Key Stage I Activities (Incremental Definition): Re-engineer/refactor legacy into services
Key Stage II Activities (Incremental Development/Operations): Incremental legacy phaseout
Time/Build: 2-6 weeks/refactor
Time/Increment: 2-6 months

Common Risk-Driven Special Cases of the ICM (Cases 12a/b)

Case 12a: Net-Centric Services – Community Support	Case 12b: Net-Centric Services – Quick Response Decision Support
Example: Community services or special interest group	**Example:** Response to competitor initiative
Size, Complexity: Low to medium	**Size, Complexity:** Medium to high
Typical Change Rate/Month: 0.3-3%	**Typical Change Rate/Month:** 3-30%
Criticality: Low to medium	**Criticality:** Medium to high
NDI Support: Tailorable service elements	**NDI Support:** Tailorable service elements
Organizational Personnel Capability: NDI-experienced	**Organizational Personnel Capability:** NDI-experienced
Key Stage I Activities (Incremental Definition): Filter, select, compose, tailor NDI	**Key Stage I Activities (Incremental Definition):** Filter, select, compose, tailor NDI
Key Stage II Activities (Incremental Development/Operations): Evolve tailoring to meet community needs	**Key Stage II Activities (Incremental Development/Operations):** Satisfy quick response; evolve or phase out
Time/Build: <= 1 day	**Time/Build:** <= 1 day
Time/Increment: 2-12 months	**Time/Increment:** Quick response-driven

LEGEND

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.
LSI: Large Scale Integration.
LRIP: Low-Rate Initial Production.
NDI: Non-Development Item.
SW: Software

Outline

- Challenges for developing next-generation ground systems
- Overview of ICM
  - How ICM implements the new DoDI 5000.02
- Risk-based balance of agility and assurance
- ICM process decision table
- *Ground system example for using the ICM*
- Conclusions

Ground System COTS: Is This A Risk?

- We just started integrating the software
  - and we found out that COTS* products A and B just can't talk to each other
- We've got too much tied into A and B to change
- Our best solution is to build wrappers around A and B to get them to talk via CORBA**
- This will take 3 months and $300K
- It will also delay integration and delivery by at least 3 months

*COTS: Commercial off-the-shelf

**CORBA: Common Object Request Broker Architecture

Ground System COTS:Is This A Risk?

- We just started integrating the software
  - and we found out that COTS* products A and B just can't talk to each other
- We've got too much tied into A and B to change

****

- No, it is a problem
  - Being dealt with reactively
- Risks involve uncertainties
  - And can be dealt with pro-actively
  - Earlier, this problem was a risk

ICM FCR Milestone: Expert Evidence Review

- The Java telemetry COTS package A and the dotNet Health Monitoring COTS package B perform best
  - But it is likely that they will have interoperability problems
  - Probability of loss P(L)
- If we commit to using A and B
  - And we find out in integration that they can't talk to each other
  - We'll add more cost and delay delivery by at least 3 months
  - Size of loss S(L)
- We have a risk exposure of

  RE = P(L) * S(L)

Options for Responding to Risk Finding

- Buying information
- Risk avoidance
- Risk transfer
- Risk reduction
- Risk acceptance

Developer Risk Management Plan: Begin by Buying Information

- We'll spend $30K and 2 weeks prototyping the integration of A and B
- This will buy information on the magnitude of P(L) and S(L)
- If RE = P(L) * S(L) is small, we'll accept and monitor the risk
- If RE is large, we'll use the information to choose the best of the other strategies

Other Risk Management Strategies

- Risk Avoidance
  - The Java-based Health Monitoring COTS product C performs 80% as well as B, and it can interoperate with A
  - Delivering on time may be worth more to the customer than the small performance loss
- Risk Transfer
  - If the customer values the extra performance obtained by using A and B, have them establish a risk reserve.
  - To be used to the extent that A and B can't talk to each other
- Risk Reduction
  - If we build the wrappers and the CORBA connections right now, we add cost but minimize the schedule delay
- Risk Acceptance
  - If we can solve the A and B interoperability problem, we'll have a big competitive edge on the future procurements
  - Let's do this on our own money, and patent the solution
- Customer agrees to enter Foundations phase based on plan

Outline

- Challenges for developing next-generation ground systems
- Overview of ICM
  - How ICM implements the new DoDI 5000.02
- Risk-based balance of agility and assurance
- ICM process decision table
- Ground system example for using the ICM
- *Conclusions*

ICM Summary

- **Current processes not well matched to future challenges**
  - **Emergent, rapidly changing requirements**
  - **High assurance of scalable performance and qualities**
- **Incremental Commitment Model addresses challenges**
  - **Assurance via evidence-based milestone commitment reviews, stabilized incremental builds with concurrent V&V**
    - **Evidence shortfalls treated as risks**
  - **Adaptability via concurrent agile team handling change traffic and providing evidence-based rebaselining of next-increment specifications and plans**
  - **Use of critical success factor principles: stakeholder satisficing, incremental growth, concurrent engineering, iterative development, risk-based activities and milestones**
  - **Can be adopted incrementally**
- **Major implications for funding, contracting, career paths**

Implications for funding, contracting, career paths

- Incremental vs. total funding
  - Often with evidence-based competitive downselect
- No one-size-fits all contracting
  - Separate instruments for build-to-spec, agile rebaselining, V&V teams
    - With funding and award fees for collaboration, risk management
    - Compatible regulations, specifications, and standards
    - Compatible acquisition corps education and training
  - Generally, schedule/cost/quality as independent variable
    - Prioritized feature set as dependent variable
- Multiple career paths
  - For people good at build-to-spec, agile rebaselining, V&V
  - For people good at all three
    - Future program managers and chief engineers

ICM Transition Paths

- Existing programs may benefit from some ICM principles and practices, but not others
- Problem programs may find some ICM practices helpful in recovering viability
- Primary opportunities for incremental adoption of ICM principles and practices
  - Supplementing traditional requirements and design reviews with development and review of feasibility evidence
  - Stabilized incremental development and concurrent architecture rebaselining
  - Using schedule as independent variable and prioritizing features to be delivered
  - Continuous verification and validation
  - Using the process decision table
- See http://csse.usc.edu (tech report 2009-500)

References - I

- Beck, K., Extreme Programming Explained, Addison Wesley, 1999.
- Boehm, B., "Some Future Trends and Implications for Systems and Software Engineering Processes", Systems Engineering 9(1), pp. 1-19, 2006.
- Boehm, B., Brown, W., Basili, V., and Turner, R., "Spiral Acquisition of Software-Intensive Systems of Systems, CrossTalk, Vol. 17, No. 5, pp. 4-9, 2004.
- 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., and Lane, J., "A Process Decision Table for Integrated Systems and Software Engineering," Proceedings, CSER 2008, April 2008.
- Boehm, B., "Future Challenges and Rewards for Software Engineers," DoD Software Tech News, October 2007, pp. 6-12.
- Boehm, B. et al., Software Cost Estimation with COCOMO II, Prentice Hall, 2000.
- Boehm, B., Software Engineering Economics, Prentice Hall, 2000.
- Carlock, P. and Fenton, R., "System of Systems (SoS) Enterprise Systems for Information-Intensive Organizations," Systems Engineering, Vol. 4, No. 4, pp. 242-26, 2001.
- Carlock, P., and J. Lane, "System of Systems Enterprise Systems Engineering, the Enterprise Architecture Management Framework, and System of Systems Cost Estimation", 21st International Forum on COCOMO and Systems/Software Cost Modeling, 2006.
- Checkland, P., Systems Thinking, Systems Practice, Wiley, 1980 (2nd ed., 1999).
- 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.
- Electronic Industries Alliance (1999); EIA Standard 632: Processes for Engineering a System
- Galorath, D., and Evans, M., Software Sizing, Estimation, and Risk Management, Auerbach, 2006.
- Hall, E.T., Beyond Culture, Anchor Books/Doubleday, 1976.

References -II

- Highsmith, J., Adaptive Software Development, Dorset House, 2000.
- International Standards Organization, Information Technology Software Life Cycle Processes, ISO/IEC 12207, 1995
- ISO, Systems Engineering – System Life Cycle Processes, ISO/IEC 15288, 2002.
- Jensen, R. "An Improved Macrolevel Software Development Resource Estimation Model," Proceedings, ISPA 5, April 1983, pp. 88-92.
- Krygiel, A., Behind the Wizard's Curtain; CCRP Publication Series, July, 1999, p. 33
- Lane, J. and Boehm, B., "System of Systems Cost Estimation: Analysis of Lead System Integrator Engineering Activities", Information Resources Management Journal, Vol. 20, No. 2, pp. 23-32, 2007.
- Lane, J. and Valerdi, R., "Synthesizing SoS Concepts for Use in Cost Estimation", Proceedings of IEEE Systems, Man, and Cybernetics Conference, 2005.
- Lientz, B., and Swanson, E.B., Software Maintenance Management, Addison Wesley, 1980.
- Madachy, R., Boehm, B., Lane, J., "Assessing Hybrid Incremental Processes for SISOS Development", USC CSSE Technical Report USC-CSSE-2006-623, 2006.
- Maier, M., "Architecting Principles for Systems-of-Systems"; Systems Engineering, Vol. 1, No. 4 (pp 267-284).
- Maier, M., "System and Software Architecture Reconciliation," Systems Engineering 9 (2), 2006, pp. 146-159.
- Northrop, L., et al., Ultra-Large-Scale Systems: The Software Challenge of the Future, Software Engineering Institute, 2006.
- Pew, R. W., and Mavor, A. S., Human-System Integration in the System Development Process: A New Look, National Academy Press, 2007.
- Putnam, L., "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Trans SW Engr., July 1978, pp. 345-361.
- Rechtin, E. Systems Architecting, Prentice Hall, 1991.
- Schroeder, T., "Integrating Systems and Software Engineering: Observations in Practice," OSD/USC Integrating Systems and Software Engineering Workshop, http://csse.usc.edu/events/2007/CIIForum/pages/program.html, October 2007.