# Using the Incremental Commitment Model to Achieve Successful System Development

Barry Boehm and Jo Ann Lane
University of Southern California
Center for Systems and Software Engineering

*Studies to evaluate the usage and success of the spiral development model have shown mixed results—many successes but many misinterpretations and neglect of its underlying principles, leading to continuing development problems and failed systems. This article describes a recent improvement to the spiral development process model that is easier to understand, is harder to misinterpret, and better enables integration of the human, hardware, and software aspects of a software-intensive system's development and evolution. This model, called the Incremental Commitment Model (ICM), embodies the principles underlying the spiral model, but organizes its process into multiple views (including a spiral view) that are more straightforward to apply and better aligned with mainstream system acquisition phases and milestones.*

## Introduction

The ultimate goal of system development is to deliver a system that satisfies the needs of its operational stakeholders—users, operators, administrators, maintainers, interoperators, the general public—within satisfactory levels of the resources of its development stakeholders—funders, acquirers, developers, suppliers, others. From the human-system integration perspective, satisfying operational stakeholders' needs can be broadly construed to mean a system that is usable and dependable; permits few or no human errors; and leads to high productivity and adaptability (Pew and Mavor, 2007). Developing and delivering systems that simultaneously satisfy all of these success-critical stakeholders usually requires managing a complex set of risks such as usage uncertainties, schedule uncertainties, supply issues, requirements changes, and uncertainties associated with technology maturity and technical design. Each of these areas poses a risk to the delivery of an acceptable operational system within the available budget and schedule. End-state operational system risks can be categorized as uncertainties in achieving a system mission, carrying out the work processes, operating within various constraints such as cost or personnel, satisfying operational stakeholders, or achieving an acceptable operational return on investment.

This article presents a re-engineered version of the spiral model called the Incremental Commitment Model (ICM) It is based onfive critical success factor principles emerging from an analysis of the factorsdistinguishing unsuccessful from successful "spiral development" projects:

1. *Stakeholder satisficing*. If a system development process presents a success-critical operational or development stakeholder with the prospect of an unsatisfactory outcome, the stakeholder will generally refuse to cooperate, resulting in an unsuccessful system. Stakeholder satisficing involves identifying

the success-critical stakeholders and their value propositions; negotiating a mutually satisfactory set of system requirements, solutions, and plans; and managing proposed changes to preserve a mutually satisfactory outcome.

2. *Incremental growth of system definition and stakeholder commitment*. This characteristic encompasses the necessity of incremental discovery of emergent human-system requirements and solutions via such discovery methods as prototyping, operational exercises, and use of early system capabilities. Requirements and commitment cannot be monolithic or fully pre-specifiable for complex, human-intensive systems; understanding, trust, definition and commitment is achieved through an evolutionary process.
3. *Iterative system development and definition*. The incremental and evolutionary approaches lead to cyclic refinements of requirements, solutions, and development plans. Such iteration helps projects to learn early and efficiently about operational and performance requirements.
4. *Concurrent system definition and development*. Initially, this includes concurrent engineering of requirements and solutions, and integrated product and process definition. In later increments, change-driven rework and rebaselining of next-increment requirements, solutions and plans occurs concurrently with development of the current system increment. This allows early fielding of core capabilities, continual adaptation to change, and timely growth of complex systems without waiting for every requirement and subsystem to be defined.
5. *Risk management – risk driven activity levels and anchor point milestones*. The level of detail of specific products and processes will depend on the level of risk associated with them. If the user interface is considered a high-risk area, for example, then more design activity will be devoted to this component to achieve stakeholder commitments at particular design anchor points. On the other hand, if interactive graphic user interface (GUI) builder capabilities make it low-risk not to document evolving GUI requirements, much time-consuming effort can be saved by not creating and continually updating GUI requirements documents while evolving the GUI to meet user needs.

## The Evolving Nature of System Requirements

Traditionally, requirements have served as the basis for competitive selection of system suppliers and subsequent contracts between the acquirer and selected supplier. As such, they are expected to be prespecificably complete, consistent, unambiguous, and testable. Frequently, progress payments and award fees are based on the degree to which these properties are satisfied.

However, particularly as systems depend more and more on being parts of network-centric, collaboration-intensive systems of systems, the traditional approach to system requirements has encountered increasing difficulties. The key ICM principles above have been synthesized to mitigate many of these difficulties which include:

- *Emergent requirements*. The most appropriate user interfaces and collaboration modes for a complex human-intensive system are not specifiable in advance, but emerge with system prototyping and usage. Forcing them to be prematurely and precisely specified generally leads to poor business or mission performance and expensive late rework and delays (Highsmith, 2000).

- *Rapid change*. Specifying current-point-in-time snapshot requirements on a cost-competitive contract generally leads to a big design up front, and a point-solution architecture that is hard to adapt to new developments. Each of the many subsequent changes then leads to considerable nonproductive work in redeveloping documents and software, and in renegotiating contracts (Beck 1999).
- *Reusable components.* Prematurely specifying requirements (e.g., hasty specification of a 1-second response time requirement when later prototyping showed that 4 seconds would be acceptable) that disqualify otherwise most cost-effective reusable components often leads to overly expensive, late, and unsatisfactory systems (Boehm 2000).

The key principles above address such requirements issues via incremental and evolutionary acquisition of the most important and best-understood capabilities first; concurrently engineering requirements and solutions; using prototypes, models, and simulations as ways of obtaining information to reduce the risk of specifying inappropriate requirements; and basing requirements on stakeholder negotiations once their implications are better understood.

The use of these principles works best when the stakeholders adopt a different vocabulary when dealing with requirements. The primary (Webster, 2006) definition of a requirement is, "something required, i.e., claimed or asked for by right and authority." It is much easier to make progress toward a mutually satisfactory negotiated solution if the stakeholders use more negotiation-oriented terms such as "goals", "objectives", or "value propositions" rather than assuming that they are dealing with non-negotiable "requirements". And when tradeoffs among cost, schedule, performance, and capabilities are not well understood, it is better to specify prioritized capabilities and ranges of mutually satisfactory performance, rather than to insist on precise and unambiguous requirements. However, following Principle 5 above on risk-driven level of product detail, it is important to converge on precise requirements where the risk of having them be imprecise is high. Some good examples are human-computer interaction protocols for safety-critical systems and interfaces among separately-developed mission-critical subsystems.

## Principles-Based Comparison of Alternative Process Models

The human-system integration study described in (Pew and Mavor, 2007) included an analysis of candidate systems development process models with respect to the five critical success factor principles. The candidate models included the waterfall, V, spiral, and concurrent engineering process models discussed in the first two chapters of the Handbook of Systems Engineering and Management (Sage and Rouse, 1999; Patterson, 1999), plus emerging candidates such as agile methods (Beck, 1999; Highsmith, 2000), V-model updates (V-Modell XT, 2005), and 2001 extensions of the spiral model (Boehm and Hansen, 2001)

The analysis indicates that all of the models made useful contributions, but exhibited shortfalls with respect to human factor considerations, particularly in explicit guidance for stakeholder satisficing. Pure-sequential implementations of the Waterfall and V-models are not good matches for human-intensive systems. They are becoming less frequent, but are still often encountered due to imposition of legacy contracting clauses and standards. More recently, the V-Model XT has adopted more risk-driven and incremental approaches that encourage more concurrent engineering (V-Modell XT, 2005), but it takes some skill to build in stakeholder satisficing and to avoid overly heavyweight implementations and difficulties in coping with rapid change. Risk-driven evolutionary development is better at coping with rapid change, but can have difficulties in optimizing around early increments with architectures that encounter later scalability problems. Concurrent engineering explicitly addresses incremental growth, concurrency, and iteration. It is compatible with stakeholder satisficing and risk management, but lacks much explicit guidance in addressing them.

Agile methods are even better at coping with rapid change, but can have even more difficulties with scalability and with mission-critical or safety-critical systems, in which fixing shortfalls in the next increment is not acceptable. There are a wide variety of agile methods; some such as Lean and Feature-Driven Development are better at scalability and criticality than others. The version of spiral development in (Boehm and Hansen, 2001) with stakeholder satisficing and anchor point milestones covers all of the principles, but is unspecific about just how risk considerations guide iteration and incremental growth.

The analysis indicted primary shortfalls in support of human factors integration and unproven ability to scale up to the future process challenges involving emergent, network-centric, massively-collaborative systems of systems (Maier, 1998; Sage and Cuppan, 2001). The study undertook to integrate human factors considerations into the Spiral 2005 process model (Boehm and Lane, 2006), a generalization of the WinWin Spiral Model being used in the Future Combat Systems system of systems (Boehm et al., 2004). The result is the Incremental Commitment Model to be discussed next.

## Overview of the ICM

The ICM builds on early verification and validation concepts in the V-model, concurrency concepts in the Concurrent Engineering model, lighter-weight concepts in the Agile and Lean models, risk-driven concepts in the spiral model, the phases and anchor points in the Rational Unified Process (RUP) (Royce, 1998; Kruchten, 1999; Boehm, 1996), and recent extensions of the spiral model to address systems of systems acquisition (Boehm and Lane, 2006). In comparison to the software-intensive RUP, the ICM also addresses hardware and human factors integration. It extends the RUP phases to cover the full system life cycle: an Exploration phase precedes the RUP Inception phase, which is refocused on valuation and investment analysis. The RUP Elaboration phase is refocused on Architecting (a term based on (Rechtin, 1991) describing concurrent development of requirements, architecture, plans, and feasibility evidence); the RUP Construction and Transition phases are combined into Development; and an additional Operations phase combines operations, production, maintenance, and phase-out.

In comparison to the sequential waterfall (Royce 1970) and V-model (Patterson, 1999), the ICM explicitly emphasizes concurrent engineering of requirements and solutions, establishes explicit Feasibility Rationales as pass/fail milestone criteria; explicitly enables risk-driven avoidance of unnecessary documents, phases, and reviews; and provides explicit support for a stabilized current-increment development concurrently with a separate change processing and rebaselining activity to prepare for appropriate and stabilized development of the next increment. These aspects can be integrated into a waterfall or V-model, enabling projects required to use such models to cope more effectively with systems of the future.

An overview of the ICM life cycle process is shown in Figure 1. It identifies the concurrently engineered life cycle phases, the stakeholder commitment review points and their use of feasibility rationales to assess the compatibility, feasibility and risk associated with the concurrently-engineering artifacts; and the major focus of each life cycle phase. There are a number of alternatives at each commitment point. These are: (1) the risks are negligible and no further analysis and evaluation activities are needed to complete the next phase; (2) the risk is acceptable and work can proceed to the next life cycle phase: (3) the risk is addressable but requires backtracking; or (4) the risk is too great and the development process should be rescoped or halted. These risks are assessed by the system's success-critical stakeholders, whose commitment will be based on whether the current level of system definition gives sufficient evidence that the system will satisfy their value propositions (see (Boehm and Jain, 2006) for additional information on value-based systems engineering).

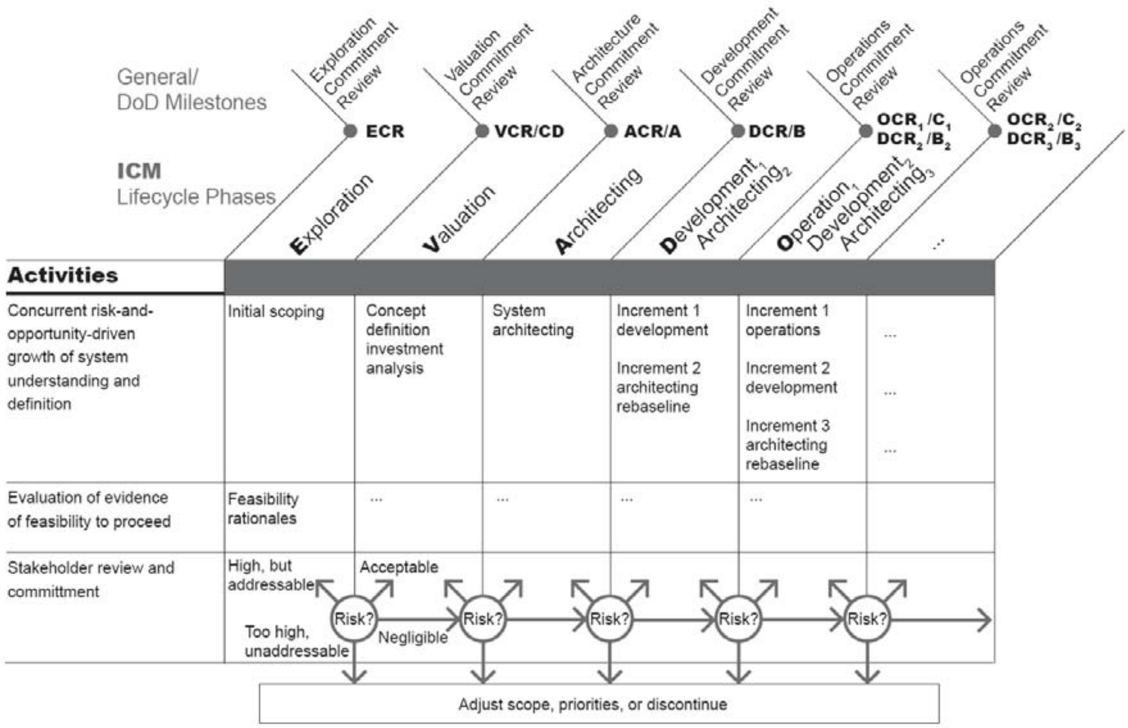


**Figure 1. Overview of the Incremental Commitment Life Cycle Process.**

The ICM commitment milestones correspond fairly closely with the Department of Defense acquisition milestones as defined in DoDI 5000.2 (U.S. DoD, 2003). For example, the ICM DCR-milestone commitment to proceed into Development based on the validated Life Cycle Architecture package (an Operations Concept Description, Requirements Description, Architecture Description, Life Cycle Plan, working prototypes or high-risk elements, and a Feasibility Rationale providing evidence of their compatibility and feasibility) corresponds fairly closely with DoD's Milestone B commitment to proceed into the System Development and Demonstration phase.

A simple metaphor to help understand the ICM is to compare ICM to poker games such as Texas Hold'em, as compared with single-commitment gambling games such as Roulette. At each round of betting, each stakeholder looks at their own hole cards and the jointly-visible community cards, and decides whether it is worth adding further resources to the pot of resources on the table, in order to see further community cards and to win the pot based on having the best poker hand constructible from one's own hole cards and the community cards. With the ICM, however, there will be negotiations designed to make win conditions for each success-critical stakeholder.

## Using the ICM to Define and Understand Complexities of Development Process

The ICM can be used in many ways to define and analyze various aspects of the development process.  Some of the key views include:

- *Process model generator*:  As with the spiral model, the ICM is a risk-driven process model generator—different risks create different processes.  For example, a project to implement a simple business application based on a stable, well-understood Enterprise Resource Planning package could follow the Negligible Risk arrows to go directly into the Development phase using an agile method such as Scrum or Extreme Programming.  On the other hand, if a project's goal is to upgrade and integrate several somewhat incompatible legacy applications into a service-oriented web-based system, considerable exploration, prototyping, and analysis might be required to ensure a viable approach for the integration of the application databases and user interfaces before beginning the concurrent upgrade of the various legacy applications.
- *Concurrent level of activity*:  This view, shown in Figure 2, is an extension of a similar view of concurrently engineered software projects developed as part of the Rational Unified Process (Kruchten, 1999).  As with the RUP version, it should be emphasized that the magnitude and shape of the levels of effort will be risk-driven and likely to vary from project to project.  In particular, they are likely to have mini risk/opportunity-driven peaks and valleys, rather than the smooth curves shown for simplicity in Figure 2. The main intent of this view is to emphasize the necessary concurrency of the primary success-critical activities shown as rows in Figure 2. Thus, in interpreting the Exploration column,  although system scoping is the primary objective of the Exploration phase, doing it well involves a considerable amount of activity in understanding needs, envisioning opportunities, identifying and reconciling stakeholder goals and objectives, architecting solutions, life cycle planning, evaluation of alternatives, and negotiation of stakeholder commitments.

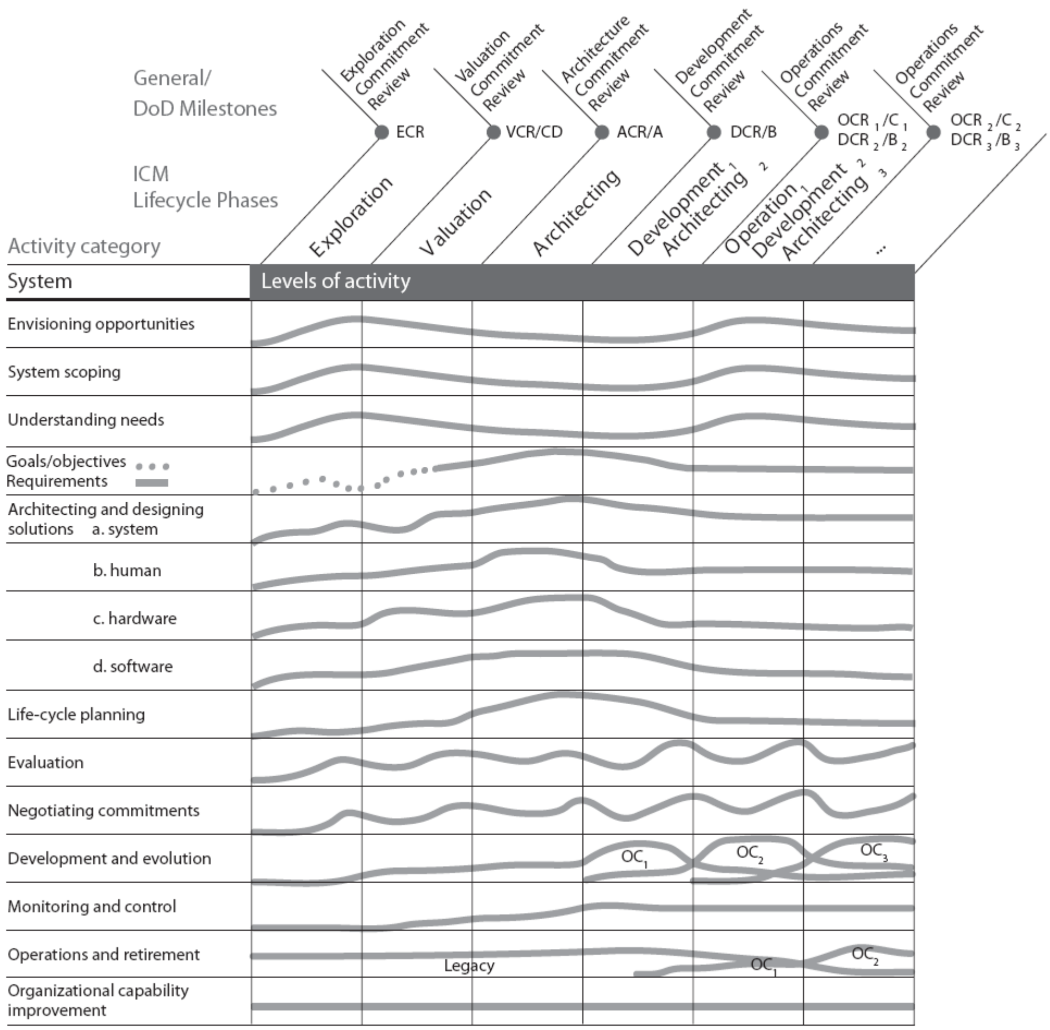


**Figure 2. ICM Activity Categories and Level of Effort.**

- *Anchor point milestones*: Figure 2 indicates that a great deal of concurrent activity occurs within and across the various ICM phases. To make this concurrency work, the next view, the anchor point milestones view, addresses how the many concurrent activities are synchronized, stabilized, and risk-assessed at the end of each phase. Each of these anchor point reviews, labeled at the top of Figure 1, is focused on developer-produced *evidence,* instead of PowerPoint charts, to help the key stakeholders determine the next level of commitment. For the Exploration Commitment Review (ECR), the focus is on a review of an Exploration Phase plan with the proposed scope, schedule, deliverables, and required resource commitment, by a key subset of stakeholders. The plan content is risk-driven, and could be put on a single page for a small and non-controversial Exploration phase. For the Valuation Commitment Review (VCR), the risk-driven focus is similar; the content includes the Exploration phase results and a valuation phase plan; and a review by all of the stakeholders involved in the Valuation phase. The Architecture Commitment Review (ACR) and the Development Commitment Review (DCR) reviews are based on the highly successful AT&T Architecture Review Board procedures described in (Marenzano et al., 2005). For the ACR, only high-risk aspects of the

Operational Concept, Requirements, Architecture, and Plans are elaborated in detail. And it is sufficient to provide evidence that at least one combination of those artifacts satisfies the Feasibility Rationale criteria (similar to the RUP Life Cycle Objectives (LCO) milestone), as compared to demonstrating this at the DCR for a particular choice of artifacts to be used for development. The Operations Commitment Review (OCR) is different, in that it addresses the often much higher operational risks of fielding an inadequate system. In general, stakeholders will experience a factor of 2-to-10 increase in commitment level in going through the sequence of ECR to DCR milestones, but the increase in going from DCR to OCR can be much higher. The OCR focuses on evidence of the adequacy of plans and preparations with respect to doctrine, organization, training, material, leadership, personnel, and facilities (DOTMLPF), along with plans, budgets, and schedules for production, fielding, and operations.

Taking this to the next level for system of systems development, Figure 3 shows how these anchor point milestone reviews can be used to synchronize, stabilize, and manage risks across multiple supplier/vendor/strategic partner activities. The elaboration of the concurrent engineering and feasibility evaluation activities makes it clearer what needs to be concurrently engineered and evaluated in each phase. For additional information on the ICM anchor point reviews, see Chapter 2 of (Pew and Mavor, 2007).

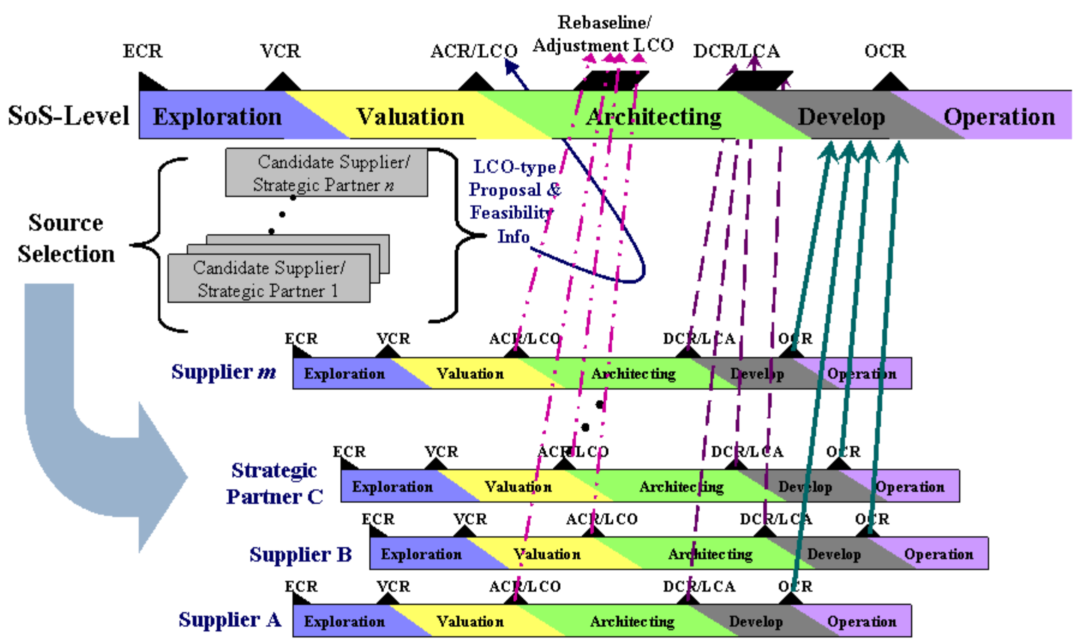


**Figure 3. Combining SoS Engineering and Component Supplier Processes using ICM Anchor Point Reviews [Boehm and Lane, 2007].**

- *Spiral process view:* A simplified spiral model view of the ICM is provided in Figure 5. It avoids sources of misinterpretation in previous versions of the spiral model, and concentrates on the five key spiral development principles. Stakeholder satisficing is necessary to pass the stakeholder commitment review points or anchor point milestones. Incremental growth in system understanding, cost, time, product, and process detail is shown by the spiral growth along the radial dimension. Concurrent engineering is shown by progress along the angular dimension. Iteration is shown by taking several spiral cycles both to define and develop the system. Risk management is captured by indicating that the activities' and products' levels of detail in the angular dimension are risk-driven, and by the risk-driven arrows pointing out from each of the anchor point commitment milestones. These arrows show that the spiral model is not a sequential, unrollable process, but that it incorporates many paths through the diagram including skipping a phase or backtracking to an early phase based on assessed risk.

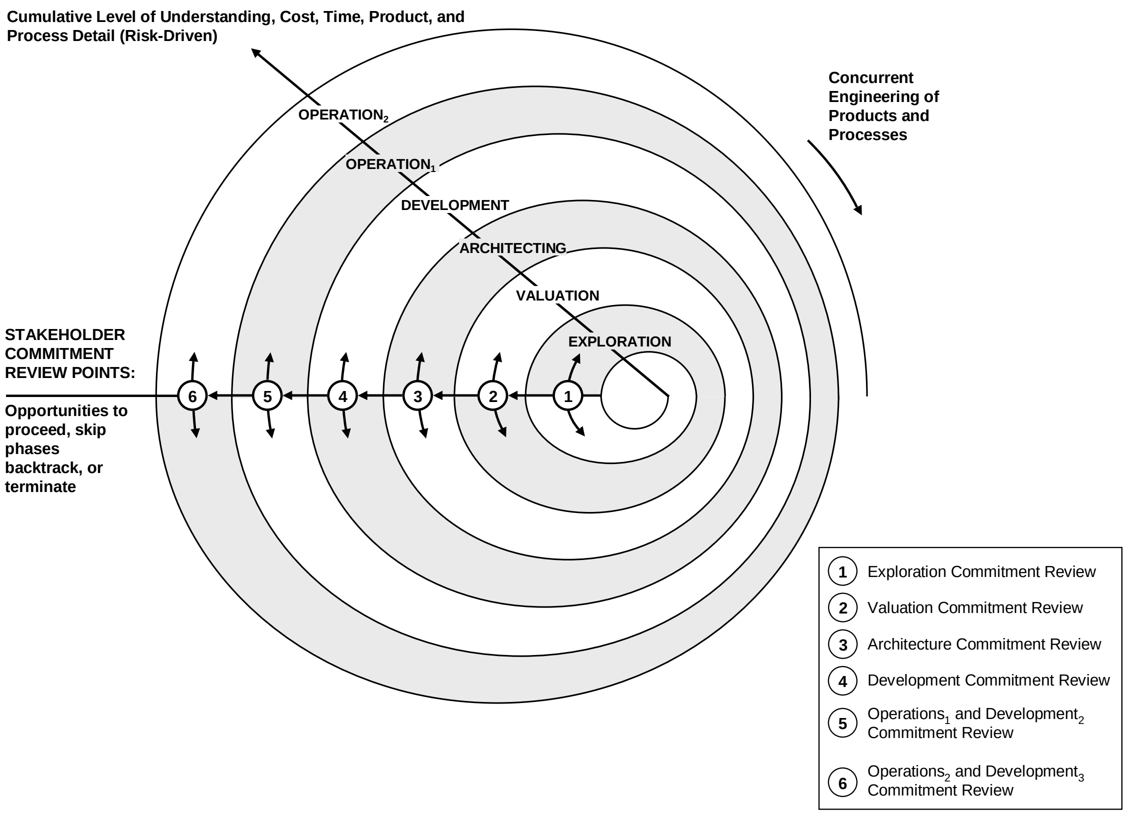


**Figure 4. Simplified Spiral View of the ICM.**

- *Incremental development view for incorporating rapid change and high assurance:* Many future systems and systems of systems will need to simultaneously achieve high assurance and adaptation to both foreseeable and unforeseeable rapid change, while meeting shorter market windows or new defense threats. To achieve these goals, many are adapting to the complexity of these environments by integrating agile teams with more traditional plan-driven (PD) teams and continuous V&V teams. Figure 5 provides an overview of this hybrid process in the systems of systems environment. The agile teams respond to the changing environment and define short, stable increments for development. The plan-driven teams implement capabilities in accordance with the stable increment definitions. The continuous V&V teams support the integration and test of the plan-driven increments. The efforts of the SoS-level team and the supplier teams are planned and synchronized using the anchor point milestone reviews discussed above. More detail on this process, and its staffing and contracting implications, is provided in (Boehm and Lane, 2006).

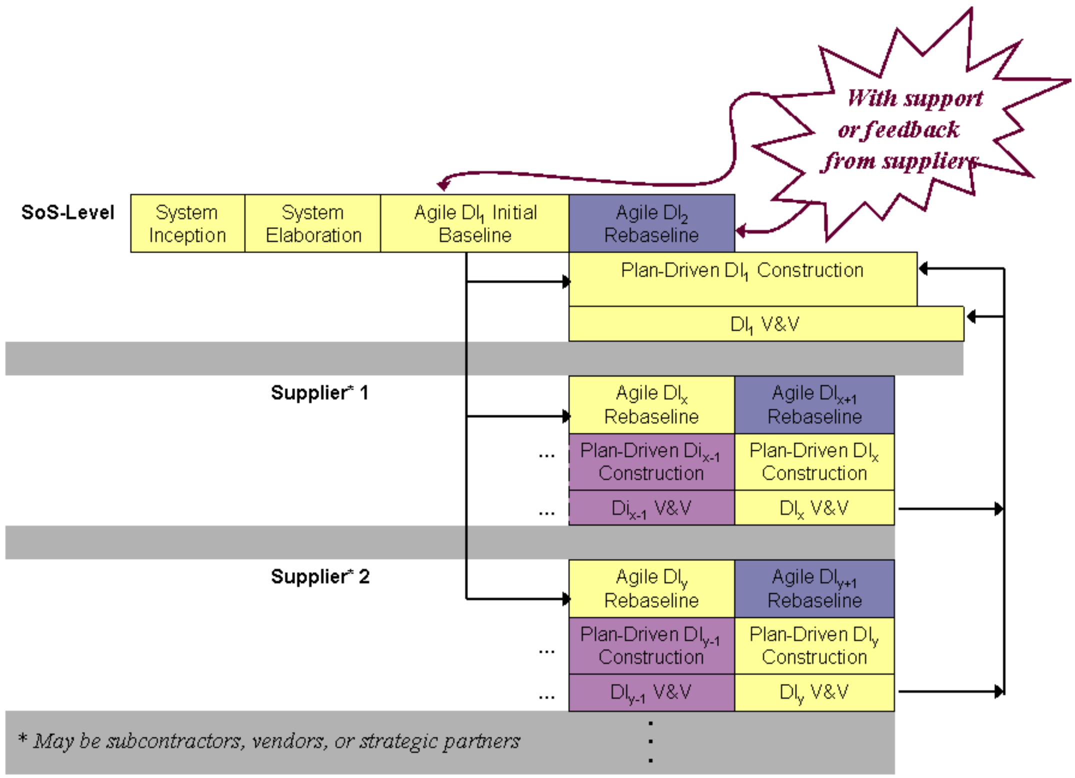


**Figure 5. Using ICM to Define a Hybrid Agile and Plan-Driven Process for SoS Development [Boehm and Lane, 2007].**

## Project Experience with ICM Principles

The ICM uses the critical success factor principles to extend several current spiral-related processes such as the Rational Unified Process, Win-Win Spiral process, and Lean Development process, in ways that more explicitly integrate human-system integration into the system life cycle process. A good source of successful projects that have applied the critical success factor principles is the annual series of Top-5 software-intensive systems projects published in CrossTalk (CrossTalk, 2002-2005).

**Table 2. Number of Top-5 Projects Explicitly Using ICM Principles**

| Year | Concurrent Engineering | Risk-Driven | Evolutionary Growth |
|---|---|---|---|
| 2002 | 4 | 3 | 3 |
| 2003 | 4 | 3 | 2 |
| 2004 | 2 | 2 | 4 |
| 2005 | 4 | 4 | 5 |
| Total (of 20) | 14 | 12 | 14 |

The "Top-5 Quality Software Projects" are chosen annually by panels of leading experts as role models of best practices and successful outcomes. Table 2 summarizes each year's record with respect to usage of four of the five principles: concurrent engineering, risk-driven activities, and evolutionary and iterative system growth (most of the projects were not specific about stakeholder satisficing). Of the 20 top-5 projects in 2002 through 2005, 14 explicitly used concurrent engineering, 12 explicitly used risk-driven development, and 14 explicitly used evolutionary and iterative system growth, while additional projects gave indications of their partial use. Table 3 provides more specifics on the 20 projects (zero, one, or two stars correspond with minimal, partial, and strong application of the CSFs).

**Table 3. Critical Success Factor (CSF) Aspects of Top-5 Software Projects**

| Software Project | CSF Degree | Concurrent Requirements/ Solution Development | Risk-Driven Activities | Evolutionary, Incremental Delivery |
|---|---|---|---|---|
| STARS Air Traffic Control | * | Yes | HCI, Safety | For multiple sites |
| Minuteman III Messaging (HAC/RMPE) | * | Yes | Safety | Yes; block upgrades |
| FA-18 Upgrades | * | Not described | Yes | Yes; block upgrades |
| Census Digital Imaging (DCS2000) | ** | Yes | Yes | No; fixed delivery date |
| FBCB2 Army Tactical C3I | ** | Yes | Yes | Yes |
| Defense Civilian Pay (DCPS) | | No; waterfall | Yes | For multiple organizations |
| Tactical Data Radio (EPLRS) | ** | Yes | Yes | Yes |
| Joint Helmet-Mounted Cueing (JHMCS) | * | Yes; IPT-based | Not described | For multiple aircraft |
| Kwajalein Radar (KMAR) | * | Yes; IPT-based | Not described | For multiple radars |
| One SAF Simulation Testbed (OTB) | ** | Yes | Yes | Yes |
| Advanced Field Artillery (AFATDS) | | Initially waterfall | Not described | Yes; block upgrades |
| Defense Medical Logistics (DMLSS) | | Initially waterfall | Not described | Yes; block upgrades |
| F-18 HOL (H1E SCS) | | Legacy requirements-driven | Yes; COTS, display | No |
| One SAF Objectives System (OOS) | ** | Yes | Yes | Yes |
| Patriot Excalibur (PEX) | ** | Yes; agile | Not described | Yes |
| Lightweight Handheld Fire Control | ** | Yes | Yes | Yes |
| Marines Integrated Pay (MCTFS) | | Initially waterfall | Not described | Yes; block upgrades |
| Near Imaging Field Towers (NIFTI) | ** | Yes; RUP based | Yes | Yes |
| Smart Cam Virtual Cockpit (SC3DF) | ** | Yes | Yes | Yes |
| WARSIM Army Training | ** | Yes | Yes | Yes |

Evidence of successful results of stakeholder satisficing can be found in the annual series of USC e-services projects using the win-win spiral model as described in (Boehm et al., 1998). Since 1998 over 50 user-intensive e-services applications have used the win-win spiral model to achieve a 92% success rate of on-time delivery of stakeholder satisfactory systems.

## Conclusions

Future transformational, network-centric systems will have many usage uncertainties and emergent characteristics. Their hardware, software, and human factors will need to be concurrently engineered, risk-managed, and evolutionarily developed to converge on cost-effective system operations. They will need to be both highly dependable and rapidly adaptable to frequent changes.

The Incremental Commitment Model described in this article builds on experience-based critical success factor principles (stakeholder satisficing, incremental definition, iterative evolutionary growth, concurrent engineering, risk management) and the strengths of existing V, concurrent engineering, spiral, agile, and lean process models to provide a framework for concurrently engineering system-specific critical factors into the systems engineering and systems development processes. It provides capabilities for evaluating the feasibility of proposed solutions; and for integrating feasibility evaluations into decisions on whether and how to proceed further into systems development and operations. It presents several complementary views showing how the principles are applied to perform risk-driven process tailoring and evolutionary growth of a systems definition and realization; to synchronize and stabilize concurrent engineering; and to enable simultaneous high-assurance development and rapid adaptation to change. It analyzes the use of the critical success factor principles on the best-documented government software-intensive system acquisition success stories: the 2002-2005 *Cross Talk* Top-5 projects, and shows that well over half of them explicitly applied the principles.

Unfortunately, the current path of least resistance for a government program manager is to follow a set of legacy regulations, specifications, and standards that select, contract with, and reward developers for doing almost the exact opposite. Most of these legacy instruments emphasize sequential vs. concurrent engineering; risk-insensitive vs. risk-driven processes; early definition of poorly-understood requirements vs. better understanding of needs and opportunities; and slow, unscalable contractual mechanisms for adapting to rapid change.

This article has provided a mapping of the ICM milestones to the current DoD 5000.2 acquisition milestones that shows that they can be quite compatible. It also shows how projects could be organized into stabilized build-to-specification increments that fit current legacy acquisition instruments, along with concurrent agile change-adaptation and V&V functions that need to use alternative contracting methods. Addressing changes of this nature will be important, particularly if organizations are to realize the large potential value offered by investments in future network-centric systems of systems.

## References

Beck, K. (1999) Extreme Programming Explained, Addison Wesley, Reading, MA

Boehm, B. (July 1996). Anchoring the Software Process. Software. pp. 73-82.

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

Boehm, B., and Hansen, W. (May 2001). The Spiral Model as a Tool for Evolutionary Acquisition. CrossTalk. pp. 4-11.

Boehm, B., and Jain, A. (2006) “A value-based theory of systems engineering,” Proceedings, INCOSE 2006.

Boehm, B., and Lane, J. (May 2006). 21st Century Processes for Acquiring 21st Century Systems of Systems. CrossTalk. pp. 4-9.

Boehm, B., Brown, A.W., Basili, V., and Turner, R. (May 2004) “Spiral acquisition of software-intensive systems of systems,” CrossTalk, pp. 4-9.

Boehm, B., Egyed, A., Kwan, J., Port, D., Shah, A., and Madachy, R. (July 1998). Using the WinWin Spiral Model: A Case Study. Computer. pp.33-44.

CrossTalk (2002-2005, “Top Five Quality Software Projects,” January 2002, July 2003, July 2004, September 2005.

Highsmith, J. (2000) Adaptive Software Development. Dorset House, New York, NY

Kruchten, P. (1999). The Rational Unified Process, Addison Wesley.

Maier, M. (1998). “Architecting principles for systems-of-systems, Systems Engineering 1:4, pp. 267-284.

Marenzano, J. et al. (2005) Architecture reviews: Practice and experience. IEEE Software, March/April 2005, pp. 34-43.

Merriam Webster. (2006).Webster’s Collegiate Dictionary.

Patterson, F. (1999). System engineering life cycles: life cycles for research, development, test, and evaluation; acquisition; and planning and marketing, in Sage, A., and Rouse, W., ed., (1999) Handbook of Systems Engineering and Management, Wiley, pp. 59-111.

Pew, R. W., and Mavor, A. S., (2007). Human-System Integration in the System Development Process: A New Look, National Academy Press.

Rechtin, E. (1991). Systems Architecting, Prentice Hall.

Royce, W. W. (1970) . Managing the development of large software systems: concepts and techniques, Proceedings, Wescon 1970.

Royce, W. E. (1998) Software Project Management. Addison Wesley.

Sage, A., and Cuppan, C. (2001). On the systems engineering and management of systems of systems and federations of systems, Information, Knowledge, and Systems Management 2, pp. 325-345.

Sage, A., and Rouse, W., ed. (1999). Handbook of Systems Engineering and Management, Wiley, New York, NY

Sage, A., and Rouse, W. (1999). An Introduction to Systems Engineering and Systems Management, in Sage, A., and Rouse, W., ed. (1999) Handbook of Systems Engineering and Management, Wiley, pp. 1-58.

U.S Department of Defense (2003). DoD Instruction 5000.2, Operation of the Defense Acquisition System.

V-Modell XT (2005). http://www.v-modell-xt.de

**Barry Boehm, Ph.D.**, is the TRW professor of software engineering and director of the Center for Systems and Software Engineering at the University of Southern California. He was previously in technical and management positions at General Dynamics, Rand Corp., TRW, the Defense Advanced Research Projects Agency, where he managed the acquisition of more than $1 billion worth of advanced information technology systems. Dr. Boehm originated the spiral model, the Constructive Cost Model, and the stakeholder win-win approach to software management and requirements negotiation.

University of Southern California
Center for Systems and Software Engineering
941 W. 37th Place, SAL Room 328
Los Angeles, CA 90089-0781
E-mail: boehm@usc.edu

**Jo Ann Lane** is currently a Principal at the University of Southern California Center for Systems and Software Engineering conducting research in the area of system of systems engineering. In this capacity, she is currently working on a cost model to estimate the effort associated with system-of-system architecture definition and integration. She is also a part time instructor teaching software engineering courses at San Diego State University. Prior to this, she was a key technical member of Science Applications International Corporation's Software and Systems Integration Group responsible for the development and integration of software-intensive systems and systems of systems.

University of Southern California
Center for Systems and Software Engineering
941 W. 37th Place, SAL Room 328
Los Angeles, CA 90089-0781
E-mail: jolane@usc.edu