Spiral Development - Building the Culture

A Report on the CSE-SEI Workshop, February, 2000

W. J. Hansen
J. T. Foreman
D. J. Carney
E. C. Forrester
C. P. Graettinger
W. C. Peterson
P. R. Place

July 2000

SPECIAL REPORT
CMU/SEI-2000-SR-006

Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

Spiral Development - Building the Culture

A REPORT ON THE CSE-SEI WORKSHOP, FEBRUARY, 2000

CMU/SEI-2000-SR-006

W. J. Hansen
J. T. Foreman
D. J. Carney
E. C. Forrester
C. P. Graettinger
W. C. Peterson
P. R. Place

July 2000

COTS-Based Systems

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Norton L. Compton, Lt Col., USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

List of Tables

Abstract

A number of organizations are successfully applying the Spiral Development Model (SDM) and finding it valuable in addressing such challenges as rapid development, COTS (commercial-off-the-shelf) software integration, new technologies, and product line management. However, other organizations have experienced difficulties with spiral development—due to over-relaxed controls, underestimated risks, existing sequential development policies, inflexible financing mechanisms, ingrained cultures, and confusion about what spiral development is and how to apply it. To attack these problems, a workshop was held February 9-11, 2000, at the University of Southern California under the sponsorship of its Center for Software Engineering (CSE) and the Software Engineering Institute (SEI) of Carnegie Mellon University. Work groups at the workshop recommended specific actions aimed at building and spreading a culture for the SDM community. These can be described as defining, improving, promoting, and studying SDM, educating about SDM, adapting to SDM, and enhancing teamwork. This report summarizes the workshop and presents its recommendations.

Executive Summary

Abstract: A number of organizations are successfully applying the Spiral Development Model (SDM) and finding it valuable in addressing such challenges as rapid development, COTS (commercial-off-the-shelf) software integration, new technologies, and product line management. However, other organizations have experienced difficulties with spiral development—due to over-relaxed controls, underestimated risks, existing sequential development policies, inflexible financing mechanisms, ingrained cultures, and confusion about what spiral development is and how to apply it. To attack these problems, a workshop was held February 9-11, 2000, at the University of Southern California under the sponsorship of its Center for Software Engineering (CSE) and the Software Engineering Institute (SEI) of Carnegie Mellon University. Work groups at the workshop recommended specific actions aimed at building and spreading a culture for the SDM community. These can be described as defining, improving, promoting, and studying SDM, educating about SDM, adapting to SDM, and enhancing teamwork. This report summarizes the workshop and presents its recommendations.

The workshop objectives were to

- clarify the nature of spiral development
- create a common understanding of the current state of the practice ("as is")
- share experiences in applying it in various situations
- identify its critical success factors
- create a vision of best practice ("to be")
- identify and address institutional barriers/inhibitors to successful spiral usage such as policy, financial, or cultural constraints.

Presentations

The first day and a half of the workshop were devoted to presentations by executives and practitioners representing government, commercial users, solution providers, and contractors. In retrospect, these presentations evoked these themes and comparisons:

- The term *spiral development* is not well defined or understood. For some it means any development approach with recurrent planning activities, while others add constraints such as "risk-based" and "anchor points."
- Spiral development can be sharply defined with *invariants* and *variants*; i.e., those aspects that are essential in every spiral project and those other aspects that can differ between projects.
- *Spiral development* and *evolutionary acquisition* are different, but related. An evolutionary process qualifies technology before embarking on spiral development.
- Spiral development differs between government organizations and commercial organizations.
- Some spiral time cycles are still fairly long--two or three years--while others are much shorter--two to three months. Typically, longer cycles are found in government.

- Some of the critical success factors for spiral development are
  - Risk must be managed.
  - The culture must be trusting.
  - Stakeholders must be involved.
  - The technology must be ready.
  - Requirements must be flexible.

Work Group Recommendations

The second half of the workshop was devoted to small work group sessions, each addressing a different topic. These groups were charged with recommending concrete actions for progress. They made forty-nine recommendations, falling into seven categories:

1. Define SDM. Refine and promulgate the definition of the Spiral Development Model.
2. Promote SDM. Spread awareness of the spiral model among developers, managers, and executives.
3. Educate about SDM. Provide appropriate courses through universities and professional training organizations.
4. Adapt to SDM. Revise policies, processes, and practices to encourage spiral development where appropriate. These recommendations were addressed primarily to the Department of Defense, but will reward use as a checklist for any large organization.
5. Improve SDM. Explore the Spiral Development Model and human behavior to determine what improvements are possible and how they should be formulated.
6. Enhance teamwork. Improve teaming techniques, especially as they apply to spiral development.
7. Study SDM. Conduct research to validate the Spiral Development Model, evaluate its potential for return on investment, and determine the mutual impacts between it and people.

For each recommendation, the work group proposed an action agent, the person or group most appropriate for taking the necessary actions. In general, the *Define SDM*, *Improve SDM*, and *Study SDM* actions are expected to be done by universities and research centers, especially CSE and SEI; all parties must act on *Educate about SDM* and *Promote SDM*; and OSD should *Adapt to SDM* with respect to DoD policies, processes, and practices.

1 Introduction

One approach to improved software production is the *Spiral Development Model* (SDM) [Boehm 88]. In this approach, software is developed in stages. Each stage is a normal development project producing a superset of the prior stage and yet a subset of the final system. Planning for each successive stage is structured to exploit the experiences of the former stages and to reduce perceived risk factors in the current and future iterations. Although numerous spiral projects have succeeded splendidly, SDM has not achieved wide acceptance and has not always produced the results its proponents predict. To study these problems and recommend appropriate actions, the Center for Software Engineering (CSE) of the University of Southern California, and the Software Engineering Institute (SEI) of Carnegie Mellon University, sponsored a workshop, February 9-11, 2000. This report is the result.

In the keynote address, Barry Boehm pointed to LCO and LCA, the life cycle objectives and the life cycle architecture, as critical to success of SDM. He likened their degree of commitment to getting engaged and getting married, which led to a number of facetious remarks in later presentations about various familial relationships. All such relationships occur in the context of a culture; the shared goal of participants was an industry-wide culture supportive of SDM. Hence the subtitle of this report: *Building the Culture.* This same theme echoes another critical success factor, the one that asserts that an organization's internal culture must support SDM. (See Section 2.3.2.)

The initial sessions of the workshop were devoted to presentations on successful and unsuccessful spiral projects. Presenters, as shown in Table 1, were diverse, representing organizations throughout the Department of Defense (DoD), other government agencies, consultants, commercial in-house developers, tools vendors, research organizations, and academia. On the afternoon of the second day, participants divided into five work groups to discuss specific topics. Reports from these groups occupied the third morning.

Section 2 below presents an overview of the workshop, including its themes and a summary of the recommendations from the work groups.

- **Indented blocks in this font are from the slide presentations.**

Section 3 presents the findings of the work groups. Appendices provide sorted lists of the recommendations. Barry Boehm's keynote presentation, with notes, appears as a companion report [Boehm 00]. Supplementary material, including the presentations, can be found on the workshop Web site:

http://www.sei.cmu.edu/activities/cbs/spiral2000.

Table 1: Presentations at the Spiral Workshop

Keynote Address, Wed 8:30	
Boehm, Barry, Director, CSE	*Spiral Development: Experience, Principles, and Refinements*
Executive Perspectives, Wed 9:30-11:00	
Ferguson, Jack, DoD - OSD	*Evolutionary Acquisition in DoD - Updating Acquisition Policy*
Pyster, Arthur, Deputy CIO, FAA	*Increased Responsiveness Through Spiral Process*
DeMillo, Richard, VP, Telcordia	*Telcordia Technologies: Continual Improvement Spiral Software Development*
Commercial Spiral Experience, Wed 11:30-12:30	
Leinbach, Charles, Director, C-Bridge University	*E-Business and Spiral Development*
Hantos, Peter, Manager, Xerox	*From Spiral to Anchored Processes: A Wild Ride in Lifecycle Architecting*
Government/Aerospace Spiral Experience, Wed 1:30-3:00	
Royce, Walker, VP, Rational	*The Rational Unified Process - A Commercially Available Spiral Model Implementation*
McNutt, Ross, DoD - Air Force	*Reducing Air Force Acquisition Response Times and Spiral Development*
Kitaoka, Beverly, Senior VP, SAIC (presented by Ron Warfel)	*Yesterday, Today & Tomorrow - Implementations of the Development Lifecycles*
Solution Provider Experiences, Wed 3:30-5:00	
Cross, Steve, Director, SEI	*A "Solution Provider Experience" from an "Old Practitioner"*
Finneran, Lisa, VP/CTO, SPC	*Lessons Learned from Applying the Evolutionary Spiral Process*
Saunders, Thomas, Executive Director, MITRE	*Spiral Acquisition and the Integrated Command and Control System*
Spiral Experience Presentations, Thurs 8:00-11:30	
McKinney, Dorothy, Lockheed Martin Mission Systems (presented by Bruce Long)	*Impact of Spiral Development on Integrating Software and Systems*
Bernstein, Larry, Have Laptop-Will Travel	*Importance of Software Prototyping and the Spiral Model*
Willhite, Anne Marie, MITRE	*ESC's Spiral Initiative (A Program Perspective)*
McKee, Larry, MITRE	*Aerospace C2ISR "Spiral Development"*
Bostelaar, Tom, TRW	*TRW Spiral Development Experience on Command & Control Product Line Programs*
Razouk, Rami, Aerospace	*Spiral Development Experience Report*

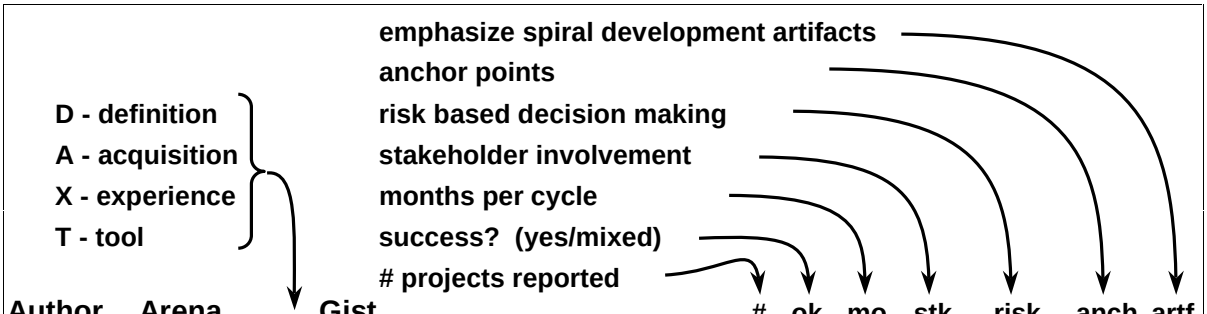
2 Workshop Overview

The Spiral Development Workshop met in the large hall of the Davidson Conference Center at the University of Southern California (USC). The five dozen attendees were seated at seven rows of four tables with a central aisle. Time was managed with a PowerPoint application showing the minutes remaining; when time expired, it applauded.

Presenters spanned the universe of developers as indicated in the second column of Table 2. Speakers were evenly divided between the Department of Defense (DoD) and the commercial sector. The DoD contingent had two speakers from the Department itself, three from MITRE personnel consulting to DoD, and three from contractors working to deliver software to the DoD. Speakers from the commercial sector included three companies developing software for in-house use, two consultancies that create and deliver software, one tool vendor, one contractor (Razouk), and the FAA. In addition, there were speakers from the two research university hosts, CSE and SEI. (The difference between contractors and consultants is one of degree. For this discussion, contractors are engaged for long terms and operate independently while consultants are hired for a short term and work jointly with employees.)

The third column of Table 2 categorizes the talks. (D) is for Professor Barry Boehm's Definition of spiral development. (A) marks talks from the DoD that focused on the acquisition of software. (T) denotes talks about tools or canned approaches to developing software with a spiral model. (X) marks the largest group, referring to talks describing experiences in doing development with the spiral model. For these, the fifth column (#) gives the number of projects covered and the sixth shows that ten presentations claimed success and five achieved mixed results. The remaining five columns touch on aspects of the definition of SDM and will be discussed below.

Table 2: Some Presentation Notes



Author	Arena		Gist	#	ok	mo	stk	risk	anch	artf
Boehm	university	D	definition of SDM	2	y		yes	yes	yes	yes
Ferguson	DoD	A	evolutionary acquisition							
Pyster	FAA	X	example of spiral development	1	y		yes	on HCI		
DeMillo	in-house	T	S/W tools for large projects							
Leinbach	consultant	T	benefits of “RAPID”		y	1-3	yes		yes	
Hantos	in-house	X	Xerox adopts SDM	4	m			yes	yes	
Royce	tool vendor	T	benefits of “RUP”		y		yes			
McNutt	DoD	A	evolutionary acquisition	1	y	24	yes			
Kitaoka	contractor	X	tailor SDM to the project	3	m		yes	yes		
Cross	university	X	an early spiral; work at SEI	1	y	6	yes	yes		
Finneran	consultant	T	benefits of "ESP"		y	2-9	yes	yes		yes
Saunders	MITRE	A	conditions for SDM; JBI	6	m					
McKinney	contractor	X	summarize spiral experiences	++	m					
Bernstein	in-house	X	prototyping	2	y					
Willhite	MITRE	A	SDM in an acquisition organization				yes	yes		
McKee	MITRE	A	evolutionary acquisition	76	m		yes		yes	yes
Bostelaar	contractor	X	development of weather analysis product line	1	y	12				
Razouk	contractor	X	compare spiral projects	5	y	2-12				

Despite the presenter diversity, the workshop suffered a glaring absence of users. After the conference Thomas Saunders deplored this fact. He remarked that as a result of the absence the workshop had ignored one of the most persistent problems raised during earlier meetings: that lack of user buy-in is a significant inhibitor to the acceptability of spiral developments. Overlooked issues included

a. how to get users to accept a first delivery that only partially satisfies requirements and to await upgrade increments that more closely satisfy requirements
b. how to give users confidence that they can split up their funding to cover a multi-increment delivery process
c. how to test incremental releases to allay concerns by users and testers over trustworthiness and reliability of the system
d. how to get the user community trained on the new system. Rapid redeployment can rob users of the ability to properly exploit a system’s capabilities. Training, then retraining, then retraining again at a rapid pace, is difficult for users to tolerate. The spiral model has to balance training time against the “usage life” of a system.

2.1 DoD and Non-DoD

Clear differences separated the DoD half of the presentations from the others. Three of them follow.

2.1.1 "Spiral Development" or "Evolutionary Acquisition"

Rather than develop software, DoD "acquires" it, usually from one of a few large specialized contractors. The acquisition process is evolving toward "evolutionary acquisition," which adapts the concept of risk reduction from spiral development to earlier phases of new technology exploitation. This approach is a highlight of the new draft DoD acquisition policy, the 5000 series, as presented at the workshop by Jack Ferguson, Director of Software Intensive Systems for the Office of the Secretary of Defense (OSD). As shown in Figure 1, two phases precede development. The Science and Technology phase explores options and arrives at a possible project. The Demonstration and Risk reduction phase develops a prototype or mock-up to show that the project can be made to work outside the laboratory. After each of these phases a formal decision is made as to whether to proceed. Another decision point is scheduled part way into the development phase. Development is funded in a sequence of "blocks" or "increments" of one or two years each. As shown at the bottom, Test and Evaluation is part of the project effort from late in the Science and Technology phase until the project is cancelled or becomes operational.

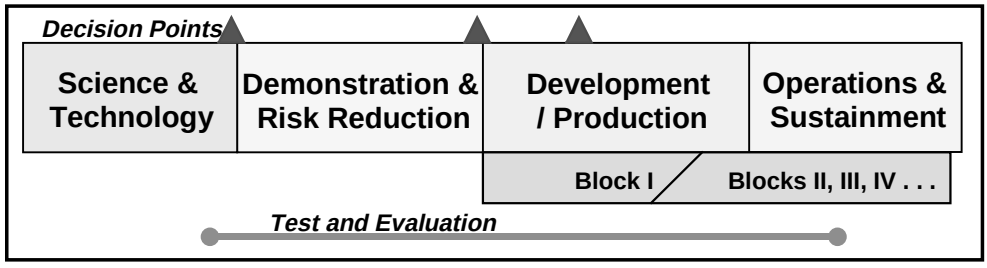


Figure 1: Sketch of New Evolutionary Acquisition Approach (Ferguson)

Evolutionary acquisition shares with SDM the risk-based management approach. The technology risks are mitigated by explorations of technology in the first two phases before a major development project is initiated. Spiral development itself is encouraged for use by contractors performing the Development/Production phase. There is no necessary relation between funding blocks and spiral iterations. As Ross McNutt's presentation noted: "Evolutionary acquisition is a strategy" while "Spiral development is a process."

2.1.2 Employees or Contractors

DoD software is acquired from contractors whose employees produce it. Software developed for organizations other than the DoD is usually either purchased or developed by the organization's own employees. The use of contractor personnel introduces a level of separation. The gap is widened to an abyss when contractors are located in an area remote

from the eventual users. Dorothy McKinney reported on DoD projects from the contractor perspective and noted that

- **Contractual provisions were not always conducive to effective use of the spiral approach**

  Cooperation between stakeholders needed for effective spiral development often clashed with desire to preserve contractual leverage

In other words, contractors anticipate more power and profit from non-spiral approaches and users are unwilling to accept a lessening of their power to hew strictly to the mark of predefined requirements.

2.1.3 Cycle times vary widely

The seventh column of Table 2, labeled "mo," shows the number of months for each spiral, where that information was reported. Invariably, DoD project cycles are a year or more and non-DoD projects are less than a year, often far less. There is a tendency in DoD work to equate a spiral cycle with a funding block or increment. McKinney reported that on one large collection of development efforts

- **The number of spirals varied from 2 to over 20, but was typically 4 - 10**
- **The time to complete all of the spirals has varied from 6 months to 10+ years**

The detrimental length of DoD development cycles has been reported to Congress by Dr. J. Gansler, Undersecretary of Defense for Acquisition Technology and Logistics [Gansler 2000]. He observed that

> *… with current cycle times in information technology now measured in months, our traditional, protracted, multi-year (often ten to twenty years) defense development methods simply cannot keep pace.*

In remarks after the workshop, Saunders observed that

> *Most of the development experience in the workshop related to large monolithic system development and acquisition. Software components with useful functionality can be delivered in "internet-time," so there may need to be a revision to the strategy for ownership, configuration control, funding, and contracting involved in assembling a new software capability. Workshop discussions persisted in drawing upon a mental model based on how systems used to be acquired. If there is a new acquisition approach involving testing, field trials, integrate after delivery, etc., then the 5000 regulation series needs to allow that. This is particularly true if new capabilities are introduced as system service "plug-in's" instead of a large stand alone capability. For example, data definition standardization starts to become more important to preserve interoperability amongst all the participating services... likewise, interaction protocols that mimic internet/web-based applications may need to be established to handle how the interfaces work when new components first come together.*

Typical of commercial cycles was the report from Charles Leinbach. His company's approach, called "RAPID," is used in delivering e-business solutions to customers. A key is that lessons learned in one development effort are codified and applied to developing for the next customer, in a process called Profit Life Cycle Management. One spiral model is used for program management. Another, shown in Figure 2, applies to project implementation. Typically, the company expects to spend 12 to 24 weeks from inception to final delivery with several iterations for each of the four phases: Define, Design, Develop, and Deploy.

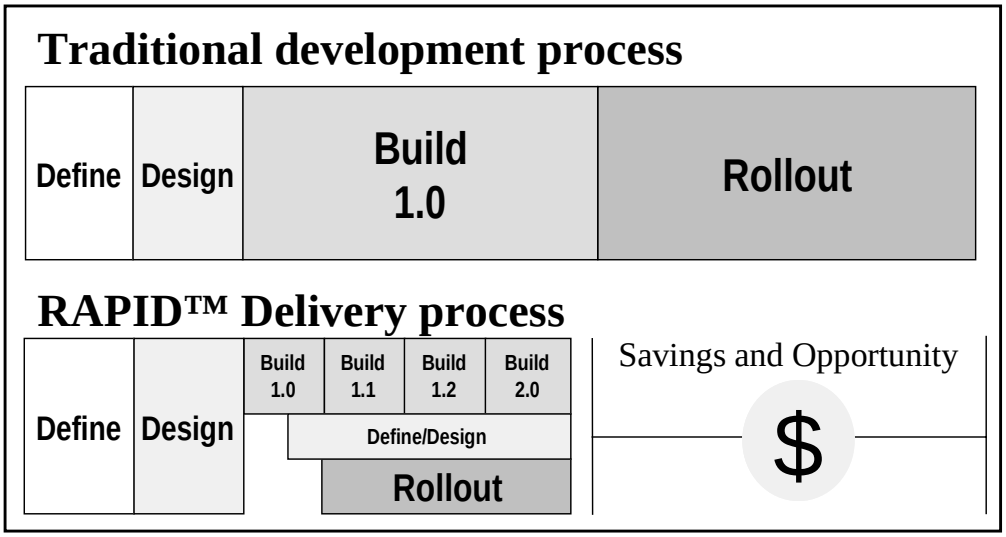


Figure 2: C-Bridge's RAPID Delivery Process (Leinbach)

Lisa Finneran was more explicit about what cycle times should be and why:

- **Cycle lengths of 2-9 months seem to be normal**
- **Short cycles cause too much overhead**
- **Long cycles cause too much replanning**

2.2 What is the Spiral Development Model?

The term *spiral development* is not well defined or understood. For some it means any development approach with recurrent planning activities, while others add constraints such as "risk-based' and "anchor points." Rami Razouk sketched five spiral development projects and asked the audience with respect to each whether it was spiral development or not. There was usually general agreement, but always some uncertainty. Razouk never did say what he felt were the "right" answers, but he did note that the following were common occurrences.

- **A wide range of definitions of spiral development**
- **A mismatch between HW and SW lifecycles**
- **A strong tendency to do the easy things first**

(This can be viewed as the right thing to do if you consider the highest risk to your program to be the need to prove that it will be useful)

- **Lack of understanding of the principles behind spiral development**

  (monotonically decreasing risk)

Dorothy McKinney's presentation reflected the practitioner's uncertainty as to the definition of SDM:

- **Spiral development was used on some programs on purpose, and inadvertently at first on others (when requirements were discovered/clarified only after use of the result of what in retrospect proved to be the first spiral)**

Barry Boehm attempted in his keynote address [Boehm 00] to define spiral development. His approach was to enumerate a set of six properties that every spiral development project must have, as shown in Table 3. These are called "invariants" because they must invariably appear in all spiral development projects.

Table 3: Barry Boehm's List of Invariants

Invariant 1:	**Concurrent determination of key artifacts (operations concept, requirements, design, code, plans)**
Invariant 2:	**Each cycle does all of these: objectives, constraints, alternatives, risks, review, commitment to proceed**
Invariant 3:	**Level of effort is driven by risk considerations**
Invariant 4:	**Degree of detail is driven by risk considerations**
Invariant 5:	**Anchor point milestones are produced**
Invariant 6:	**Emphasis on system and life cycle activities and artifacts**

The "anchor point milestones" in Invariant 5 are artifacts to be produced during spiral development. They are denoted by LCO, LCA, and IOC, which stand for Life Cycle Objectives, Life Cycle Architecture, and Initial Operating Capability. Each is produced as a by-product in some one chosen cycle, although it may evolve in subsequent cycles. Their purpose is to "anchor" the spiral process so it progresses toward goals that are comparable from one project to another.

Projects reported at the workshop exhibited the spiral invariants to varying degrees, as shown by the last four columns of Table 2. (Projects with blanks did not report the information; they may have followed the invariants in practice.) The stk column indicates that stakeholders were involved in every stage of development. The risk column shows that the project reported performing management and decision making so as to control the degree of risk. The anch column shows that lifecycle anchor points were mentioned in that presentation. The artf column shows that it was clear that the development method emphasized spiral development artifacts rather than software development artifacts.

Boehm's paper elaborates on the invariants in several directions. Each is described in terms of practical examples. Each is augmented with a number of "variant" subdimensions showing how the spiral development process can be varied to tailor a project

to its circumstances. Finally, the presentation lists “hazardous spiral look-alikes,” as shown in Table 4. Knowledge of these look-alikes can help the practitioner avoid repeating the approaches of previous unsuccessful attempts at spiral development.

Table 4: Hazardous Spiral Look-Alikes

- **incremental sequential waterfalls with significant COTS*, user interface, or technology risks**
- **sequential spiral phases with key stakeholders excluded from phases**
- **risk-insensitive evolutionary or incremental development**
- **suboptimizing increment 1 with a point-solution architecture which must be dropped or heavily reworked to accommodate future increments**
- **evolutionary development with no life-cycle architecture (LCA)**
- **insistence on complete specs for COTS, user interface, or deferred-decision situations**
- **purely logical object-oriented methods with operational, performance, or cost risks**
- **impeccable spiral plan with no commitment to managing risks**

*** commercial-off-the-shelf**

The presentation also summarizes the invariants into a candidate definition of the Spiral Development Model, to serve as a starting point for a concise consensus definition:

The Spiral Development Model is a risk-driven process model generator for guiding multi-stakeholder concurrent engineering of software-intensive systems. Its distinguishing features include a cyclic approach for incrementally growing a system's degree of definition and implementation, and a set of anchor point milestones for ensuring feasibility of the incremental definitions and implementations.

Early efforts at adopting SDM at Xerox were not completely successful. Peter Hantos attributed this in part to lack of lifecycle anchor points, and noted that

- **During the phases not only the prototypes became “throwaways,” but the architecture versions as well**
- **Risk analysis was superficial, and also inefficient**
- **As a result, architecture never stabilized**
- **Overly aggressive plan created an overload of new technologies**
- **Technology experimentation obfuscated architecture development**
- **Resolution of technology risks was overwhelming, further preventing the stabilization of the architecture**

Anchor points reduce the impact of these problems by giving specific criteria for advancement from one project phase to another, even when each phase is more than a single iteration through the spiral.

Some appreciation of the nature of SDM can be gained by describing the nature of projects for which it is suited. Beverly Kitaoka took this approach in presenting a chart listing a number of development methods and the characteristics of projects suitable for that method. An extract of her chart #12 is shown as Table 5, below. In her terms, a spiral project will have these lifecycle features:

- **requirements are not predefined**
- **there are multiple internal development cycles**
- **there may be multiple deliveries to the customer**
- **requirements are the primary driver of functional content**
- **risk reduction is the primary driver of the process**

Using the full set of properties from Kitaoka's original table, SDM can be selected if

- **the system is unprecedented**
- **requirements are not well-understood**
- **total funding is not known a priori**
- **engineering sub-projects are risk-driven**

Table 5: Process Selection Criteria

	Lifecycle Features					Lifecycle Selection Guidance			
	Requirements Defined First	**Multiple Internal Development Cycles**	**Functional Content Primary Driver**	**Process Primary Driver**		**Precedented System**	**Requirements Understood**	**Technology Understood & Stable**	**Large or Complex**
Waterfall	Yes	Y/N	Reqmts	Well Understood and Stable		Y	Y	Y	N
Incremental	Yes	Y/N	Reqmts	Resource Constraints #2		Y	Y	Y	Y
Evolutionary	No	Y/N	Reqmts	Requirements or technology not well understood or unstable		N	N	---	---
COTS Integration	Y/N	Y/N	COTS Avail'ble	Availability of suitable COTS products		Y	---	Y	---
Spiral	No	Yes	Reqmts	Risk reduction		N	N	---	---

With respect to the Rational Unified Process (RUP), Walker Royce observed

- **Biggest weakness: too easy to interpret as cookbook**

  Still requires domain tailoring, common sense to be added

This applies as well to most process definitions, including SDM. The method must be interpreted and adapted for each particular situation. Otherwise too much may be done on trivial aspects while crucial points outside the usual pathways are ignored until too late.

2.3 Critical Success Factors for SDM

Many of the talks reported actual experiences. Among the dramatic results were those shown by Figure 3, which is taken from the presentation by R. A. DeMillo. Field fault densities declined by 62% over three years with the aid of SDM, four tools described in the talk, and an organization having attained Capability Maturity Model® (CMM®) Level 5 status.

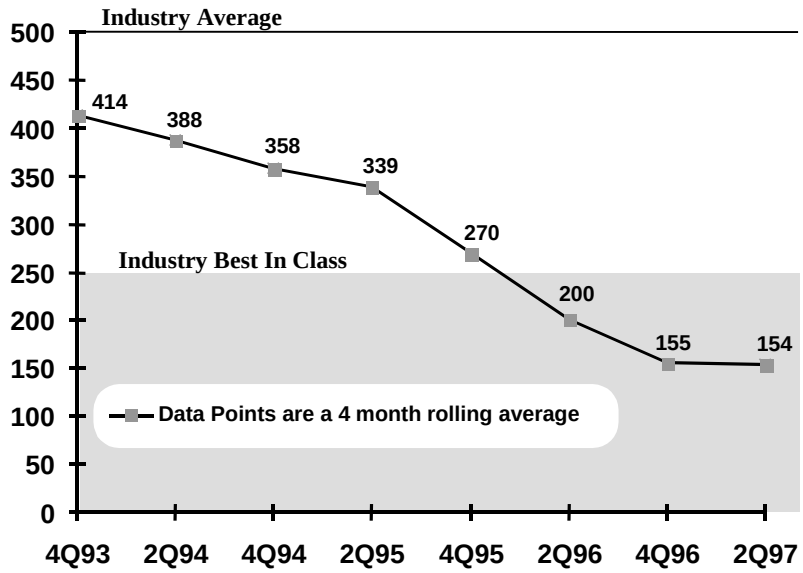


Figure 3: Telcordia Results – Cumulative Field Fault Density, Aggregate of All Business Units (DeMillo)

The observations and reported experiences of the presenters touched a number of common themes. From these we can read some conditions necessary for the success of SDM efforts and converse conditions under which failure is likely. In the following subsections we review these conditions as "Critical Success Factors" necessary for successful use of SDM.

2.3.1 Risk Must be Managed

Apart from cycles, the most important characteristic of spiral development is management of risks. Presentations stating this positively included Razouk's comment above about "monotonically decreasing risk" and a comment from Steve Cross observing that development efforts must "Resolve highest risks first." Speaking of unsatisfactory projects, Razouk observed that there was

® Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office.

- **Generally poor understanding of the role of risk management as a key element of spiral development**
- **Very poor execution of risk management during spirals (schedule pressure)**

while McKinney noted the following:

- **Customer and company management expectations did not always match the results spiral development can/did produce**
- **80% solutions from early spirals raised unrealistic expectations about total project time and cost**
- **Movement of capability to later spirals in order to meet schedule targets without replanning the resources allocated to the later spirals**
- **Risk management which did not address the full range of risks (often for political reasons)**

The remaining subsections all deal at some level with methods to control risk. In particular, the "political reasons" just mentioned are usually a factor of the prevailing culture.

2.3.2 The Culture Must be Trusting

The behavior of people in an organization depends in part on that organization's culture; that is, its beliefs and its expected modes of personal interaction. An "untrusting" culture is one where outsiders are treated with suspicion, ideas are unwanted, and areas of responsibility are jealously guarded. In contrast, a "trusting" culture welcomes outsiders, embraces new ideas, and fosters cooperation on areas of responsibility. In an untrusting culture, it is difficult to manage risk because those expressing risk are treated as responsible for the problem and may suffer career consequences.

Cross describes commercial best practices of trusting cultures:

- **Buyer, user, and vendor are a team. Attitude of partnership, trust and cooperation. Presumption of trustworthiness for reputable commercial organizations.**

In contrast, government practices are untrusting. He observes an

- **"Us vs. them" mentality about contractors. Government thinks in terms of control, accountability, detailed auditing, and double-checking. Presumption that contractors cannot be trusted.**

Similar remarks are made by Saunders who notes

- **Culture, Business Models, or Objectives clash**

  Risk-averse acquisition culture adds encumbering processes
  Contractor expectations (fear of requirements creep)

Arthur Pyster reported that six FAA projects have adopted SDM to some extent and that valuable results are being achieved. Nonetheless,

- **Measurement and visible management of risk is culturally challenging for the FAA, but there are successful examples emerging.**

Work Group 3 expressed the need for a trusting culture this way:

> *The key to successful system development is collaboration between the DoD and the contractors. If this can be achieved, through any mechanism, DoD will be able to acquire systems more successfully. Anecdotal evidence suggests that when collaboration has occurred systems have been developed in an economical and timely fashion. Spiral development was seen as requiring a shift in mindsets where collaboration and cooperation is the norm rather than the exception.*

2.3.3 Stakeholders Must be Involved

The stakeholders of a project include all those with financial and operational risks: senior management who funds the effort and expects a performance payoff, managers responsible for the operation affected, and the operators who will interact with the developed system. Neglect of any of these groups can mean additional expense or eventual failure.

The SDM process requires that stakeholders be consulted early in *every* cycle and that they help decide the development agenda for the cycle. Boehm illustrates this point with an example of development of a web-based image viewer for USC based on a COTS product. The first product selected had excellent capabilities, but only for one of the three operating systems in use on campus. There were promises of support for the other operating systems, but these receded into an unknown future. Boehm states that more involvement of all three user communities could have avoided wasting effort on that first product before it abandoning it, as later happened, in favor of another COTS image viewer compatible with all three operating systems.

The current choice for stakeholder involvement is the integrated product team (IPT), as noted by Saunders in a list of conditions for spiral success: “Functional and empowered IPTs.” He goes on to prescribe

- **Beta site users and usability testing is included from program start**
  (and the most challenging requirements are explored early!)

Cross cites another list of stakeholders and suggestions for their work:

- **Build the team (developers, users, acquirers)**
  create shared understanding based on use case
  agree to measures of success
  rehearse key processes

Sporadic involvement of stakeholders may not be enough. Thomas Bostelaar recommends

- **Customer / contractor communication**
  collocation
  peer level interchange with upward /downward review of recommendations/decisions

McKinney notes two problems that arise in trying to accomplish stakeholder involvement:

- **Major changes in key personnel in more than one stakeholder, where expectations and (non-contractual) agreements were not passed on effectively**
- **Tension between getting much coverage of Ops Concept (for obtaining stakeholder buy-in) versus solving hard technical problem (to establish technical feasibility early)**

2.3.4 The Technology Must Be Ready

One of the key problems for DoD is that it attempts to stay at the technological leading edge in order to field the most advanced fighting forces possible. As a consequence, new systems often employ new and untested technology. Experience has shown that projects are likely to fail unless the underlying technology has reached Level 6 on a scale of nine levels of "technology readiness." (See Table 6, which Ferguson derived from a Government Accounting Office report [GAO 1999]). This experience is being written into the new 5000 series acquisition policy. This same notion of waiting for mature technology before developing it into a system is also apparent in the presentations on evolutionary acquisition by McKee, McNutt, and Anne Willhite.

Table 6: Technology Readiness Levels

1. **Basic principles observed and reported**
2. **Technology concept and/or application formulated.**
3. **Analytical and experimental critical function and/or characteristic proof of concept.**
4. **Component and/or breadboard validation in laboratory environment.**
5. **Component and/or breadboard validation in relevant environment.**
6. **System/subsystem model or prototype demonstration in a relevant environment.**
7. **System prototype demonstration in an operational environment.**
8. **Actual system completed and "flight qualified" through test and demonstration.**
9. **Actual system "fight proven" through successful mission operations.**

From the commercial world outside the DoD, Hantos also observed the need for technological readiness this way:

> **It needs to be demonstrated that the hardware will be manufacturable, and neither the hardware nor the software will need extraordinary, open-ended efforts during the development and manufacturing process phases.**
>
> - It is **o.k.** to combine Research and Technology
> - It is **not o.k**. to combine Research or Technology Development with Product Development

Larry Bernstein reported two prototype efforts. The first effort showed that the prototyped approach was inadequate and another approach was adopted. In the second effort, the prototype succeeded and became the basis of the final system. In both, the prototyping effort ensured technology readiness of at least Level 6, so the eventual success of both projects was not an accident.

2.3.5 Requirements Must Be Flexible

The Spiral Development Model is particularly well suited to development of systems where the requirements can evolve over the course of the project. This is common practice in industry, as described by Cross, who observed

- **More detailed analysis of cost versus feature. Dropping lower value/higher cost options or reducing requirements is practiced.**
- **More requirements trade-off decisions (involving complexity and schedule) for reduced time to field.**

In contrast, government software is developed inflexibly. Cross also described

- **Very little flexibility to trade-off requirements creep versus complexity and schedule.**
- **Little or no requirements reductions on high cost items.**

Boehm illustrated the value of flexibility with a description of an information query and analysis system. The contract was written to require a one-second maximum response time, which turned out, after 2000 pages of design and documentation were written, to cost $100 million. At that point a prototype, which would have been created sooner had the requirement been more flexible, showed that four-second response time was acceptable and would cost a third as much.

In a report on a particular spiral process used for commercial development, Lisa Finnernan noted that one reason for failure was that the process did not work “when the initial plan could not be updated.”

Bostelaar relates the evolution of requirements to the culture by recommending that

- **Effective requirements change management / tracking process**
  government personnel aware of requirements creep impact,
  requirements priority and deferral process

One of the important tools for describing systems is “use cases,” a technique in which threads of user behavior are described. As requirements evolve, so must the system description embodied in the use cases. Cross recommends the following:

- **Map evolving requirements to use case**
  analyze evolving requirements in "living" document
  evaluate tradeoff decisions in context of use case
  avoid boundless expectations by users

2.3.6 Other Critical Success Factors

The critical success factors in the preceding sections—risk management, a trusting culture, involved stakeholders, technology readiness, and flexible requirements—all appeared in multiple presentations and are undeniably critical. Other critical success factors are noted in the Work Group reports in Part 3. These include

- contract vehicles supportive of SDM
- short cycles
- regression testing
- comprehensive test planning
- experimental validation of SDM
- return-on-investment data
- consistent/persistent application of IPT principles and practices
- knowledge capture and organizational learning
- distributed performance appraisal methods
- availability of training and education on SDM
- cost and effort estimating methods for SDM

All groups made recommendations for actions on those factors they thought critical. These recommendations follow.

2.4 Summary of Recommendations

A major goal of the Spiral Development Workshop was to foster further work in the field, directing it to ends that appear fruitful. To this end, each work group recommended one or more appropriate actions. In this summary, the recommendations are considered in these classes:

- **Define SDM**. Refine and promulgate the definition of the spiral development model.
- **Promote SDM**. Spread awareness of the Spiral Development Model among developers, managers, and executives.
- **Educate about SDM**. Provide courses in universities and professional training organizations.
- **Adapt to SDM**. Revise policies, processes, and practices to encourage spiral development where appropriate, especially in the Department of Defense.
- **Improve SDM**. Explore the model and related human behavior to determine what improvements are possible and how they should be formulated.
- **Enhance teamwork**. Improve teaming techniques, especially as they apply to spiral development.
- **Study SDM**. Conduct research to validate the Spiral Development Model and evaluate its potential for return on investment.

The full set of recommendations are in the Work Group reports in Part 3 and are summarized in Appendix A. Each action class is discussed in a subsequent section. In these sections, recommendations are denoted in the form WG1:2, where the first number is that of the work group and the second is the recommendation number assigned by that group.

Define SDM

Recommendations WG1:1, WG4:S1, WG5:1, WG5:6

Participants in several of the work groups felt that SDM was not understood by practitioners because it is not—or is not perceived to be—well defined. Further work on the definition should be the aim of a follow-on workshop and could utilize the notions of variants and invariants as presented by Barry Boehm. One group recommended that the definition be strengthened to explicitly require "collective decision making." Another recommendation called for a parallel definition of evolutionary acquisition in sufficient detail to discriminate it from spiral development.

Promote SDM

Recommendations WG2:2, WG3:1, WG4:S3, WG4:S11, WG4:L1, WG4:L6, WG5:5

Efforts are needed, according to group members, to widen the awareness among developers, managers, and executives of the Spiral Development Model as a valuable tool for system development. Particular efforts espoused include the authoring and publication of books, the introduction or expansion of SDM in university courses, and the publishing and publicizing of case studies of spirally organized developments in both the commercial and governmental sectors. Instructional projects must utilize the spiral model so that students will be exposed to its workings from a hands-on level. Additional workshops should be directed toward building a community of SDM practitioners and managers. More specific actions are to work toward having the spiral model as part of the Project Management Institute's curriculum and the Capability Maturity Model Integration[SM] (CMMI[SM]) framework.

Educate about SDM

Recommendations WG2:1, WG4:S2, WG4:S6, WG4:S8, WG4:S17, WG4:L2, WG4:L5, WG4:R5, WG5:4

Beyond promotion of SDM with its introduction of SDM into existing courses, there is a need to develop entire courses on the model. In undergraduate and post-graduate formal education, the spiral model must be integrated into textbooks and courses. The professional training community, including the Project Management Institute, must offer a range of courses to present the method at a variety of levels: practitioner, contracting officer, manager, or executive. More specific recommendations involve writing a "field

[SM] Capability Maturity Model Integration and CMMI are service marks of Carnegie Mellon University.

guide" on SDM, incorporating material from such a guide into the nascent Electronic Systems Command (ESC) handbook on SDM, and doing a technology transfer of knowledge management methods and tools.

Adapt to SDM

Recommendations WG1:2, WG1:3, WG1:4, WG1:6, WG2:3a,b,c, WG4:S4, WG4:S5, WG4:L4, WG5:2

This set of recommendations suggests how organizations can adapt themselves to take full advantage of the benefits of SDM. Although originally directed at the DoD, most of these are applicable to all development organizations. In the short term, participants should brief their management on the workshop and the results reflected in this report. Longer-term efforts require adaptation of existing policies and practices in directions such as streamlining contracting and using Integrated Product Teams. Organizations should not just permit new methods, but should pro-actively offer incentives for using SDM, building trust, and encouraging communication between in-house research groups. More specifically, the DoD should start a pilot project under Warfighter Rapid Acquisition Program (WRAP) and work with Congress on the funding model for DoD projects.

Improve SDM

Recommendations WG1:5, WG2:4, WG4:S10, WG4:L3, WG4:L7

SDM, like any human process, can be improved, as the work groups made clear by recommending actions at each phase of a spiral effort. One group noted that personnel selection can be improved by defining "core competencies" in evolutionary acquisition; the same applies to SDM itself. At the start of each spiral iteration, risks must be exposed and evaluated. This can be improved by providing incentives for identifying risks. The work during the cycle can be improved by use of tools designed to support SDM; these need to be created and then adopted. As a development phase completes, its results must be tested; three specific testing techniques are recommended. Finally, the project must be assessed and to do so there must be developed instruments adapted to assessing spiral projects.

Study SDM

Recommendations WG4:S7, WG4:R1, WG4:R2, WG4:R3, WG5:3

Although attendees at the workshop were of the opinion that SDM is a valuable approach, this is not universally recognized. The case for SDM must be made. One starting point is to prepare and publish a collection of data on existing spiral development efforts. On this basis the business case for spiral development must be made, including evaluation of its return on investment (ROI). Continuing efforts need to address the capability of SDM to scale to larger projects. The human side of SDM must be addressed by exploring its impact on culture, policy, and practice. Even though experiments comparing methods with these sorts of impacts are notoriously difficult to control, it is important to

undertake experimental validation of SDM in order to gain a deeper understanding of its value.

Enhance Teamwork

Recommendations WG4:S9, WG4:S12, WG4:S13, WG4:S14, WG4:S15, WG4:S16, WG4:S18, WG4:L8, WG4:L9, WG4:R4

"Teams" are an important aspect of practical implementation of the Spiral Development Model. The model demands risk exposure and assessment; neither of these tasks can be done by any group sharing only one or two work roles. A number of recommendations are directed at those implementing SDM, whether in government or industry; plans should include recognition that team building is a risk that needs to be managed and that this risk can be mitigated with IPT training and extensive early face-to-face meetings. Teams must be taught how to identify and select incentives to encourage participation; this must be augmented with distributed performance appraisal methods, which are available, but not yet widely used. It is increasingly common for members of a team to be separated by distances greater than a short walk; one recommendation calls for studying the return on investment of this distributed approach versus co-location of the team. Other recommendations call for studying the human factors surrounding use of existing tools and for developing new, improved tools. A specific recommendation was made that a pilot project in distributed teaming be conducted with the web-casting technology available at USC.

2.5 Conclusions

The Spiral Development Workshop featured presentations and work groups. The presentations were widely varied, but did suggest a few themes and some factors critical to the success of the Spiral Development Model (SDM). The work groups arrived at a set of forty-nine recommendations in seven categories: Define SDM, Promote SDM, Educate about SDM, Adapt to SDM, Improve SDM, Enhance teamwork, Study SDM. Each of these recommendations applies to one or more of the themes and critical success factors identified in the presentations:

- The term *spiral development* is not well defined or understood. (Define SDM.)
- Spiral development can be sharply defined with *invariants* and *variants*. (Define SDM.)
- *Spiral development* and *evolutionary acquisition* are different, but related. (Define SDM, Adapt to SDM.)
- Spiral development differs between government organizations and commercial organizations. (Adapt to SDM.)
- Some spiral time cycles are still fairly long—two or three years—while others are much shorter—two to three months. Typically, longer cycles are found in government. (Improve SDM, Study SDM.)
- Some of the critical success factors for spiral development are
  - Risk must be managed. (Adapt to SDM, Educate about SDM.)

- The culture must be trusting. (Enhance teamwork.)
- Stakeholders must be involved. (Enhance teamwork.)
- The technology must be ready. (Adapt to SDM.)
- Requirements must be flexible. (Adapt to SDM.)

Participants were unanimous in the opinion that spiral development is a valuable tool for the development of systems. To help extend its value to more projects, all agreed to participate in, and encourage others to participate in, the final recommendation: Promote SDM.

3 Work Group Reports

The work groups were each assigned a specific topic:

Work Group 1. Spiral Development / Evolutionary Acquisition

Work Group 2. Integrating Software and Systems

Work Group 3. Changing Role of Requirements

Work Group 4. Institutional Challenges

Work Group 5. Process Issues

Each work group was asked to cover the following aspects of their topic:

- A Vision Of Successful Spiral Development Practice
- Problem And Challenges Facing SDM
- Critical Success Factors To Success Of SDM
- Recommended Actions

3.1 Work Group 1 - Spiral Development / Evolutionary Acquisition

Panel: David Carney, SEI (co-chair); Don Reifer, USC (co-chair, scribe); Jack Ferguson, OUSD; Jude Franklin, Litton/PRC; Kevin Goeke, TRW; Bruce Long. Lockheed-Martin; Rami Razouk, Aerospace; Skip Saunders, MITRE; Gary Thomas, Raytheon; Ann Willhite, MITRE.

This group considered the relationships between spiral development and evolutionary acquisition. Its aim was first to conceptually dissociate the two, then to consider the ways in which they reinforce each other, and finally discuss their joint potential for creating complex systems. As with all of the work groups, the original expectation of its outcome was greater understanding of the critical success factors for spiral development, and a list of recommendations that might help to realize those success factors.

3.1.1 Summary of Discussion

D. Carney proposed the following questions as an initial strawman for the WG discussions:

- In what ways does the general notion of acquisition (whether evolutionary or otherwise) overlap with any development process (whether spiral or otherwise)?
- In what ways does *evolutionary* acquisition overlap with a *spiral* development process?
- Are there success factors that are common to each?
- Are there any success factors for one that contradict success factors for the other?
- What role do government policies or directives (e.g., AF 63-123) play for both spiral development and evolutionary acquisition?

The WG decided immediately that, given the short amount of time available, the most useful approach would be to concentrate on evolutionary acquisition itself, especially since the other four groups would be concentrating on spiral development. Another decision was that rather than developing a list of Critical Success Factors for evolutionary acquisition, the WG could more usefully spend some time considering the variants and invariants for evolutionary acquisition, a parallel to Boehm's list of variants and invariants for spiral development.

A pervasive question was one of scope: was the WG considering evolutionary acquisition from the viewpoint of the DoD? the government? the entire industry? Although the WG discussed this question from various aspects, it was never quite decided. (The final briefing of the WG had a government/DoD bias, a point that was criticized in several comments from the plenary audience.)

The group sought to harmonize its several different notions of evolutionary acquisition by brainstorming about its goals, defining characteristics, and constraints. This was not an attempt at a canonical definition of evolutionary acquisition, but rather to quickly gain a shared understanding of its general concepts.

After discussion, the WG agreed that the goals of an evolutionary acquisition approach were to

- achieve a better, cheaper, faster acquisition process
- satisfy user needs and gain mission success
- identify and satisfy valid requirements
- demystify the acquisition process
- be nimble enough to take advantage of technology advances

The WG also noted that most of these goals were rather general, and not truly specific to evolutionary acquisition *per se* (for instance, “better, cheaper, faster” is the goal of almost every new initiative).

The defining characteristics of evolutionary acquisition suggested by the WG were less general and better focused:

- more COTS
- more licensing
- increased partnering and outsourcing
- decreased specifications and standards
- positive instead of negative rewards for risk-taking

In reviewing these characteristics, the WG noted that the intended decrease in standards was in “how-to” standards, and not necessarily in interface standards. This observation led to an important realization by the WG. The government has expressed often its growing desire to tell contractors **what** is wanted, and to avoid telling contractors **how** to do it. And yet, the current stress by the government on spiral development is actually a contradiction of this sentiment, since it does precisely that—it mandates the “how-to” that the contractor should follow.

The WG discussed the various things that can constrain evolutionary acquisition, of which the following were seen as most significant:

- interoperability
- the law and its interpretation
- time to delivery
- colors of money
- how testing is done (incremental vs. end product?)
- difficulty in starting a new program

- culture change
- lean workforces (resource constraints force more co-location)
- continuing change

Note that these constraints vary both in scope and perspective: some reflect the current, "as-is" state (e.g., "difficulty in starting a new program") and others reflect "to-be" constraints (e.g., "[the need for] lean workforces [when doing a true evolutionary acquisition]").

All of these items (goals, characteristics, constraints) became the raw material for the remainder of the WG's discussion. The statements of challenges, recommendations, and variants/invariants found below will be seen as reformulations of most of these items.

3.1.2 Challenges/Problems

Per the instructions to all of the Work Groups, WG1 defined a set of twelve issues that posed challenges; the group focused on those things that can inhibit the successful use of evolutionary acquisition. Note that some of these problems are indicated as particular to DoD or government.

Color of money. (mainly a challenge for the government): refers to the difficulty in gaining the flexibility needed by evolutionary acquisition, yet still following regulations and laws about funding allocations, using monies for a specific purpose, etc.

Lack of teamwork: refers to the need to reduce the sense of mistrust that often exists between an acquiring organization and a contractor

User challenges: refers to the reluctance, common within the user community, to accept incremental delivery of needed functionality

Logistics, training, etc. for increments: refers to the changes within the Acquisition community that an Evolutionary approach requires, such as training management personnel, costs related to incremental deliveries of systems, and so forth

Workforce skills and experience: refers to the current lack of a broad experience base with evolutionary acquisition (and also with using a spiral development process within EA). This lack of experience contributes to the currently risk-averse nature of the acquisition community noted below

Measurable milestones: refers to the difficulty of finding precise ways for the acquiring organization to define what will be in an increment, as well as to determine that the contractor has in fact produced that (partial version of) system

Consistent and effective risk-management processes: refers to the need, when an evolutionary acquisition approach is used, for improvements in the way that risk management is currently carried out

Staffing profiles/iterations: refers to the need for flexible staffing approaches to accommodate the changing needs of different iterations

Ability to contract quickly (mainly a challenge for the government): refers to the need, when using an evolutionary acquisition approach, for a more agile and nimble contracting process than currently exists in government acquisitions

Access to testbeds and laboratories: refers to the need for making capital investments in the infrastructure capabilities required to perform the iterative activities needed for evolutionary acquisition (and also spiral development)

Risk-averse nature of the acquisition community (mainly for government): refers to the current climate in government acquisition, in which risks are considered dangerous—and thus often suppressed by all parties—rather than seen as unavoidable, and in need of careful management during an acquisition

Ability to collect and analyze metrics: refers to the difficulty of determining which data are truly indicative of good (or bad) progress as a system is iteratively created and delivered

3.1.3 Recommendations

The WG made the following six recommendations. With the exception of the first and the sixth, the WG named a likely person or organization to act on the recommendation, and this person or organization is indicated as the agent for each recommendation.

1. **Define evolutionary acquisition and its relation to spiral development.**

   Define more clearly, perhaps in a commercial standard, what evolutionary acquisition is, and how it relates to spiral development. There is a particular need to clarify the roles associated with these processes. This issue is of major importance, and is significant at a level wider than DoD or even the federal government. Agent: possibly INCOSE/IEEE; the WG was reluctant to name a specific individual or organization as agent for this recommendation.

2. **Formulate incentives to adopt evolutionary acquisition practices.**

   Formulate and recommend incentives to change the behavior of stakeholders (program managers, users, etc.) to adopt evolutionary acquisition practices.

   The WG felt that a major need for the success of evolutionary acquisition was to reverse many aspects of the current acquisition climate. Today, there is every incentive to continue a program, whatever its condition, and no incentive (and, in fact, considerable career peril) to cancel a program. So while a goal of evolutionary acquisition is precisely to facilitate “go-nogo” decisions early in the life of a pro-

gram, there is no reward, and no reward infrastructure, for a manager to so do. The WG considered that a neutral party such as the SEI would be an ideal entity to begin this work, though it is likely that other organizations (one such is IDA) would be likely contributors. Agent: SEI, IDA.

3. **Streamline the contracting process.**

   Letting a contract takes far too long. These steps are needed:

   - The program initialization process must speed up.
   - Contracts must be chosen based on risk.
   - New starts should be made easier, and there should be no penalty for "fast failures."
   - The POM process should have "financial wedges" for executing more rapidly.

   The WG discussed whether it was appropriate for it to make recommendations to high-level persons and decided that only by making its opinions and views known can such high-level executives become aware of the feelings of the wider community. Agent: OUSD/AR, Mr. Soloway.

4. **Strengthen the IPPD policy to include ways to build teams and trust in EA.**

   Strengthen the IPPD policy and guidelines to include mechanisms to build teams and trust as part of the evolutionary acquisition process. Such mechanisms can include off-sites, "bootcamps" and so forth. Organizations must have the ability to periodically rebuild and refresh teams as staff turnovers occur. The WG discussed the difficulty and intangibility of culture change, climate, and similar questions. Still, there are steps that can be taken, and the WG felt that policies that encourage talking those steps should be created. Agent: USD/AT&L, Dr. Ganssler.

5. **Define and certify core competencies in evolutionary acquisition.**

   There currently exists the notion of certification of competency in the Acquisition community; this needs to be extended to those activities (i.e., those implied in the previous four recommendations) that are critical to evolutionary acquisition. Agent: DAU.

6. **Coordinate personnel changes with increment boundaries.**

   Given the centrality of multiple deliveries to the evolutionary approach, it would be of great benefit if those milestones were harmonized with personnel changes within a program. Agent: Unknown; some person(s) responsible for personnel issues within DoD.

3.1.4 Variants/Invariants

The final portion of the WG1 session was devoted to creating a list of variants and invariants for evolutionary acquisition The WG felt that some set of characteristics or attributes that were specific to evolutionary acquisition would provide a useful parallel to

the set that Boehm developed for spiral development. Table 7 shows the list that the WG developed.

Table 7: Variants and Invariants for Evolutionary Acquisition

Invariants	**Possible Variants**
Accommodating evolving requirements	Degrees of flexibility
	Degrees of time phasing
Multiple deployments to the field	Overlapping vs. non-overlapping increments
	O&M vs. RDT&E funds
Consideration in each cycle … PLUS technology insertion	Choice of development process
	Choice of contract type
	Nature of incentives
	Choice of risk reduction techniques
Emphasis on total lifecycle activities	Testing, training, supportability, etc.
Decision point at the end of each increment involving multiple stakeholders	Choice of decision-making method (e.g., risk-based, funding, etc.)
Content of deployment driven by risk (market place constraints, operational context, etc.)	Tradespace for deciding what to deploy
Managing stakeholder lifecycle commitments via anchor points	Management of teamwork issues

3.2 Work Group 2 - Integrating Software and Systems

Panel: Eileen Forrester, SEI (co-chair); John Foreman, SEI (co-chair); Chris Abts, USC; Elliot Axelband, USC; John Cosgrove, Cosgrove Computer Systems; Peter Hantos, Xerox; Joshua Hurvitz, Israel Aircraft Industries.

3.2.1 Summary of Discussion

The task of this group was to consider integrating software and systems during spiral development. We began with general discussion of issues participants wanted to bring to the table. In the course of that discussion, the group realized the need to adjust the scope or focus of our topic (more on this below). The desired outcome for the group was an improved understanding of challenges and success factors that prohibit or contribute to integration of systems with software that are developed using the spiral method. In addition, the group was to make recommendations that would contribute to successful use of spiral development, particularly in the area of integration. A number of the group's items are applicable to spiral development in general, rather than to integration of software and systems during spiral development.

3.2.2 Topics Covered

E. Forrester proposed these questions for the work group discussions:

- What does effective integration of software and systems during spiral development look like? Can we articulate a "to be" vision?
- What are the critical success factors we can identify for integrating software and systems when using spiral development?
- What are the key challenges and inhibitors to effective integration (such as cultural issues, skill gaps, financial constraints, collaboration failures, acquisition pressures)?
- What recommendations for action can we make to improve the chances of success for developers integrating software and systems during spiral development?

The group covered the above questions and also found itself returning often to these topics:

- business model and contracting
- systems engineering; inclusive or exclusive of software engineering and effect on integration during spiral development

We used a combination of group discussion and structured brainstorming to cover our questions.

3.2.3 Scope

After some discussion, we agreed that our topic was *integrating systems that include software*, rather than *integrating systems and software*. In other words, we realized that the task of integrating is one that deserves attention as a critical success factor for successful spiral development, and that treating software as a spoiled stepchild is not productive. We struggled in the course of the afternoon's work to avoid privileging either systems engineers or software engineers and their work products and to adequately account for the inclusion of other disciplines. Given the conflicts we noted amongst this group of seasoned professionals on this score, this is probably a topic for further discussion and action. In addition, of course, it is not software alone that needs to be integrated with the system. We returned often to the idea of integrating hardware and people. The domain of interest is systems that include software, but the task of integration goes beyond—and is improperly characterized as—integrating the software.

3.2.4 Vision

Our vision is that

- In 5-10 years, integrated systems (software, hardware, people, etc.) are created in a manner that satisfies the customer's requirements in a cost-effective, risk-sensitive way.
- The systems creation process
  - is change friendly
  - integrates the waterfall, spiral, evolutionary, and new-system views

We developed this statement to express our vision of successful spiral development. It is inclusive of integration and goes beyond it. Effective integration practice would contribute to bringing about this vision and some elements in this expression build on later statements of challenges and critical success factors. Note that the phrase we chose is "change friendly"; we wanted to go beyond the sense of accommodating change as a necessary evil to an attitude of embracing change. Integration is one of the critical ongoing processes and stress points where this attitude must be real. We wanted to convey that an integrating approach to systems could encompass the development approaches we are familiar with and allow for innovations we have not yet seen.

3.2.5 Problems and Challenges

We noted the following general problems:

1. Systems, software, and hardware disciplines often have differing or clashing processes. It would be beneficial for integration if they worked according to compatible processes—these need not be the same processes, but they must work together successfully. To enjoy compatibly working processes would require removing barriers between the disciplines in terms of assumptions, success and progress models, working vocabulary, and so forth.

2. System, software, and hardware people need to have a common language that allows them to recognize and understand the biases and points of view each group brings to the integration task. While this item is a corollary to item 1, it may provide sufficient leverage to be treated separately.
3. We noted several issues of contracting as related to integration. The top-level issue is the ability of the government-sector contracting communities to create contract vehicles that are appropriate to an evolutionary development approach. The lower-level aspects of this issue are that contractual success factors and progress indicators are necessary for each spiral cycle and billable delivery; however these indicators should be driven by the individual project business models. It is critical to have consensus about the basis on which one gets paid.

In the course of discussing Item 3, we noted a topic that goes far beyond our work group, and perhaps beyond the topic of the entire workshop. The government must provide systems contractors—who deal with risk, and who are enjoined by those of us promoting spiral development to tolerate and manage ever-greater risk—adequate financial return to ensure their long-term financial viability. That is, when the government tries to lower costs at every turn but also demands increased innovation and flexibility to absorb risk, a whole class of contractors finds their survival in question. If these contractors are lost, the government will lose its manufacturing base for critical systems. In turn, this leads to Item 4, which is also not specific to integration alone.

4. It is a key challenge to move from a risk-avoidance culture to a risk-acknowledging and risk-managing (perhaps even risk-embracing) culture that allows candid discussion of tradeoffs. Remember: without risk, there is no profit. If we ask contractors to tolerate, even embrace, more risk, and we allow them too little profit, what reward is in place for them to continue?

3.2.6 Critical Success Factors

Among the success factors for integration during spiral development are these:

1. Critical success factors derived by inversion of the list of problems and challenges. These are
    a. Various disciplines require compatible processes and common language as they do the work of integration.
    b. The government must create contract vehicles supportive of spiral development; it must negotiate and clearly express progress indicators and basis for payment as spirals conclude.
    c. The government must reward the risk taking that is integral to spiral development.
    d. When controlling costs, the government must consider the long-term health of the contractors who make up its manufacturing base.
    e. The ability to make informed tradeoffs while mitigating risks is a key feature of integration during spiral development.

In addition to these restatements of the challenges, we noted the following success factors related to the integration process:

2. The integration process must minimize the dichotomy in perspectives between product/process producers and customers/consumers; ideally they would come to a common view.

   The producers must be attuned to the big picture, roles and responsibilities, general requirements in operational context, and other system elements.

3. Consumers or customers must attend to business strategy, risk management, expectation management, and the tasks of remaining informed and educated. It is critical that the process architecture for integration of systems have
   a. short, manageable, minor loops (~ 3-6 months)
   b. proper focus on regression testing policy (re-validate functions implemented in prior cycle)
   c. test planning that is related to spiral
   d. balance between white-box and black-box testing

The issues of appropriate planning, practices, and policy for testing comprise one of the points that we found to be uniquely critical to integration.

In addition to a suitable process architecture, the cost effectiveness and general effectiveness of integration activities will contribute to successful use of spiral development. Cost effectiveness considerations must include the effort-to-accomplish, including schedule, labor, and reusability of processes.

Measures of success for the effectiveness of integration activities include

- verification of requirements; examination of current and prior functions during spiral iterations, checking for resolution of discrepancies
- validation of system behavior; response to anomalies, boundary – near and exceeded, and time domain (overload, endurance, etc.)
- emergent properties

Successful integration practice must be designed for all of these.

The group did not have time to discuss which among these issues are truly critical and which are less so. Nor did we debate whether any of these factors, or perhaps some details of them, are invariants of spiral development in Boehm's sense.

3.2.7 Recommendations

In our recommendations, we attempted to synthesize those few actions that would allow us to capitalize on success factors and overcome problems or challenges. In all cases, suggested agents were our first guesses—these require further consideration.

1. **Pay greater attention to education and selection of integration practitioners.**

   Integration during spiral development should be specifically taught during training for systems integrators. Aptitudes and skills for integrators should be enumerated

and integrators selected based on demonstrated skill. Agents: INCOSE, DAU, DSMC, corporations with training programs, and universities.

2. **Deliberately build an SDM community.**

   Support forums and workshops to develop and promote mutual understanding among practitioners of all involved systems disciplines. Agents: CSE, SEI, INCOSE, etc.

3. **Enhance program management to support effective integration in spiral developments.**

   a. Promote better linking, understanding of purpose, and interaction of R&D organizations to support technology evaluation and product development.

   b. Implement IPTs effectively; where they are implemented well, they appear to support effective integration (and many other elements of good practice supportive of spiral development).

   c. Develop contracting methods and business models compatible with evolving products and systems. Include the ability to measure progress and billing milestones.

   Agents: DAU, DSMC, SEI, PMI (Program Management Institute).

4. **Adopt these testing practices.**

   a. Test for success, anomalies, time-based emergent behavior, and architectural integrity.
   b. Evaluate systems integration testing results for adequacy and worst-case behavior that may affect mission success and safety.

Agents: DAU, DSMC, AFOTEC, INCOSE.

3.2.8 Conclusion

As we evaluated this work group, we felt that we made progress toward understanding issues of common concern. However, we felt we were an insufficiently diverse group. We had very few users, acquisition folks, or uniformed practitioners. We were notably lacking in representatives from companies involved in system development as opposed to software development. To continue the "dating and marriage" analogy used throughout the workshop, we are concerned that our deliberations may be too much like a marriage between first cousins.

3.3 Work Group 3 - Changing Role of Requirements

Panel: Pat Place, SEI (co-chair); Paul Grünbacher, USC (co-chair); Verna Griffin, Lockheed Martin; Judy Kerner, Aerospace Corporation; David Klappholz, Stevens Institute of Technology; Julie Kwan, USC; Jan Putman, MITRE; Walker Royce, Rational; Lt. Col. Bob Wind, AC2ISRC.

This breakout group considered the role of requirements in the context of spiral development. The group's mission was to understand what differences in both the development and use of requirements arise due to the nature of spiral development. The waterfall model, where requirements are developed during one phase of the life cycle and then used as a "contract" for development, was used as a baseline of traditional thinking within the DoD. This baseline was contrasted with the spiral model where there is an expectation that requirements will change during each spiral.

3.3.1 Summary of Discussion

The following questions were proposed as initial foci for discussion:

- Is it possible to delay development of requirements to later spirals? If so, how many of the requirements are needed initially?
- If requirement development is to be spread over a number of cycles, how can consistency of inputs be maintained?
- When, if ever, should requirements be considered complete?
- Given that requirements are expected to be amended in each cycle of the spiral, how can they be used for either contracts or the development of tests?
- Given the most desirable (but realistic) future state, how do we get there?

The first issue discussed by the work group was the nature of requirements. There was considerable discussion as to what is and what is not a requirement. This discussion centered on the issue of whether or not non-functional requirements (for example, the programming language to be used) really were requirements or should be considered "constraints," which we consider as being more flexible than requirements. The final solution to the issue of terminology was to adopt the MBASE terminology where the word requirement is preceded by an appropriate adjective – "property," "project," "process," "functional" and so on.

The discussion on the nature of requirements was independent of spiral or any other development model and was, perhaps, typical of a number of discussions. These arose because of the WG's overall feeling that, in general, current requirements engineering approaches are profoundly dissatisfying, leading to poor systems acquisitions. Other general issues discussed were

- The right people must set the requirements. Too often the users aren't directly involved in requirements setting but are represented by, perhaps, a functional manager who may not understand the actual needs as well as real users.

- Too many people are involved in requirements development, over-shadowing the needs of the actual users. This often leads to operational requirements not being specified in sufficient detail.
- Requirements, when too detailed, prevent solutions; the requirement for a pop-up alert on a display precludes the use of an audible alarm.

One important vision of the future developed. In this vision, requirements would, initially, be abstract with minimal detail. Each cycle of the spiral development would refine the requirements, concentrating on adding levels of detail rather than adding new functionality. One problem with this approach is that, initially, it is difficult (or impossible) to know how to award the contract without getting a protest from losing bidders. One suggested solution to this problem was, in the initial stages of development, to award the contract to multiple bidders and use their performance through the early (and quick) spirals as a mechanism for down-selecting among the contractors.

The spiral model centers on the management of risks. Risks arise when decisions are made based on incomplete information. One of the greatest uncertainties in traditional development is the requirements; either their meaning, or flexibility, or certainty of completeness. Spiral development brings with it a different management paradigm. Specifically, spiral development is about the management of unknowns. Attendant with this management is the realization that requirements will change as new information becomes available, regardless of whether that new information is a change in the nature of the needs for the system or in users' understanding that alternative solutions are possible.

Throughout the discussions, there was an implicit (and at times explicit) sense that the biggest barrier to spiral development of systems and requirements is the current state of the contractual relationship. The DoD and contractors too often see each other as adversaries rather than collaborators leading to the requirements as being the arbiter of conflict. When such a mindset prevails, requirements must be developed up front (so that conflict can be arbitrated). This up-front development is the source of current dissatisfaction and is antithetical to the ideals of spiral development as discussed in the vision.

Spiral development, we believe, is a generator of change, particularly in terms of the nature of the contractual relationship. For spiral development to be a success, there will be a need for an evolutionary, iterative and, most important, shared view of the requirements. In such a view, requirements will be developed cooperatively rather than being a specification from the government to the contractor. However, such an approach will require

- automated environments to facilitate collaboration, not more meetings
- rigorous notations for capturing differing views of requirements
- acceptance by acquirers of the limitations listed by the contractors

3.3.2 Challenges/Problems

One of the areas each work group was asked to address was that of challenges and problems to be faced. WG3 developed the following ten major challenges in response. Each of these challenges is coupled with suggested solutions. While these solutions may not be fully actionable recommendations, they are steps toward such recommendations.

1. Terminology for requirements continues to be a problem. As seen in our work group, there was a distinct sense that “my requirements are your constraints,” with the consequence that such requirements may be given less attention.

   Rather than try to specify what is and what is not a requirement, classifying the requirements into appropriate categories can bring agreement. The MBASE classification scheme is sufficiently general to be used successfully.

2. There is a distinct tendency to try to develop too many of the requirements too quickly. The DoD interpretation of fairness means that every contractor is provided with the same data to be used in bids. This data includes the complete set of requirements for a system. However, this means that the DoD develops the requirements without the benefit of contractor input and also requires that too much detail to be provided too soon in the life of a program.

   If spiral development and evolutionary acquisition can be institutionalized (and internalized) by both the DoD and the contracting community, requirements can be initially abstract and can also evolve over time with more detail being added as necessary. Note, this isn't a complete solution; still remaining is the problem of determining the minimum set of requirements to start out with.

3. Communicating requirements is a continual problem; far too often requirements are ambiguous and open to misinterpretation. Although rigorous languages for expressing requirements exist, these languages are rarely, if ever, applied in the initial stages of development. A number of approaches can be adopted.

   It would be possible to teach more rigorous notations in universities and then wait until there is a critical mass of people who understand them. This is clearly a weak strategy, in that by the time such critical mass develops, notations will have changed and the users will not be familiar with the new notations. Also, it's a passive strategy and the communication problem requires more urgent action.

   The alternative is to be more proactive and teach some set of appropriate notations at the military colleges – furthermore, encourage the use of such notations with incentives to program offices using them.

4. Miscommunication often occurs due to the stakeholders’ failure to understand the scope of a program. This leads to functional requirements being developed that are not directly relevant to the particular program. In the extreme case, functional requirements for system A may be project requirements for system B and should not be presented as functional requirements to the developers of system B.

   Educating stakeholders in the nature of requirements will help solve this problem. In particular, stakeholders need to understand that the classification of a requirement is context sensitive and may also change for different levels of abstraction (for example in different spirals).

5. Coupled with the communication problem is the difficulty of managing expectations. Current practice is that once a requirement has been created, and expressed

as part of the contract, it must be satisfied; the DoD must reject the contractor's delivery if the requirement isn't satisfied (obviously, this is the extreme view and contractual modifications can always remove a requirement). Users (or their representatives) have learned that they must specify every detail and that each of those details must then be delivered.

In a world of spiral development and acquisition, all of the stakeholders, users, acquirers, and developers must manage their expectations. Acquirers must understand that satisfaction of a requirement may be postponed for later spirals. Users need to understand that there is need for flexibility in requirements. Finally developers must be flexible in adjusting to the changing needs and desires of the users and acquirers.

6. Identifying risks is one of the hardest challenges for programs. Furthermore, the risks may not be functional or technical in nature. A likely risk, for any program, is cancellation –if certain, minimal, functionality isn't provided early enough.

   This means that prioritizing requirements is important, but that the prioritization should be based not only on technical risk, but also on political reality. However, prioritizing requirements, which is a natural part of spiral development will lead to

   - the most important capabilities being fielded first
   - sub-components being available (perhaps) for use earlier than planned
   - the least important requirements being dropped if program money runs out due to either cuts in funding or early cost overruns

7. Changing requirements are a fact of life and cannot be avoided. Any program that takes significant time to develop is liable to having user needs change either due to the changing environment or increased understanding of system capabilities. The effects of changes in requirements are far-reaching: the entire system, including the architecture, has to be reconsidered. It is vital, in such case, to perform an analysis of the effect of any given change. In order to do this, though, system models must be maintained. In a typical waterfall development, once a model has satisfied its purpose it is no longer maintained. For example, a high-level design is rarely changed if the program is in the implementation phase even if a changed requirement would, logically need a change in the high-level design documentation.

   Spiral development, because of its cyclic nature, accommodates changing requirements. Furthermore, because future cycles may need to modify decisions made during earlier cycles, there is a greater likelihood (unfortunately, though, not a certainty) that the models will be appropriately maintained.

8. When requirements are created it is all too often the case that some requirements are missed, simply because they are "obvious" or a case of "everyone knows." This results in requirements documents containing implicit assumptions about the operation of the system with the final consequence that conflicts between different requirements (implicitly or explicitly stated) may be missed.

   Under a spiral model of development, there is expectation for the addition of new requirements or the refinement of existing requirements. This expectation means that there is an opportunity for omitted requirements to be added as part of the development process without need for significant contractual re-negotiation.

9. One of the biggest challenges is, frequently, that system acquisition is combative rather than collaborative. The DoD and the contractors have an "us and them" mentality with the contract acting as the fence keeping the parties apart. This often

leads to an unwillingness of either side to be flexible with respect to any of the requirements. The ultimate consequence is that users don't get the systems they need.

The introduction of a spiral development and evolutionary acquisition process is as much a cultural as it is a technical change. Both parties expect changes at the start of each spiral. Although the extent of the change may still be an area of conflict, agreement to make changes is a step forwards.

10. Requirements, which are an important aspect of the contractual relationship and the basis for system development (and also tests and documentation) are frequently developed by the wrong group of people. Users are omitted, being represented instead by functional managers who know less about day-to-day needs. Contractors (who have implementation experience) are omitted due to fairness concerns. The result is that some key stakeholders in the system are omitted from its definition. Failure to bring together a suitable requirements team leads to many of the problems outlined above.

    Spiral development, as a cultural change, admits the possibility that the government and contractors can work together as a team, even as early as requirements definition. This is particularly true when the culture has changed sufficiently that the RFP may contain little more than a statement of operation or a mission needs statement. With an expectation that the contractor and the government will cooperatively develop requirements, refining them as appropriate, during the spiral development of the system.

3.3.3 Critical Success Factor

The key to successful system development is collaboration between the DoD and the contractors. If this can be achieved, through any mechanism, DoD will be able to acquire systems more successfully. Anecdotal evidence suggests that when collaboration has occurred systems have been developed in an economical and timely fashion. The group viewed spiral development as requiring a shift in mindsets to a state where collaboration and cooperation is the norm rather than the exception.

3.3.4 Recommendations

As stated in the preceding section, the WG generated some solutions to the problems and challenges. In some cases, these solutions approach a recommendation. However, the WG felt that the time allowed was insufficient to develop well-reasoned recommendations other than the following:

Publish success stories. There have been cases where programs have been delivered using spiral development and evolutionary acquisition processes. However, these cases are not widely known in the community. Maintaining a collection of spiral development stories (the good and the bad) will help program managers and contractors form convincing business arguments for taking the risk of adopting spiral development as a means to delivering systems. Agent: CSE is the obvious candidate for maintaining this data.

3.3.5 Conclusion

Current requirements engineering practices are clearly unsatisfactory. The introduction of spiral development, with the associated cultural change, is both an enabler and an opportunity provider for the DoD and the contractors to collaboratively agree on changes to the requirements development process.

3.4 Work Group 4 - Institutional Challenges

Panel: Caroline Graettinger, SEI (co-chair, scribe); Wilfred J. Hansen, SEI (co-chair); Barry Boehm, USC, CSE Director; Tom Bostelaar, TRW, CCPL Project Manager; Winsor Brown, USC, Asst. Director CSE; Steve Cross, SEI, Director; Ross Dudley, Maj. USAF, AC2ISRC; Larry McKee, MITRE, AC2ISRC.

This work group considered "institutional challenges" to the Spiral Development Model. Since the spiral development approach suggests specific policy and process guidelines, it clearly offers challenges to an institution's existing policies, process, and culture. At the same time, this existing environment poses many challenges to the adoption of the spiral approach.

Through initial discussions, the group defined "institutional challenges" to include:

1. challenges from institutions on the adoption or use of spiral development
2. collective constraints on building a successful spiral development team

These two sets of challenges are somewhat different. The first refers to challenges in policy, process and culture that can prevent an organization from a successful and sustained adoption of a spiral development method or from achieving its full benefit. The second refers to the challenges that may prevent the spiral team from being fully functional and achieving its goals.

3.4.1 Summary of Discussion

Overall, this work group identified and classified institutional challenges and then discussed how the challenges could be met. A pervasive question was one of scope, i.e., was spiral development being considered from the viewpoint of the DoD, the government, or the entire industry? Since most of the participants had a DoD perspective, this was reflected in the discussions and in the final outcome.

The group began by attempting to quickly gain a shared understanding of the general concepts of spiral development. After discussion, the WG agreed that a shared definition of spiral development includes

- short spiral cycles
- anchor point milestones: LCO, LCA, IOC
- risk-based decision making
- model generation based on project conditions
- a collaborative team of users, developers, acquirers, and test/evaluation people, i.e., an integrated product team (IPT)

This list was not intended to be all-inclusive, but to include key characteristics that we could address in our discussion of institutional challenges.

The group discussed a vision of software engineering that included spiral development and arrived at the following "vision" for a future, where the Department of Defense and the government in general are

1. at Level 5 of the "SMM" (thereby suggesting a future "Spiral Maturity Model")
2. as effective as the best commercial practices in using spiral models to manage their investment portfolio of software intensive systems

In (1) we intended primarily to liven the final presentation by mimicking the term Capability Maturity Model. Nonetheless, we thought it may be possible to define multiple levels of spiral development use, each more effective than the previous. Item (b) reveals our bias toward viewing the problems of the government. Indeed, about half our group was involved in the Joint Aerospace Applications investment category for the Air Force's AC2ISRC (Aerospace Command and Control, Intelligence, Reconnaissance, and Surveillance Center). In (2) we deliberately refer to "investment portfolio of software intensive systems" instead of "spiral development." The team feared that "development" is but a small portion of the software lifecycle and that spiral methods should be applied to all phases of the project from concept to sustainment.

The above discussions became the foundation for the remainder of the WG's discussion. The results are statements of necessary changes, challenges, and recommendations, which are presented below.

3.4.2 Challenges/Problems

Per the instructions to all of the Work Groups, WG4 defined a set of issues that posed challenges, focussing on those things that can inhibit the successful adoption and sustained use of spiral development. Brainstorming produced thirty-three challenges, which we were able to organize into five categories. Some of these, as noted, are more pertinent to the government and DoD rather than industry. The five categories are below.

1. ***Poor understanding of spiral development*** refers to the lack of a common understanding across the software engineering community on the definition and implementation of spiral development.
2. ***Rigid funding cycles and contracting policies*** (mostly for government) refers to the general inflexibility of funding and contracting policies, and which are not well suited to the special needs associated with spiral development.
3. ***Existing cultures, policies and practices*** refers to the challenges associated with getting an organization to adopt any new technology or process that requires associated changes in the ways of doing its day-to-day work.
4. ***Organizational and individual risk aversion*** refers to the barriers some organizations have that discourage or otherwise prevent individuals and groups from raising risks to a visible level, where they can be managed.

5. ***All the usual teaming challenges*** refers to the challenges associated with creating a trusting, team environment from multi-disciplinary members, who are often geographically distributed. The issue of creating continuity of leadership, sponsorship and membership is particularly relevant to DoD where two-to-three-year tours of duty in one location are common.

3.4.3 Changes Necessary to Address the Challenges

After identifying the challenges the group asked what changes are necessary in order for these challenges to diminish or vanish. This table depicts our answers.

Challenges	Necessary Changes
Poor understanding of spiral development	• Grass roots initiated definition of spiral model • Common understanding of the benefits • Get community to internalize the definition
Rigid Funding Cycles and Contracting Policies *"A vision without funding is a hallucination"*	• Introduce flexible contracting strategies • Include spiral planning and funding in ASR • Introduce out of cycle funding methods • Understand TSPR* and its relation to spiral
Existing cultures, policies and practices	• Modify or replace cultures, policies and practices • Leadership buy-in
Organizational and individual risk aversion *a.k.a. "Shoot the Messenger" syndrome*	• Adopt view that risk management is necessary • Promulgate team risk management • Remove organizational barriers to identifying risks
All the usual teaming challenges	• Eliminating "us vs. them" attitudes • Multiple chains of command for reporting (home org and IPT) • Means to accommodate geographical distribution • Continuity of leader-/sponsor-/member-ship
* Total System Performance Responsibility	

3.4.4 Critical Success Factors

Some factors are more important than others in the adoption and successful use of spiral methods. We identified the following as critical to the success of resolving the challenges described above:

Challenges	Critical Success Factors
Poor understanding of spiral development	• Definition must be clear and succinct (invariants and variants) • Must have experimental validation
Rigid Funding Cycles and Contracting Policies *"A vision without funding is a hallucination"*	• Make the acquisition people part of the team • Participate with Office of the Secretary of Defense (OSD) on policy planning
Existing cultures, policies and practices	• Understand what cultures, policies and practices enable or prohibit spiral • Return of Investment (ROI) data, compelling business case analysis, success stories
Organizational and individual risk aversion *a.k.a. "Shoot the Messenger" syndrome*	• Incentives for identifying and managing risks • Structured risk-based approach for project assessment • Trusting cross-institution environment
All the usual teaming challenges	• Consistent/persistent application of IPT principles and practices • Knowledge capture and organizational learning • Distributed performance appraisal methods

3.4.5 Recommendations

Based on our understanding of the challenges facing the spiral model and the factors critical to success, we made recommendations for further work at three time frames. For each, the "Agent" list indicates our best guess as to which parties can and should carry out these tasks.

Short-Term Recommendations

1. Develop and evangelize the definition of the spiral model. Agents: future workshops, all spiral model promulgators
2. Develop a range of training courses. Agents: SEI, DSMC, consultants

3. Document commercial case studies of the use of the spiral model. Agents: SEI, CSE, universities
4. Brief DoD leadership on workshop results. Agents: McNutt-Secretary of the Air-force Acquisition (SAF/AQ), Jack Ferguson-OSD
5. Submit a spiral pilot project through the Warfighter Rapid Acquisition Process (WRAP) Agents: AC2ISRC
6. Iterate the ESC handbook to reflect workshop results. Agents: MITRE/SEI/CSE work with ESC, Aeronautical Systems Center (ASC), Air Force Space and Missile Systems Center (SMC)
7. Study and disseminate the business case for using spiral. Agents: SEI, CSE, universities
8. Develop a focused risk section in ESC handbook. Agents: ESC/SMC/ASC
9. Teach teams how to identify and select incentives. Agents: Barry Boehm, consultants
10. Provide incentives for identifying risks. Agents: all
11. Create Project Management Institute (PMI) connection to spiral. Agents: SEI
12. As a pilot project, do a distributed team task w/ USC web-casting. Agents: CSE, ESC
13. Plan for more face-to-face at project onset. Agents: all
14. IPT training. Agents: consultants
15. Recognize team building as a risk in each iteration. Agents: ESC/ASC/U.S. Army Communications-Electronics Command (CECOM)/Space and Naval Warfare Systems Command (SPAWAR)/SMC
16. Study ROI of distributed vs. co-located. Agents: SEI, CSE, Universities
17. Do technology transfer of knowledge mgmt methods/tools. Agents: MITRE
18. Do technology transfer of distributed performance appraisal methods. Agents: MITRE

Long-Term Recommendations

1. Incorporate spiral development in university courses; both theoretical courses and practicums. Agents: universities
2. Write books. Agents: Barry Boehm, university faculty
3. Create supporting tools. Agents: consultants
4. Work with Congress to improve the funding model for DoD projects. Agents: Office of the Secretary of Defense
5. Develop a guide to the spiral model for military use. Agents: DoD, DCMC
6. Document governmental and military case studies of the use of the spiral model. Agents: Office of Secretary of Defense
7. Develop a structured, risk-based approach for project assessment. Agents: ESC/SMC/ASC/SPAWAR/CECOM
8. Develop better ways of doing distributed teaming. Agents: CSE, SEI, universities
9. Develop better collaborative tools. Agents: CSE, SEI, universities, vendors

Research Recommendations

1. Develop experimental validation of spiral method. Agents: CSE, SEI, universities.
2. Gather ROI data. Agents: CSE, SEI, universities
3. Understand culture, policy and practice implications for spiral success. Agents: CSE, SEI, universities
4. Understand human factors for successful distributed teaming. Agents: CSE, SEI, universities
5. Simulate/emulate distributed teaming in education environment. Agents: CSE, SEI, universities

3.5 Work Group 5 - Process Issues

Panel: Bill Peterson, SEI (co-chair); Dan Port (co-chair, scribe), CSE; Larry Bernstein, Have Laptop, Will Travel; Hal Hart, TRW; Tony Jordano, SAIC; Alex Lubashevsky, Lucent; Steven J. Lucks, ACSC (AAA); George O'Mary, Boeing

This work group considered the process issues, as well as associated issues, of spiral development. The group has members with varied backgrounds and special interests, including education/training, models, "change" management, systems and software engineering interactions, higher maturity level issues, quantitative/metrics approaches, and software estimation, as well as general process issues and solutions. A brainstorming approach was used to initially identify a mixture of questions, problems, and challenges. This was followed by a structured refining and recording approach to capture the outputs of vision, critical success factors, and recommendations of the group.

3.5.1 Vision

Our vision for spiral development is

- The Spiral Development Model (SDM) takes a seat at the table as a first class candidate lifecycle model for development projects. It is well understood, and understood in a common way, in both federal and commercial arenas. That is, all agree on when to use or not use SDM, the value of SDM, and its risks and tradeoffs.
- Projects using SDM are delivered on schedule, on budget, with high quality to satisfied customers. The business value of SDM is demonstrated through collected and analyzed data.
- SDM is acculturated within 10 years and becomes transparent best practice.

3.5.2 Challenges/Problems

Work Group 5 identified and prioritized the following nine areas of issues that pose challenges and problems to the success of spiral development from a Process perspective:

1. *Crisp definition.* There is not yet a crisp definition of spiral development based on the invariants. One is needed. There are questions to be answered about
    a. How detailed and fixed must be the lifecycle architecture (LCA) vs. project flexibility/evolvability?
    b. How to arrive at "core requirements," high priority risks, etc.?

    Also needed is an interpretation of "risk-driven" which differentiates spiral development from other process models. (For an example of choosing a process based on criteria, see Figure 3.) A further need is a set of examples of good and not-so-good attempts at implementation of SDM.

2. *Relationship with acquisition* refers to questions about the use of spiral development in a DoD Fixed Price Contract and/or including distributed IPTs.
    a. Is this feasible and what would a Fixed Price Contract have to say at Contract Award Time to make it feasible?
    b. It also refers to possible contention between LCA (and Life Cycle Objectives (LCO)) from an acquirer's perspective vs. a contractor's perspective.
    c. How can contractors be included more fully in pre-procurement to enable use of spiral development?
    d. How can the contractor effectively use spiral development if LCA is fixed?
3. *Value of spiral development.* There is a need to determine and make known, based on data, the value of the SDM and under what conditions it is most effective. This should be based on business objectives, not process objectives, should validate or refute the 0-40% productivity improvement claim, and answer the question: "Why would I want to use it?"
4. *CMMI, Spiral Development Model, and MBASE* refers to the fit of Spiral/MBASE concepts with the Capability Maturity Model Integration (CMMI) and all of its Process Areas. Specific example process areas where understanding and exploiting the fit would be useful are Risk Management, Requirements Management, the other Customer-related process areas, the Engineering process areas, and the new Measurement and Analysis process area. It would also be useful to understand and exploit the fit with the other CMMI process areas.

    Among the questions suggested for discussion are
    a. Are there special interpretations needed in order for spiral development to fit into a CMMI implementation?
    b. What are the fit/roles of Life Cycle Anchors vs. CMM compliance measures/imperatives?
    c. Can MBASE serve a role in transitioning organizations from SW-CMM and/or EIA/IS 731 to CMMI? In adopting IPTs/IPPD in CMMI?
5. *Transition and maturity* refers to the relationship and issues of spiral development and software process maturity. Among the questions are:
    a. Is Maturity Level 2 or other criteria a suggested pre-requisite for installing and using the Spiral Development Model?
    b. Should a project transition from a waterfall lifecycle to a spiral lifecycle without a mature process/team in place?
6. *Integrating Risk Management* refers to several questions raised about the details of implementing risk management as part of spiral development:
    a. What is the definition/process of risk management?
    b. What is the scope of risk management? One increment? Involve all stakeholders?
    c. When is it appropriate to start the risk assessment?
    d. How do you distinguish between low and high risks at different stages of development?
7. *Education and training* refers to identifying the key issues of the SDM that need to be covered in a training program and how to teach the process. These issues extend beyond education and training, into how to acculturate processes, including

through experience, use of measurement, and attention to people vs. process/technology. Finally, there are the challenges of transitioning traditional "programmers" to team/IPT members and overcoming resistance to change from those who already "know better" than to use SDM.

8. *Field guidance* refers to the need for documented guidance to support, and provide details of, the implementation of spiral development in actual practice, with some example questions:
    a. How do you schedule spiral lifecycles?
    b. How big can an increment be?
    c. What are the entry/exit criteria?
    d. How are contingencies handled (partially risk mitigations)?
9. *Humanistic approaches* refers to several issues raised about the people aspects of spiral development:
    a. Enable people to surface risks without punishment.
    b. Acculturate the Spiral process and make it "the way we naturally do things." A technology is mature when it "disappears."
    c. Prohibit requiring excessive documentation.
    d. Fit with/tailor to management structure.
    e. Identify indicators and alerts.
    f. Set appropriate expectations of customer and development team.
    g. Get the right 80% (e.g., 80/20 vs. 20/80).

3.5.3 Critical Success Factors

The group prioritized the following statements as critical success factors for spiral development:

- Acquisition process and contracting mechanisms are adjusted to facilitate spiral development methods.
- Organizational culture is receptive to SDM; this includes open performance of risk identification and risk management.
- A crisp, common definition of SDM is accepted.
- Training and education are available.
- SDM is refined so that it is scalable and tailorable to a wide spectrum of projects.
- The value of the SDM is widely accepted.
- The Spiral process is accommodated by cost and effort estimating methods.
- All participants actively participate in enacting the SDM.
- Expectations of customers and developers are realistic and harmonized.
- SDM is taught in all software engineering and business school curriculum.
- Some significant percentage of CSE affiliates are using SDM "correctly."

3.5.4 Recommendations

The group made the following six recommendations:

1. **Publish the definition of SDM.**

   A single paper defining, crisply, the spiral model in terms of the spiral invariants is essential to the success of any other recommendation or action. This paper should include the value of using spiral development, as well as the risks and tradeoffs of using it. There are other areas to be covered that may or may not fit within the same paper, so additional articles may be needed, but they should relate to each other and, perhaps, form a series or book. The areas most needing to be addressed, after the crisp definition, are:

   - Transition from waterfall/incremental to spiral and from spiral to waterfall/incremental
   - Process selection criteria, with risks, tradeoffs, and values
   - The relationships between Spiral, MBASE, CMMI, etc.

   Agent: Barry Boehm

2. **Convene a DoD/industry IPT to address acquisition process and contracting mechanisms.**

   Although it is recommended that processes not be put on contracts, there is the need for addressing whether and how a contractor can employ the Spiral Process within the current acquisition process. What disincentives for using spiral development are there and what should be done in contracting to encourage use of spiral development? Agent: OSD

3. **Collect data and experience and address the scale issue.**

   Continuing on the workshop presentations and potential expanding use of spiral development, a proactive attempt to gather data and field experiences is needed. This data can then be used in assessing the success and problems of spiral development and the value of using it. The successes and value must be publicized. The problems must be addressed. One concern the work group has is in the scalability of SDM, as well as data on experiences that validate and modify it based on larger scale usage. Agent: CSE

4. **Write a "field guide" to enacting SDM.**

   More documentation, based on experiences to date, is needed for the "how to" of SDM. Users of SDM need a documented set of practices to draw from in order to successfully transition SDM ideas into practice. Agent: CSE

5. **CMMI consider spiral model as acceptable alternative practice.**

   The Spiral Development Model should be investigated for its fit as an accepted best practice for satisfying CMMI Process Areas. A detailed mapping of SDM to CMMI should be prepared. A similar mapping of the incremental waterfall process to CMMI should also be prepared. Another useful mapping, and subsequent "gap analysis," for implementers of SDM (whether for CMMI purposes or not) would be from incremental waterfall to spiral. Agents: CSE and CMMI

6. **Elaborate spiral invariants for stakeholder tasks to explicitly require "collective decision making."**

This is a minor effort compared to the preceding recommendations, but needed for clarity in SDM. The spiral invariants should be elaborated to clearly communicate that “collective decision making” is a part of the model. Individual or independent sub-group decisions are not the expected set of stakeholders for such decision making. Agent: CSE

References

[Blanchard 98] Blanchard, B. & Fabrycky, W. *Systems Engineering and Analysis* (3rd ed.). Englewood Cliffs, NJ: Prentice Hall, 1998.

[Boehm 88] Boehm, B. "A Spiral Model of Software Development and Enhancement." *Computer* (May 1988): 61-72.

[Boehm 89a] Boehm, B. *Software Risk Management*. Los Alamitos, CA: IEEE Computer Society Press, 1989.

[Boehm 89b] Boehm, B. & Ross, R. "Theory W Software Project Management: Principles and Examples." *IEEE Trans. Software Engr.* (Jul. 1989).

[Boehm 94] Boehm, B. & Bose, P. "A Collaborative Spiral Software Process Model Based on Theory W." *Proceedings, ICSP* Reston, VA: IEEE, Oct. 1994.

[Boehm 97] Boehm, B.; Egyed, A.; Kwan, J. & Madachy, R. "Developing Multimedia Applications with the WinWin Spiral Model." *Proceedings, ESEC/FSE 97* New York, NY: Springer Verlag, 1997.

[Boehm 98] Boehm, B.; Egyed, A.; Kwan, J.; Port, D.; Shah, A. & Madachy, R."Using the WinWin Spiral Model: A Case Study." *IEEE Computer* (Jul. 1998): 33-44.

[Boehm 99a] Boehm, B. & Port, D. "When Models Collide: Lessons from Software Systems Analysis." *IEEE IT Professional* (Jan./Feb. 1999): 49-56.

[Boehm 99b] Boehm, B. & Port, D. "Escaping the Software Tar Pit: Model Clashes and How to Avoid Them." *ACM Software Engineering Notes* (Jan. 1999): 36-48.

[Boehm 00] Boehm, B. *Spiral Development: Experience, Principles, and Refinements* (CMU/SEI-00-SR-008) Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. http://www.sei.cmu.edu/cbs/spiral2000/boehmSR.html

[Carr 93] Carr, M.; Konda, S.; Monarch, I.; Ulrich, F. & Walker, C. *Taxonomy-Based Risk Identification.* (CMU/SEI-93-TR-06 ADA266992). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993. http://www.sei.cmu.edu/ publications/documents/93.reports/93.tr.006.html

[Charette 89] Charette, R. *Software Engineering Risk Analysis and Management*. New York, NY: McGraw Hill, 1989.

[Cusumano 95] Cusumano, M. & Selby, R. *Microsoft Secrets*. New York, NY: Free Press, 1995

[Ganlser, 2000] Gansler, J. S., The Honorable, Under Secretary Of Defense, Acquisition, Technology And Logistics. *Opening Statement.* Before House Armed Services Committee, Military Research and Development Subcommittee, Defense-Wide Research and Development, March 1, 2000.

[GAO, 1999] Government Accounting Office, *BEST PRACTICES: Better Management of Technology Development Can Improve Weapon System Outcomes.* Report to the Chairman and Ranking Minority Member, Subcommittee on Readiness and Management Support, Committee on Armed Services, U.S. Senate, July, 1999.

[Royce 98] Royce, W. *Software Project Management: A Unified Framework.* Reading, MA: Addison Wesley, 1998.

[Thorp 98] Thorp, J., DMR's Center for Strategic Leadership, *The Information Paradox.* New York, NY: McGraw Hill, 1998.

Appendix A Recommendations to Agents

The table beginning on the next page lists all recommendations made by the work groups sorted according to the "agent type" of their first specified agent. The agent types are:

Universities - The academic community, especially CSE and SEI

Consultants - Organizations who provide advice on development methods

Vendors - Those selling tools to assist in software development

Acquisition - Those units in the armed services responsible for acquiring systems that include software. In particular, AC2ISRC, AFOTEC, ASC, CECOM, DCMC, ESC, SMC, SPAWAR

Training - Organizations that provide professional training in software development, including DAU, DSMC, INCOSE, Program Management Institute

OSD - Policy personnel in the Office of the Secretary of Defense and associated offices.

In the table, the Agents, ID, and Title are as specified by the work group. The Action Class field is one of the classes listed in Section 2.4.

Recommendations Sorted by Agent Type

Action Class	Agent Type	Agents	ID Title
Improve	all	all	WG4:S10. Provide incentives for identifying risks
Educate	Acquisition	MITRE, SEI, CSE work with ESC, ASC, SMC	WG4:S6. Iterate the ESC handbook to reflect workshop results
Educate	Acquisition	ESC, SMC, ASC	WG4:S8. Develop a focused risk section in ESC handbook
Improve	Acquisition	ESC, SMC, ASC, SPAWAR, CECOM	WG4:L7. Develop a structured, risk-based approach for project assessment
Teams	Acquisition	ESC, ASC, CECOM, SPAWAR, SMC	WG4:S15. Recognize team building as a risk in each iteration
Educate	Consultants	MITRE	WG4:S17. Do technology transfer of knowledge mgmt methods/tools
Improve	Consultants	consultants	WG4:L3. Create supporting tools
Teams	Consultants	consultants	WG4:S14. IPT training
Teams	Consultants	MITRE	WG4:S18. Do technology transfer of distributed performance appraisal methods
Adapt	OSD	OUSD/AR, Mr. Soloway	WG1:3. Streamline the contracting process

Adapt	OSD	USD/AT&L, Mr. Gansler	WG1:4. Strengthen the IPPD policy to include ways to build teams and trust in EA
Adapt	OSD	OSD office handling personnel issues	WG1:6. Coordinate personnel changes with increment boundaries
Adapt	OSD	McNutt-SAF/AQ, Ferguson-OSD	WG4:S4. Brief DoD leadership on workshop results
Adapt	OSD	AC2ISRC	WG4:S5. Submit a spiral pilot project through the WRAP
Adapt	OSD	OSD	WG4:L4. Work with Congress to improve the funding model for DoD projects
Adapt	OSD	OSD	WG5:2. Convene a DoD/industry IPT to address acquisition process and contracting mechanisms
Educate	OSD	DoD, DCMC	WG4:L5. Develop a guide to the spiral model for military use
Promote	OSD	OSD	WG4:L6. Document governmental and military case studies of the use of the spiral model
Teams	OSD	all	WG4:S13. Plan for more face-to-face at project onset
Improve	Training	DAU	WG1:5. Define and certify core competencies in Evolutionary Acquisition
Educate	Training Consultants	SEI, DSMC, consultants	WG4:S2. Develop a range of training courses
Adapt	Training OSD	DAU, DSMC, SEI, PMI	WG2:3 Enhance program management to support effective integration in spiral developments
Educate	Training OSD	INCOSE, DAU, DSMC, corp. with training programs, universities	WG2:1. Pay greater attention to education and selection of integration practitioners
Improve	Training OSD	DAU, DSMC, AFOTEC, INCOSE	WG2:4. Adopt these specific testing practices
Define	Universities	INCOSE, IEEE	WG1:1. Define evolutionary acquisition and its relation to spiral development
Define	Universities	future workshops, all spiral model promulgators	WG4:S1. Develop and evangelize the definition of the spiral model
Define	Universities	Barry Boehm	WG5:1. Publish the definition of SDM
Define	Universities	CSE	WG5:6. Elaborate spiral invariants for stakeholder tasks to explicitly require “collective decision making”
Promote	Universities	SEI	WG4:S11. Create Project Management Institute (PMI) connection to spiral
Promote	Universities	universities	WG4:L1. Incorporate spiral development in university courses
Educate	Universities	Barry Boehm, university faculty	WG4:L2. Write books
Educate	Universities	SEI, CSE, universities	WG4:R5. Simulate/emulate distributed teaming in education environment
Educate	Universities	CSE	WG5:4. Write a “field guide” to enacting SDM
Promote	Universities	CSE	WG3:1. Publish Success Stories

Promote	Universities	SEI, universities	WG4:S3. Document commercial case studies of the use of the spiral model
Promote	Universities	CSE and CMMI	WG5:5. CMMI consider spiral model as acceptable alternative practice
Study	Universities	SEI, universities	WG4:S7. Study and disseminate the business case for using spiral
Study	Universities	SEI, CSE, universities	WG4:R1. Develop experimental validation of spiral method
Study	Universities	SEI, CSE, universities	WG4:R2. Gather ROI data
Study	Universities	SEI, CSE, universities	WG4:R3. Understand culture, policy and practice implications for spiral success
Study	Universities	CSE	WG5:3. Collect data and experience and address the scale issue
Teams	Universities	CSE, ESC	WG4:S12. Pilot dist. Team w/ USC webcasting
Teams	Universities	SEI, CSE, universities	WG4:S16. Study ROI of distributed vs. colocated
Teams	Universities	universities	WG4:L8. Develop better ways of doing distributed teaming
Teams	Universities	SEI, CSE, universities	WG4:R4. Understand human factors for successful distributed teaming
Teams	Universities Consultants	Barry Boehm, consultants	WG4:S9. Teach teams how to identify and select incentives
Adapt	Universities OSD	SEI, IDA	WG1:2. Formulate incentives to adopt evolutionary acquisition practices
Promote	Universities Training	CSE, SEI, INCOSE, etc.	WG2:2. Deliberately build an SDM community
Teams	Universities Vendors	universities, vendors	WG4:L9. Develop better collaborative tools

Appendix B Boehm's Summary of Recommendations

In a closing note to the workshop, Barry Boehm presented a quick summary of the recommendations. This may serve as a better overview than the more complete treatment in Chapters 2 and 3 above. The second column has been added here.

Task	Class	Working Group					Agents
		1	2	3	4	5	
Clear definitions, publish paper	Define	of EA			√	√	EIA, INCOSE (EA), SEI, university's, next workshop
Refine spiral development handbook	Educate				√	√	MITRE, SEI, USC, others
Education, selection, qualification	Educate	of EA	SE/ SWE				DAU, DMSC, corp's, univ's, prof. Soc's
Community building, team building, associated incentives, culture change	Promote	of EA	For IPT's		√√		All levels of organizations
Better contracting mechanisms	Adapt	√	√		√		OSD, DAV, DSMC, INCOSE
Better policy	Adapt	For IPPD			√		OSD
Coordinate personal shifts, increment boundaries	Adapt	√					
Work testing implications of spiral	Study		√				
Publish success stories	Promote			√	√	√	USC, SEI, univ's
Provide business case, validation	Promote				√		
Rapid early funding increments	Adapt	√			√		OSD; Service Experiments
Propagate workshop results – debriefing, actions, follow on workshops, IPTs	Promote				√	√	
Relate CMMI to spiral	Promote					√	USC, SEI

Acronyms

AC2ISRC	Aerospace Command and Control, Intelligence, Surveillance, and Reconnaissance Command (Air Force)
AFOTEC	Air Force Operational Test and Evaluation Center
ASC	Aeronautical Systems Center
C2ISR	Command and Control, Intelligence, Surveillance, and Reconnaissance
CECOM	US Army Communications-Electronics Command
CIO	Chief Information Officer
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CMU	Carnegie Mellon University, home of SEI
COTS	Commercial-Off-The-Shelf
CSE	Center for Software Engineering, USC
DAU	Defense Acquisition University
DCMC	Defense Contract Management Command
DoD	Department of Defense
DSMC	Defense Systems Management College
EA	Evolutionary Acquisition
ESC	Electronic Systems Command (Air Force)
ESP	Evolutionary Spiral Process (Software Productivity Consortium)
FAA	Federal Aviation Agency
FFRDC	Federally Funded Research and Development Center
IDA	Institute for Defense Analysis
INCOSE	International Council on Systems Engineering
IOC	Initial Operating Capability
IPPD	Integrated Product and Process Development
IPT	Integrated Product Team
JAD	Joint Application Development
LCA	Life Cycle Architecture
LCO	Life Cycle Objectives
MBASE	Model-Based Architecting and Software Engineering
MITRE	MITRE (an FFRDC)
OSD	Office of the Secretary of Defense
OUSD/AR	Office of the Under Secretary of Defense / Acquisition Reform
PMI	Program Management Institute
QFD	Quality Function Deployment
ROI	Return on investment
RUP	Rational Unified Process
SAF/AQ	Secretary of the Air Force/Acquisition
SAIC	Science Applications International Corporation
SA/SD	Structured Analysis/Structured Design
SDM	Spiral Development Model
SEI	Software Engineering Institute, CMU
SMC	Air Force Space and Missile Systems Center

SPAWAR	Space and Naval Warfare Systems Command
SPC	Software Productivity Consortium
TSPR	Total System Performance Responsibility
UML	Unified Modeling Language
USC	University of Southern California, home of CSE
USD/AT&L	Under Secretary of Defense/Acquisition, Technology, and Logistics
WG	Work Group (of the workshop)
WRAP	Warfighter Rapid Acquisition Process (SAF/AQ)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (LEAVE BLANK)	**2.** REPORT DATE June 2000	**3.** REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE Spiral Development – Building the Culture A Report on the CSE-SEI Workshop, February 2000		**5.** FUNDING NUMBERS C — F19628-95-C-0003
6. AUTHOR(S) W. J. Hansen, J. Foreman, D. Carney, E. Forrester, C, Graettinger, W. Peterson. P. Place		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		**8.** PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2000-SR-006
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		**10.** SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES		
12.A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		**12.B** DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

A number of organizations are successfully applying the Spiral Development Model (SDM) and finding it valuable in addressing such challenges as rapid development, COTS (commercial-off-the-shelf) software integration, new technologies, and product line management. However, other organizations have experienced difficulties with spiral development—due to over-relaxed controls, underestimated risks, existing sequential development policies, inflexible financing mechanisms, ingrained cultures, and confusion about what spiral development is and how to apply it. To attack these problems, a workshop was held February 9-11, 2000, at the University of Southern California under the sponsorship of its Center for Software Engineering (CSE) and the Software Engineering Institute (SEI) of Carnegie Mellon University. Work groups at the workshop recommended specific actions aimed at building and spreading a culture for the SDM community. These can be described as defining, improving, promoting, and studying SDM, educating about SDM, adapting to SDM, and enhancing teamwork. This report summarizes the workshop and presents its recommendations.

14. SUBJECT TERMS Spiral Development Model, SDM, Evolutionary Acquisition, CSE-SEI Workshop	**15.** NUMBER OF PAGES 61
16. PRICE CODE	

17. SECURITY CLASSIFICATION OF REPORT	**18.** SECURITY CLASSIFICATION OF THIS PAGE	**19.** SECURITY CLASSIFICATION OF ABSTRACT	**20.** LIMITATION OF ABSTRACT
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**UL**

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102