



Spiral Acquisition of Defense and Space Systems of Systems

Barry Boehm, USC

Keynote Address, GSAW 2004

April 1, 2004



Outline

- **Trends in Defense and Space Systems of Systems**
- **Role of Spiral Development**
 - Concurrent engineering of requirements and architecture; systems and software
 - Emphasis on risk management
- **Example system-of-systems top-10 risk list**
 - Representative risks and mitigations
- **Conclusions**



Trends in Defense Software-Intensive Systems

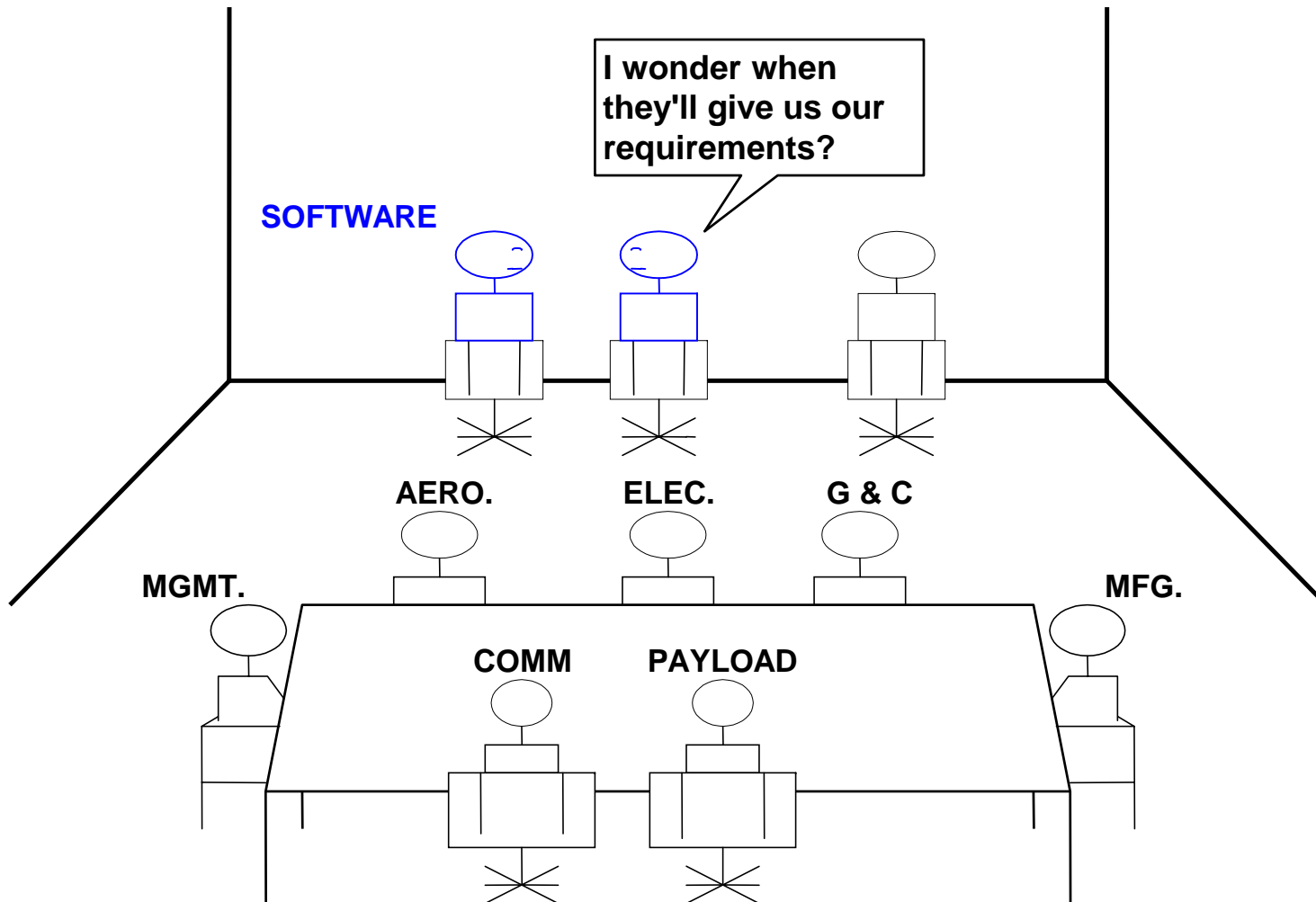
- **Transformational, network-centric systems**
 - These are fundamentally software-intensive
- **Emphasis on joint, interoperable, capability-based systems**
 - And increasingly, systems of systems
- **Increasing requirements emergence, COTS-dependence, environmental change**
- **Traditional sequential acquisition practices increasingly inadequate**
 - Fixed-requirements, -cost, -schedule contracting
 - Waterfall legacies: MIL-STD-1521B, parts of Software CMM



Waterfall Legacies: SW CMM v.1.1

- **Requirements Management, Ability 1:**
“Analysis and allocation of the system requirements
is not the responsibility of the software engineering group
but is a prerequisite for their work.”
- **Concurrent engineering emphasized in CMMI, DoDD 5000.1, DoDI 5000.2**

Resulting Project Social Structure





DoDI 5000.2 “Spiral Development”

Section 3.3.2.1

- **Desired capability is identified**
 - End-state requirements not initially known
- **Requirements refined through demonstration and risk management**
 - Continuous user feedback
 - Each increment provides user the best possible capability
- **Requirements for future increments depend on feedback from users and technology maturation**



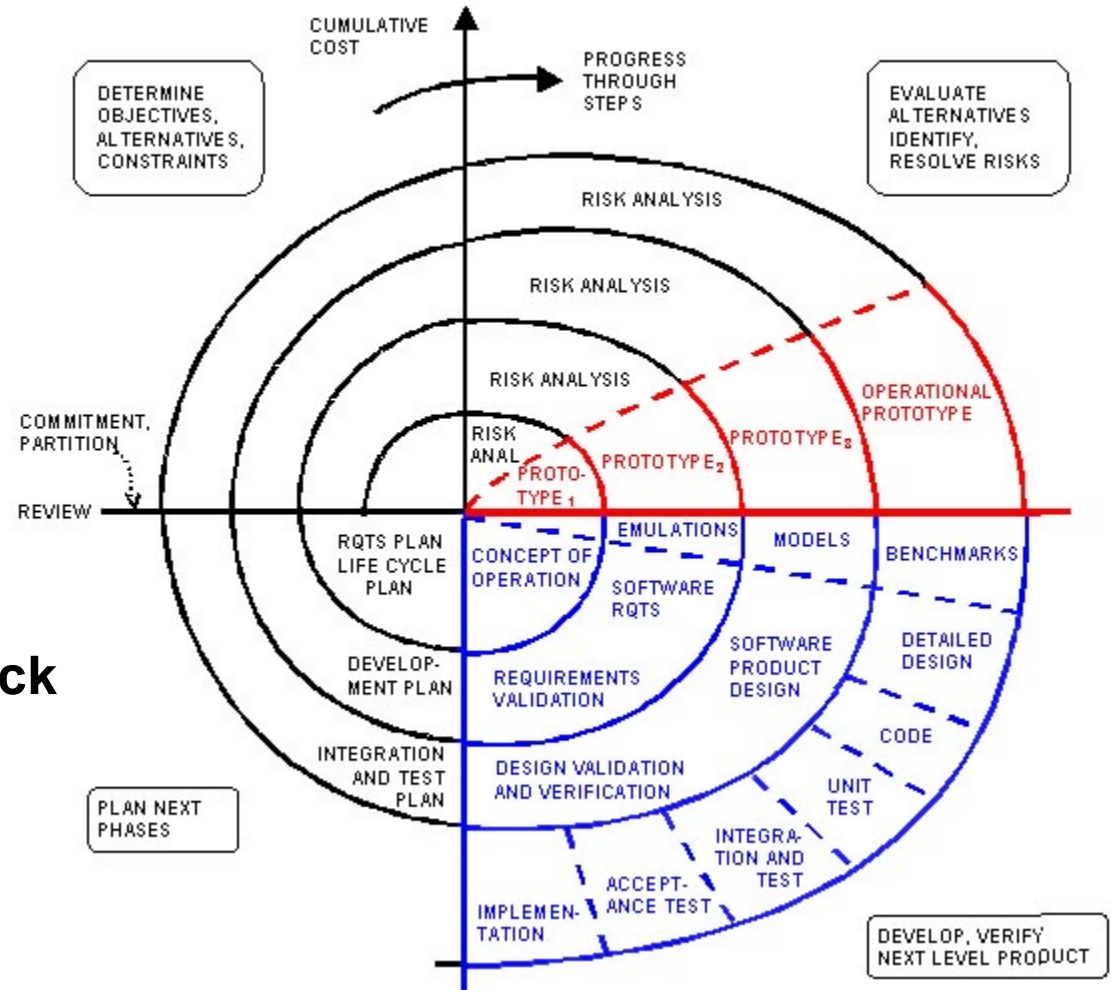
What Is The Win Win Spiral Model?

- **A stakeholder-driven and risk-driven process model generator**
 - There are no one-size-fits-all software process models
 - Different stakeholders and different risks generate different process models
- **A way to perform controlled concurrent engineering**
 - Of systems and software; of development and evolution; of product and process
 - Controlled by anchor point milestones and Feasibility Rationales
- **An upward-compatible extension of the Rational Unified Process**
 - Common risk and anchor-point orientation
 - With stakeholder and value-based extensions
 - Used successfully on a wide variety of applications
- **A way to implement DoDD 5000.1 and DoDI 5000.2**

Original Spiral and Misinterpretations

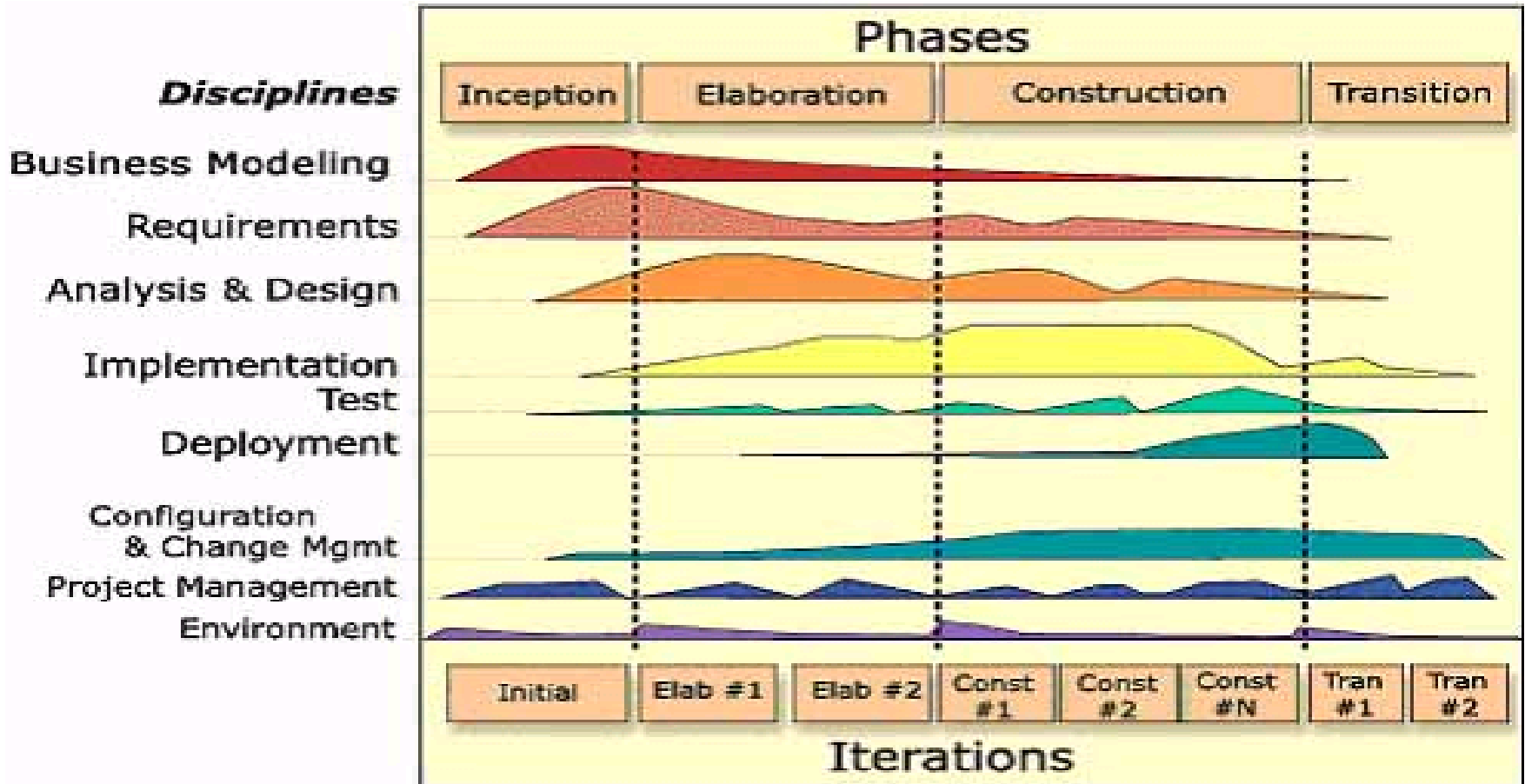
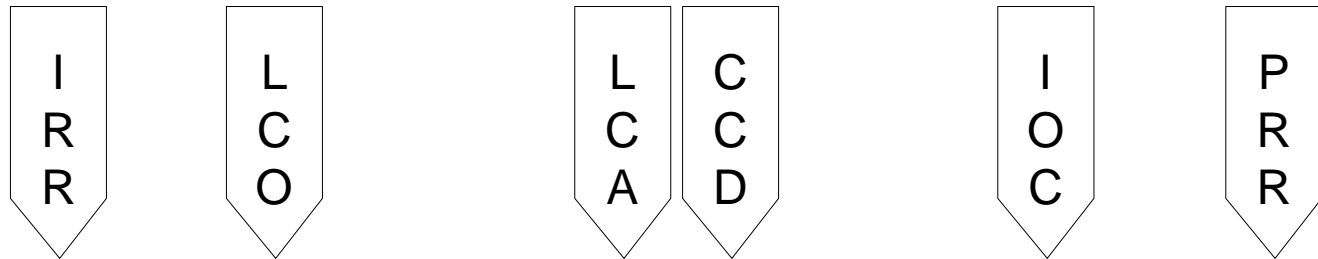
• Common Misinterpretations

- Hack some prototypes
- Fit spiral into waterfall
- Incremental waterfalls
- Suppress risk analysis
- No concurrency, feedback
- One-size-fits-all model





The WWSM Enables Concurrent Engineering





Pass/Fail Feasibility Rationales

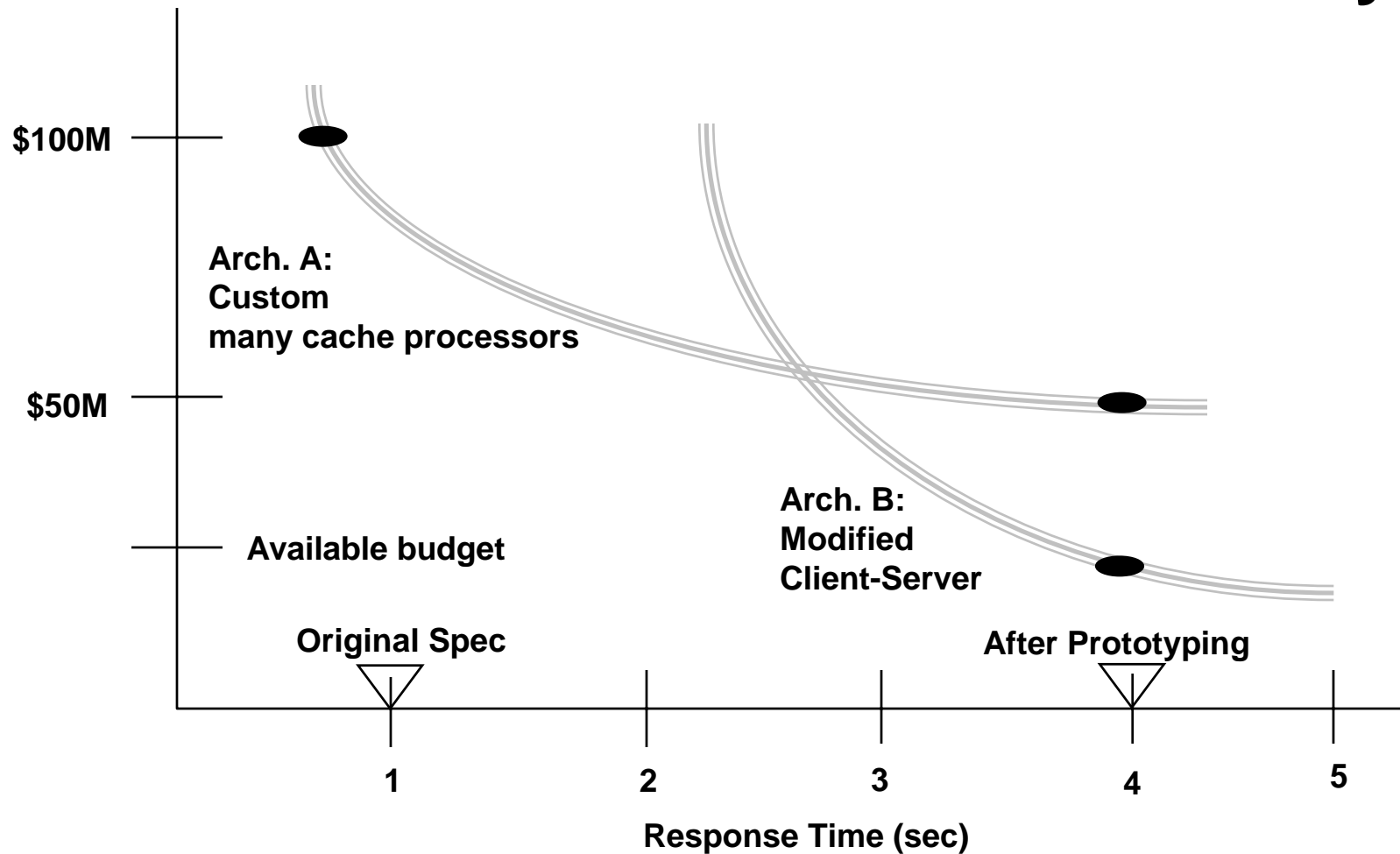
- **Evidence provided by developer and validated by independent experts that:**

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

- **Satisfy the requirements: capability, interfaces, level of service, AND evolution**
 - **Support the operational concept**
 - **Be buildable within the budgets and schedules in the plan**
- **All major risks resolved or covered by risk management plans**
 - **Serves as basis for stakeholders' commitment to proceed**



Effect of Unvalidated Requirements -15 Month Architecture Rework Delay





Effect of Waterfall SEMP and Spiral SDP

- **Delays in starting critical software infrastructure**
 - OS, networking, DBMS, transaction processing, ...
- **Infeasible infrastructure**
 - Premature performance requirements (e.g., 1 second)
- **Premature hardware selection overconstrains software**
 - Can also induce premature COTS commitments
- **Waterfall-based progress payments undermine-spiral tasks**
 - Develop prototypes or get paid for specifications



Top-10 Risks: Software-Intensive Systems of Systems

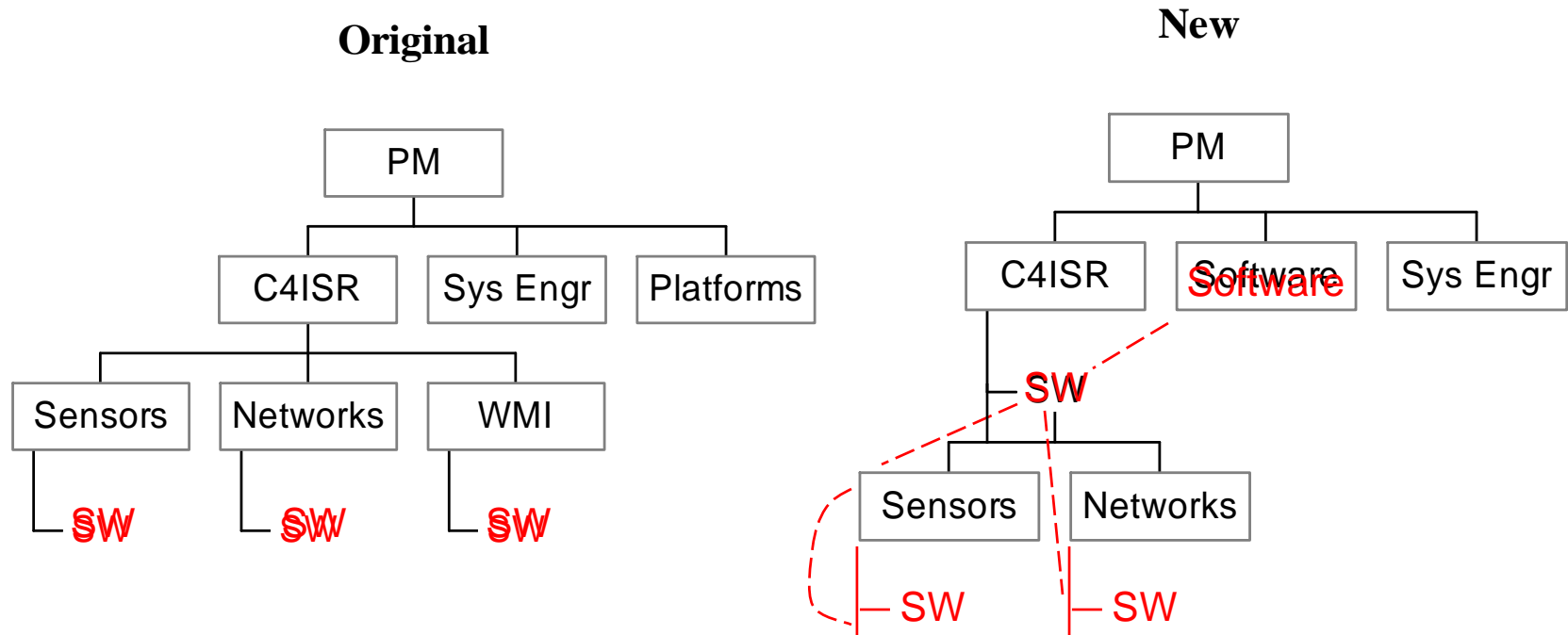
- CrossTalk, May 2004

- 1. Acquisition management and staffing**
- 2. Requirements/architecture feasibility**
- 3. Achievable software schedules**
- 4. Supplier integration**
- 5. Adaptation to rapid change**
- 6. Quality factor achievability and tradeoffs**
- 7. Product integration and electronic upgrade**
- 8. Software COTS and reuse feasibility**
- 9. External interoperability**
- 10. Technology readiness**



Effect of Software Underrepresentation

- Software risks discovered too late
- Slow, buggy change management
- Recent large project reorganization





Need for CRACK Integrated Team Members

- CrossTalk, December 2003

- **Not Collaborative: Discord, frustration, loss of morale**
- **Not Representative: Delivery of unacceptable systems, late rework**
- **Not Authorized: Authorization delays, unsupported systems**
- **Not Committed: Missing homework, discontinuities, delays**
- **Not Knowledgeable: Unacceptable systems, delays, late rework**



Effect of Unvalidated Software Schedules

- Original goal: 18,000 KSLOC in 7 years
 - Initial COCOMO II, SEER runs showed infeasibility
 - Estimated development schedule in months for closely coupled SW with size measured in equivalent KSLOC (thousands of source lines of code):

$$\text{Months} \approx 5 * \sqrt[3]{\text{KSLOC}}$$

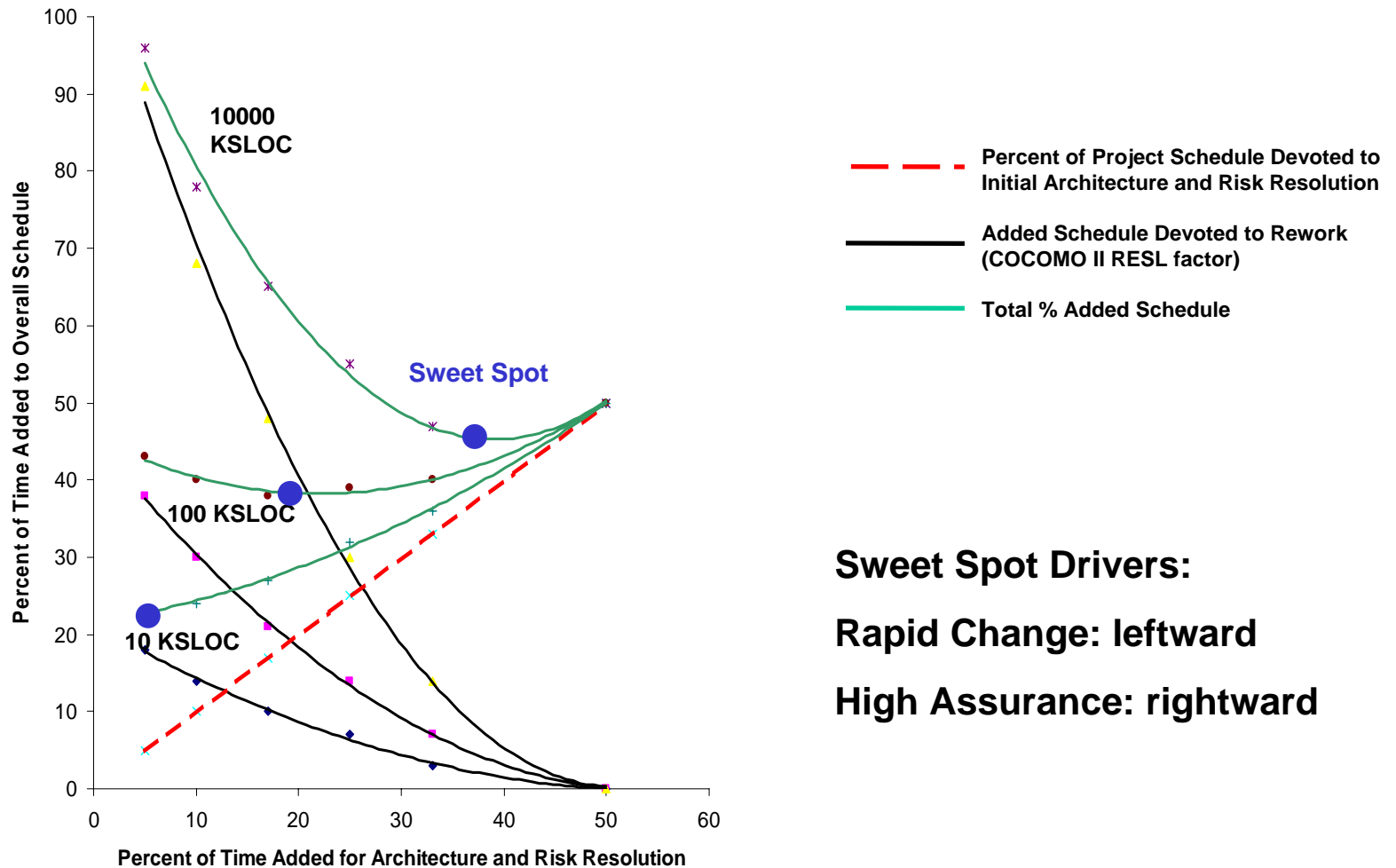
- KSLOC	300	1000	3000	10,000
- Months	33	50	72	108

- Solution approach: architect for decoupled parallel development; Schedule As Independent Variable (SAIV) process



How Much Architecting Is Enough?

-A COCOMO II Analysis



Sweet Spot Drivers:
Rapid Change: leftward
High Assurance: rightward



The SAIV* Process Model

– Cross Talk, January 2002 (<http://www.stsc.hill.af.mil/crosstalk>)

1. Shared vision and expectations management
2. Feature prioritization
3. Schedule range estimation and core-capability determination
 - Top-priority features achievable within fixed schedule with 90% confidence
4. Architecting for ease of adding or dropping borderline-priority features
 - And for accommodating past-IOC directions of growth
5. Incremental development
 - Core capability as increment 1
6. Change and progress monitoring and control
 - Add or drop borderline-priority features to meet schedule

*Schedule As Independent Variable; Feature set as dependent variable

– Also works for cost, schedule/cost/quality as independent variable



Supplier Integration: Rapid Adaptability to Change

- **Risk #4/5. Inflexible subcontracting will be a major source of delays and shortfalls.**
- **Strategy #4/5. Develop subcontract provisions enabling flexibility in evolving deliverables. Develop an award fee structure based on objective criteria for:**
 - **Schedule Preservation**
 - **Cost Containment**
 - **Technical Performance**
 - **Architecture and COTS Compatibility**
 - **Continuous Integration Support**
 - **Program Management**
 - **Risk Management**



Top-10 Risks: Software-Intensive Systems of Systems

- CrossTalk, May 2004

- 1. Acquisition management and staffing**
- 2. Requirements/architecture feasibility**
- 3. Achievable software schedules**
- 4. Supplier integration**
- 5. Adaptation to rapid change**
- 6. Quality factor achievability and tradeoffs**
- 7. Product integration and electronic upgrade**
- 8. Software COTS and reuse feasibility**
- 9. External interoperability**
- 10. Technology readiness**



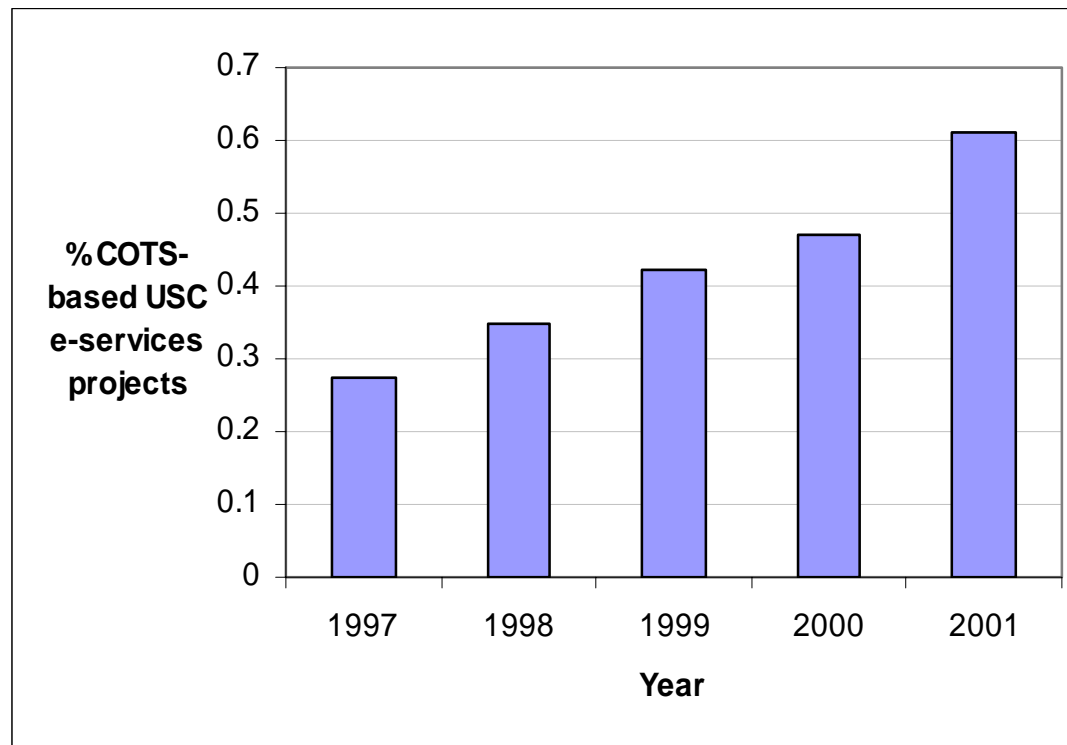
Rapid, Synchronous Software Upgrades

- **Risk #7. Out-of-synchronization software upgrades will be a major source of operational losses**
 - Software crashes, communication node outages, out-of-synch data, mistaken decisions
 - Extremely difficult to synchronize multi-version, distributed, mobile-platform software upgrades
 - Especially if continuous-operation upgrades needed
- **Strategy #7a. Architect software to accommodate continuous-operation, synchronous upgrades**
 - E.g., parallel operation of old and new releases while validating synchronous upgrade
- **Strategy #7b. Develop operational procedures for synchronous upgrades in software support plans**
- **Strategy #7c. Validate synchronous upgrade achievement in operational test & evaluation**



COTS: The Future is Here

- **Escalate COTS priorities for research, staffing, education**
 - It's not “all about programming” anymore
 - New processes required





COTS Upgrade Synchronization and Obsolescence

- **Risk #8a: Many subcontractors means a proliferation of evolving COTS interfaces**
- **Risk #8b: Aggressively-bid subcontracts can lead to delivery of obsolete COTS**
 - **New COTS released every 8-9 months (GSAW)**
 - **COTS unsupported after 3 releases (GSAW)**
 - **An actual delivery: 120 COTS; 46% unsupported**
- **Strategy #8a: Emphasize COTS interoperability in source selection process**
- **Strategy #8b: Contract provisions ensuring delivery of refreshed COTS products.**



Conclusions

- **Defense and space systems undergoing transformation**
- **Need emphasis on spiral systems engineering**
- **Need to integrate systems and software engineering**
- **Spiral approach enables concurrent engineering**
 - **And emphasis on risk management**
- **New systems of systems risks emerging**
 - **And new mitigation approaches**



References

B. Boehm, W. Hansen, “The Spiral Model as a Tool for Evolutionary Acquisition,” Cross Talk, May 2001.

B. Boehm, D. Port, “Balancing Discipline and Flexibility with the Spiral Model and MBASE,” CrossTalk, December 2001, pp. 23-28.

B. Boehm et al., “Using the Win Win Spiral Model: A Case Study,” IEEE Computer, July 1998, pp. 33-44.

B. Boehm, D. Port, L. Huang, and W. Brown, “Using the Spiral Model and MBASE to Generate New Acquisition Process Models: SAIV/ CAIV, and SCQAIV,” CrossTalk, January 2002, pp. 20-25.

D. Reifer and B. Boehm, “A Model Contract/Subcontract Award Fee Plan for Large, Change-Intensive Software Acquisitions,” USC-CSE Technical Report, April 2003.

B. Boehm, A.W. Brown, V. Basili, and R. Turner, “Spiral Acquisition of Software-Intensive Systems of Systems,” Cross Talk, May 2004, pp. 4-9.

MBASE web site : sunset.usc.edu/cse/pub/research/mbase

Spiral/EA workshops web site : www.sei.cmu.edu/cbs/spiral2000

CrossTalk articles: www.stsc.hill.af.mil/crosstalk