



# Product and Process Architectures for Integrating Agile and Plan-Driven Methods

Barry Boehm, USC

XP2006 Keynote Talk  
June 20, 2006



## Summary

- **Under the right conditions, agile methods scale up to teams of teams**
  - But very difficult to go above 100 people
- **However, aspects and portions of large projects need agility**
  - Rapid change, emergent requirements
- **Product and process architectures are emerging to integrate agile and plan-driven methods**
  - Scalable to ultra-large systems
  - Also usable for medium-sized projects, product lines

## Outline

- **Industry experience in scaling up agile methods**
- **Characteristics of ultra-large systems**
- **Product and process architectures for integrating agile and plan-driven methods**
- **Summary and references**

## Industry Experience in Scaling up Agile Methods

- **Case studies from USC Affiliates' workshops**
- **Critical success factors**
- **Scalability strains**



## Agile Experiences

- **Generally successful up to mid-range on 5 critical dimensions (size, criticality, skills, volatility, culture)**
  - Most common agile approaches included selected application of XP and Scrum
  - Some successes with teams of teams, up to 4 teams
  - Common approach was applying XP practices in a Scrum management environment
- **Need to reinterpret some XP practices**
  - Metaphor :: shared vision, common terminology
  - On-site customer :: customer proxy
  - Collective ownership :: Responsible ownership, refactoring; designated functional experts
  - 40-hour work week :: Sustainable development



## Agile Experiences - 2

- **Need to add further techniques**
  - Business process analysis
  - Risk management
  - Wall Gantts; story-based earned value
  - Independent reviews
  - Big-picture architecture
- **Some situation-dependent differences of opinion**
  - Simple design vs. architecture
  - Optional pair-programming vs. reviews
  - Stories vs. formalized acceptance criteria
  - Separate QA, CM



## Agile Experiences - 3

- **Main Challenge: Coexisting with large organization culture**
  - **Organizational processes**
    - Contracts
    - Requirements
    - Change control
    - QA
  - **Closely-coupled non-agile or legacy systems**
  - **Upper manmanagement buy-in**
    - Business case
    - Controls
    - Risk
  - **Sarbanes-Oxley requirements**

06/20/06

© USC-CSE

7



## Sarbanes-Oxley

- **A new US Law**
  - Congress' response to Enron, WorldCom, et al
  - **Internal Controls: evaluate and disclose effectiveness**
  - **Disclose fraud**
  - **Affects public companies and "significant" vendors**
- **Development process must include internal controls for**
  - **Fraud**
  - **Asset Management and Safeguarding**
  - **Financial Reporting**
- **Why is this important to executive management?**
  - **Executives can go to jail.**
  - **IT management can be held grossly negligent and sued by a company or shareholders.**
- **In effect since 2004**

06/20/06

© USC-CSE

8

## What an **Auditor** Looks for...

Processes and tools **over individuals and interactions**

Comprehensive documentation **over working software**

Contract negotiation **over customer collaboration**

Following a plan **over responding to change**

### An Auditor Manifesto?

## Northrop Grumman Pilot

- **Build a large system of analyst workstations working across different databases and networks in a rapid iterative environment using agile processes**
  - Abandon traditional “shall” statement based requirements development
  - Quickly develop a common understanding of how the analysts will utilize the system through a use case development
  - Developers to be responsible for implementing use cases not individual subsystem components. Everyone responsible for daily integration and integrity of the entire system.
  - Schedule use case development through a series of priority setting meetings with the customer and users

## Northrop Grumman Results

- **There are gains to be made in productivity and reduced delivered defects with use case based system specification and implementation**
  - Need to determine the detail required and configuration management required to successfully manage this process
  - Need customer buy in at the beginning along with agreement on how the system will be sold off.
  - The cost to refine the system specification from high level to implementation level is still present, but the result of doing this refinement with use cases is a better common understanding of how the system will function at the end state.
- **Embedded Government representatives with the developers are a necessity for rapid development and acceptance**
  - They must have the authority to make cost/schedule/function trade-offs within the current iteration

## Lockheed Martin

- **5 programs that used Agile to some degree:**
  - Maritime Systems and Sensors
  - Aeronautics
  - Space Systems
- **Organized as follows:**
  - Agile processes/practices used
  - What went well
  - What didn't go quite so well

## Lockheed Martin: Agile Practices Used

- **Project Planning**
  - Sprint Planning
  - Backlog Lists
  - Risk Management
  - Manage scope, not cost/schedule/quality
- **Design**
  - Use agile modeling techniques
  - Keep it simple
  - Document just enough to keep you going
- **Implementation and Test**
  - Pair Programming
  - Refactoring
  - Test driven design
  - Continuous integration
  - Development on the target system

## Lockheed Martin: What went well

- **Team empowerment/group ownership**
- **Plan for and embrace change**
- **Short cycle times allowed for prompt and frequent feedback**
- **Continuous integration**
- **Customer involvement**
- **Pair Programming**

## Lockheed Martin: Needed improvement

- Increased levels of stakeholder involvement
- Manage expectations
- Agile development processes require agile organizational processes

## Agile Tek and Agile+

- Agile+ is XP + ...
  - + Business Process Analyses (BPAs)
  - + Story “Actors”
  - + Delphi-STE Estimation
  - + Risk-Based Situation Audits (RBSAs)
  - + Componentized Architecture
  - + Wall Gantt and Instrument Panels
  - + Automated Contract and Regression Testing
  - + Automatic Document Generation
  - Strict Pair Programming
  - 40-Hour Work Week Restriction
  - + *Flexibility to meet special needs*



## Agile Tek: Solutions to ADM issues

- **Scaling**
  - Componentized Architecture/Interface Definitions
  - Automated Build and Test Processes
  - (Virtual) Team Rooms
- **Unpredictability at Macro Scale**
  - Delphi Estimation
  - STE usage for larger projects
- **Vulnerability to changes at system level**
  - Componentized Architecture
- **Vague about system testing**
  - Automated Contract and Regression Testing
- **Inflexible to special needs**
  - Treat the Special Need as a User Story and prioritize it accordingly
- **Some ADM Practices Are Impractical**
  - Use practices that make sense and work in real-world situations
  - Abandon or modify those that don't
- **Do not Manage Risks Explicitly**
  - Use Risk Based Situation Audits
  - Establish a risk management philosophy

06/20/06

© USC-CSE

17



## Software Experience Center

- **Founded in 1999**
- **Members include ABB, Boeing, DaimlerChrysler, Motorola, Nokia**
- **Goal is to improve members' software competencies, practices**
- **Actively share experiences**
- **Topics include Subcontracting, Requirements Engineering, Product Lines**
- **Facilitators, Experience Collectors**
  - Fraunhofer Virtual Institute for Empirical Software Engineering

06/20/06

© USC-CSE

18

## SEC Study Background

- **Based on 4 SEC meetings and one eWorkshop on agile**
  - Experience was openly shared
  - Decision to compile, report on this experience
- **Complemented with internal reports, published papers**
- **Material from 15 XP-influenced pilots (various level of detail)**
- **Identified common experiences across organizations**
- **The final report is awaiting formal approval**

## Summary of SEC Findings

- **Positive improvements on productivity without compromising quality**
- **Few defects could be traced to XP practices**
- **Best for independent collocated projects involving few people**
- **Can be used for large, complex, safety-critical systems with long life cycles**
- **Always requires tailoring to fit the task at hand**
- **A broader implementation requires changes to culture and quality system**
- **Use of selected agile practices will become more and more common**
- **Agile methods will not be used much, but will influence other processes**
- **Hybrid processes will be the primary way to apply agile principles**

## Critical-Success Factors for Scaling Up

### Agile=

- Timeboxed Iterations + Close stakeholder involvement + Adaptive specs and plans
- Scrum of scrums: need more leaders
- “Spirals” (several months) + “Iterations” (several weeks)
- Architecture essential, but lighter
- Set expectation early; keep setting them
- Hardest-first elements: demos not pretty
- Technical risk vs. keeping sold risk
- Re-culture integration and test towards test-first

## Workshop Problems of Scale - I

- Configuration Management
- Earned Value Tools
  - Agile: features (business value)
  - Traditional: activities
- Stakeholder sign-off requirements
- Planning documentation
- Deployment, life cycle support (training)
  - Long-term life cycle sustainment
  - Decay rate of tacit knowledge (Mack truck risk)

## Workshop Problems of Scale - II

- Risk Management
- Contracted/planned (BDUF) inch-pebble milestones
- Process QA/standard processes
- Process Standards (IEEE, DoD, EIA)
- Designing for the battle vs. the war

## Outline

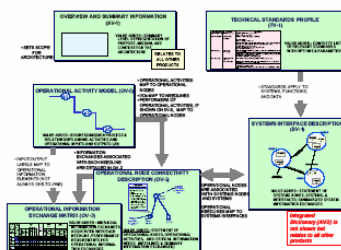
- Industry experience in scaling up agile methods
- ➔ Characteristics of ultra-large systems
- Product and process architectures for integrating agile and plan-driven methods
- Summary and references



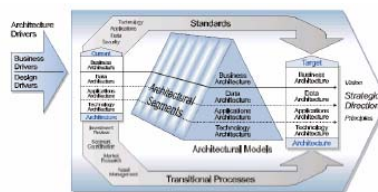
## The Need for Software-Intensive Systems of Systems (SISOS)

- Lack of integration among stove-piped systems causes
  - Unacceptable delays in service
  - Uncoordinated and conflicting plans
  - Ineffective or dangerous decisions
  - Inability to cope with fast-moving events
- Increasing SISOS benefits
  - See first; understand first; act first
  - Network-centric operations coordination
  - Transformation of business/mission potential
  - Interoperability via Integrated Enterprise Architectures

## Integrated Enterprise Architectures



DOD Architectural Framework (DODAF)



Federal Enterprise Architectural Framework (FEAF)

Zachman Framework	Data View	Function View	Network View	People View	Time View	Motivation View
Scope	Business Model	Business Model	Business Model	Business Model	Business Model	Business Model
Business Model	System Model	System Model	System Model	System Model	System Model	System Model
System Model	Technology Model	Technology Model	Technology Model	Technology Model	Technology Model	Technology Model
Technology Model Detailed Representations	Technology Model	Technology Model	Technology Model	Technology Model	Technology Model	Technology Model

Zachman Framework

## Complexity of NCSOS Solution Spaces

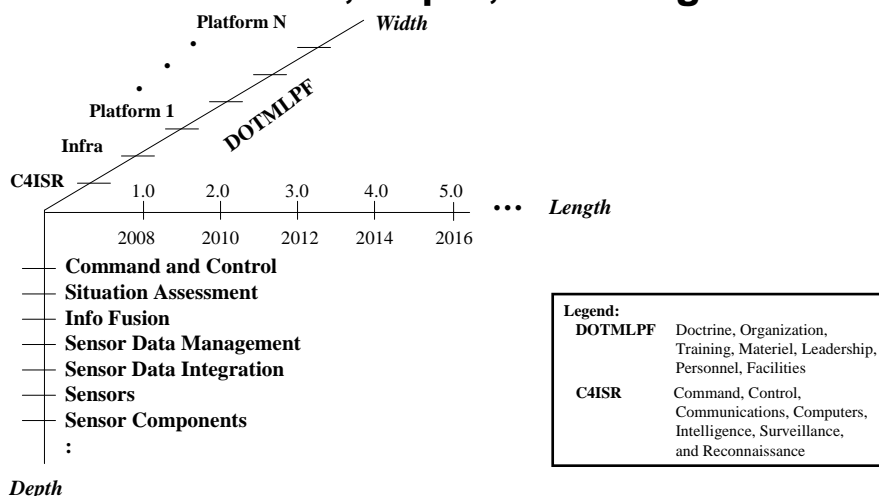
- **Size: 10-100 MLOC**
  - **Number of external interfaces: 30-300**
  - **Number of “Coopetitive” suppliers: 20-200**
    - Even more separate work locations
  - **Depth of supplier hierarchy: 6-12 levels**
  - **Number of coordination groups: 20-200**
    - Reviews, changes, risks, requirements, architecture, standards, procedures, technologies, -ilities, integration, test, deployment, personnel, infrastructure, COTS,...
    - Key personnel spend 60 hours/week in meetings
  - **Unprecedentedness**
  - **Emergence**
  - **Rapid change**
- Necessarily software-intensive...***

06/20/06

© USC-CSE

29

## Complexity of Solution Spaces - Breadth, Depth, and Length



06/20/06

© USC-CSE

30

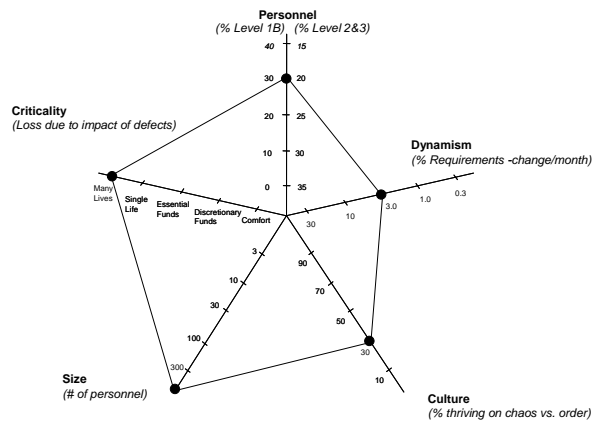
## Need Simultaneous Agility and Discipline

- **Discipline for planning and structure**
  - Foundations (architecture, organizations)
- **Agility to handle the environment**
  - Rapid, continuous change
  - Concurrency of development
  - Many suppliers, coordination groups, external interfaces
- **Use risk analysis to determine how much agility, discipline is enough**

## SISOS-Relevant Agile Practices

- **Short stabilized increments (+)**
  - Prioritized feature backlog
- **Continuous customer-developer participation (+)**
- **Early test; continuous integration (+)**
- **Tacit interpersonal vs. explicit documented knowledge ( $\pm$ )**
- **Welcome changing requirements ( $\pm$ )**
- **Simple design (-)**
  - Just for current increment
  - Refactor to accommodate later capabilities

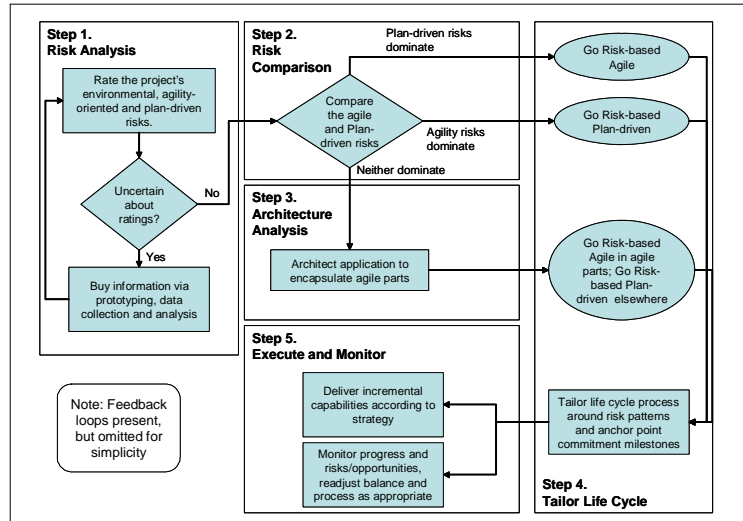
## SISOS Agile/Plan-Driven Profile



## Outline

- Industry experience in scaling up agile methods
- Characteristics of ultra-large systems
- ➔ Product and process architectures for integrating agile and plan-driven methods
- Summary and references

## Using Risk to Balance Discipline and Agility - Overview



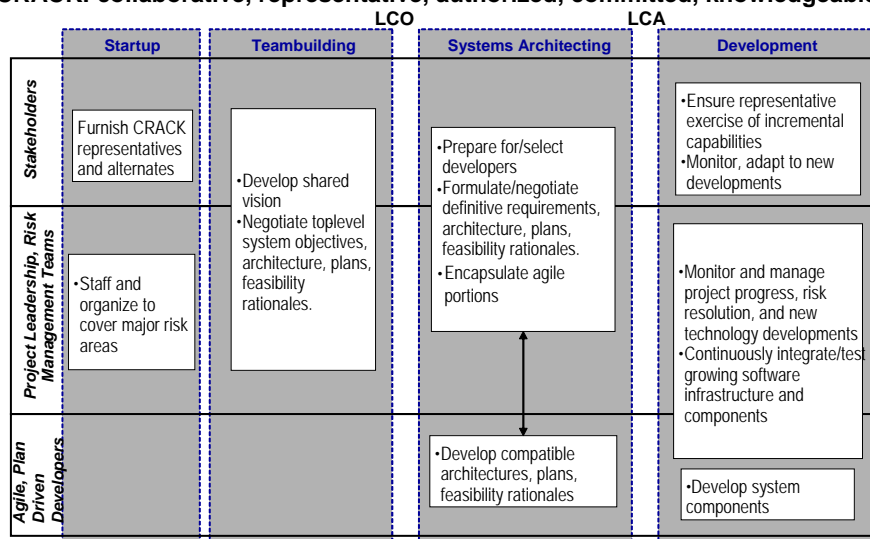
06/20/06

© USC-CSE

35

## SISOS Agile/Plan-Driven Strategy

– CRACK: collaborative, representative, authorized, committed, knowledgeable

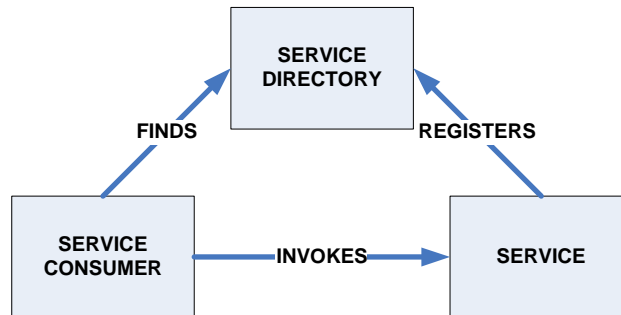


06/20/06

© USC-CSE

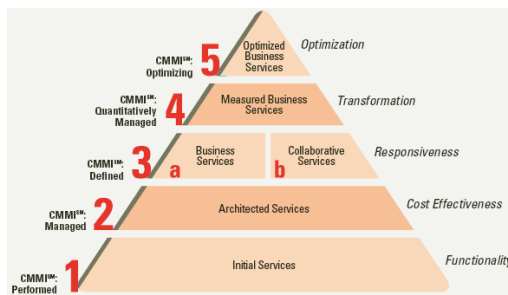
36

## Service-Oriented Architecture Overview



## Service-Oriented Architecture – Maturity Model

*\*A New Service-Oriented Architecture (SOA) Maturity model -- Sonic Software Corporation, AmberPoint Inc., BearingPoint, Inc., Systinet Corporation.*

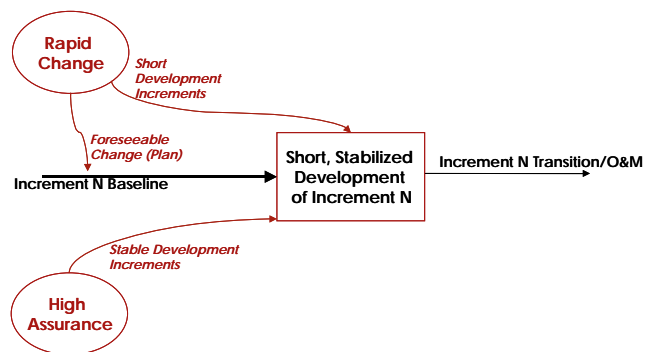


Maturity Level	Prime Business Benefits	Scope	Critical Technology Success Factors	Critical People & Organizational Success Factors	Selected Relevant Standards
1. Initial Services	New functionality	R&D experimentation, Pilot projects Web site, Portal, Custom integrations, Small number of services	Standards, Legacy Integration	Developers learn service development skills Developer Sponsorship	XML, XSLT, WSDL, SOAP, Java, .NET
2. Architected Services	IT cost reduction and control	Multiple integrated applications	Support for heterogeneity and distributed systems, Reliable Messaging, Mediation, Ease of deployment, Database integration, Versioning, Internal Security, Performance management	Architecture group provides leadership SOA Competency Center CIO Sponsorship	UDDI, WS-ReliableMessaging, WS-Policy, WS-Addressing, XQuery, WS-Security, SAML

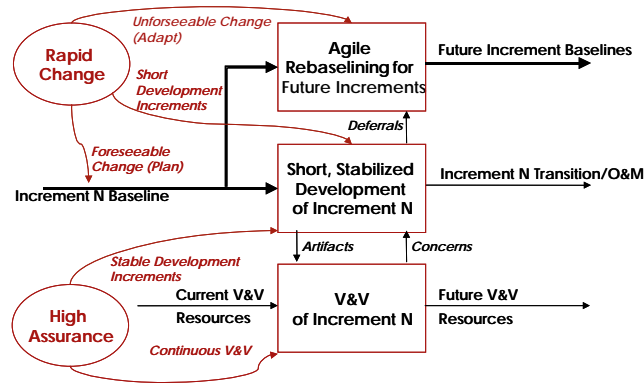
## Service-Oriented Architecture – Maturity Model

3.a. Business Services	Business responsiveness — change business processes quickly and effectively	Business processes across business unit or enterprise	Reuse, Ease of modification, Availability, Business process rules, Event-driven processes, Composite applications	IT Partnership with Business Partnership across Organizations SOA Life-cycle Governance Executive commitment Event-driven design skills Business Unit Manager Sponsorship	WS-BPEL
3.b. Collaborative Services	Business responsiveness — collaboration with business and trading partners	Services available to external partners, Cross-enterprise	External services enablement, Cross-enterprise security, Translation of cross-enterprise protocols, Long-running transactions		RosettaNet, ebXML, WS-Trust
4. Measured Business Services	Business transformation from reactive to real-time, Meet business performance metrics	Business unit or enterprise, Cross-enterprise	Business Activity Monitoring, Event Stream Processing, Complex Event Processing, Event-driven dashboards and alerts	On-going business process evaluation and response CFO sponsorship	
5. Optimized Business Services	Business optimization — react and respond automatically	Business unit or enterprise, Cross-enterprise	Event-driven automation for optimization	Continuous improvement culture CEO sponsorship	

## Risk-Driven Scalable Spiral Model: Increment View



## Risk-Driven Scalable Spiral Model: Increment View



## Summary

- **Under the right conditions, agile methods scale up to teams of teams**
  - But very difficult to go above 100 people
- **However, aspects and portions of large projects need agility**
  - Rapid change, emergent requirements
- **Product and process architectures are emerging to integrate agile and plan-driven methods**
  - Scalable to ultra-large systems
  - Also usable for medium-sized projects, product lines



## References

- B. Boehm, R. Turner, Balancing Agility and Discipline, Addison Wesley, 2004.
- B. Boehm, "Some Future Trends and Implications for Systems and Software Engineering Processes," Systems Engineering, 9, 1, 2006 pp. 1-19.
- 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/research/MBASE](http://sunset.usc.edu/research/MBASE)
- J. Surdu and D. Parsons, "Army Simulation Program Balances Agile and Traditional Methods with Success," CrossTalk, April 2006, pp. 4-8.
- Agile workshops web site : [www.cse.usc.edu/events/2004/arr](http://www.cse.usc.edu/events/2004/arr)
- CrossTalk articles: [www.stsc.hill.af.mil/crosstalk](http://www.stsc.hill.af.mil/crosstalk)