Architected Agile Solutions for Software-Reliant Systems

Barry Boehm, JoAnn Lane,
Supannika Koolmanojwong, USC

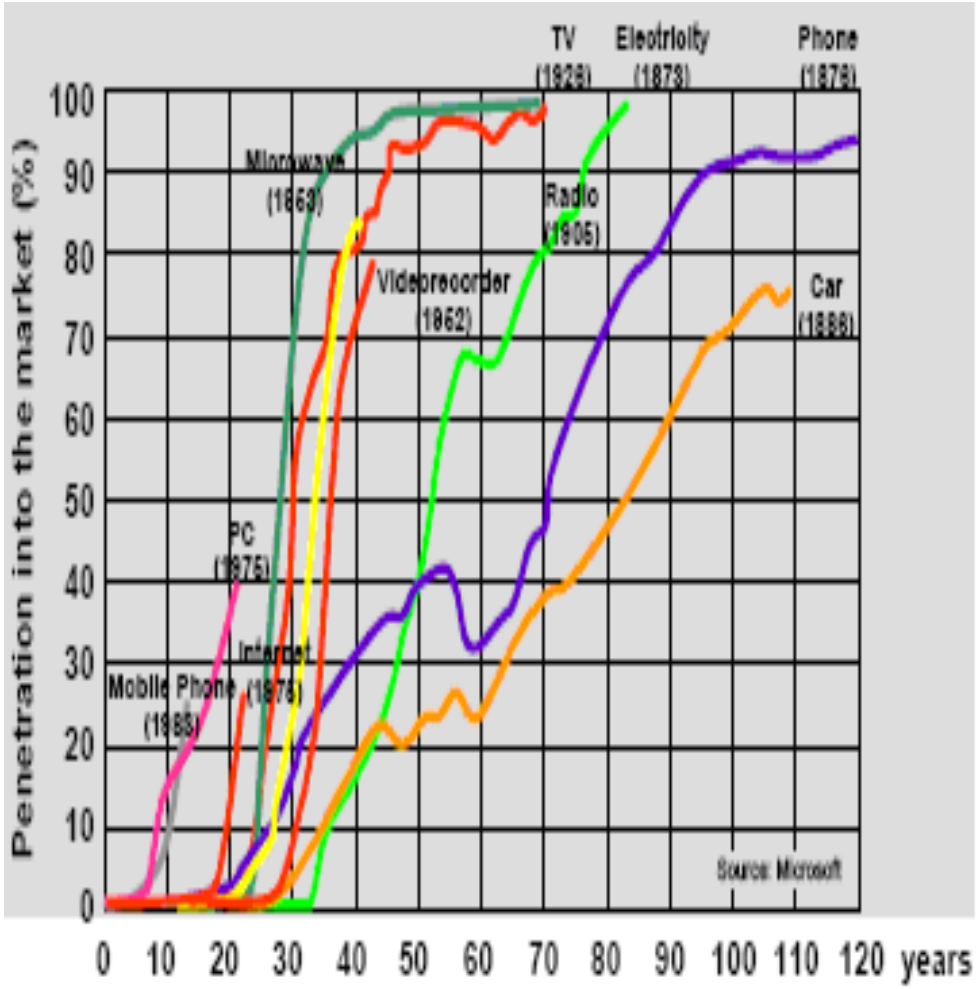
INCOSE IS 2010

July 15, 2010

Outline

- Increasing importance of both agility and quality
  - Scalability, accuracy, availability, safety, …
- Challenges of achieving both agility and quality
- Approaches for achieving both agility and quality
- Case studies and critical success factors
- Conclusions

Need for Agility: Increasing Pace of Change



- Technology change
- Related infrastructure and services
- Marketplace dynamics
- Competition dynamics
- Organizational change
  - Software is critical
  - User agility aids are also critical

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- ***Individuals and interactions*** over **processes and tools**
- ***Working software*** over **comprehensive** documentation
- ***Customer collaboration*** over **contract negotiation**
- ***Responding to change*** over **following a plan**

That is, while there is value in the items on the right, we value the items on the left more.

The Need for Software Quality

- "The world runs on software" – Stroustrup
- "With C, you can easily shoot yourself in the foot. With C++, you can easily blow off your leg" – Stroustrup

- Critical global infrastructure: finance, energy, transportation, communications, trade

- Dependability: everything you depend on
  - Accuracy, adaptability, affordability, availability, …
  - Complex attribute conflicts and tradeoffs

Traditional Quality Approach

- Complete, consistent, testable requirements
- Traceable to design, code, test cases
- Heavyweight documentation

- COCOMO documentation rates, Very High Reliability projects
  - Average 120 pp/KSLOC; median 83; range 32-241
- Rewriting needed for 1000 KSLOC project
  - 160 people; 120,000 pages of documentation
  - 1% change/month: 1200 pages (7.5 pages/person)
  - 10% change/month: 12,000 pages (75 pages/person)

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

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?

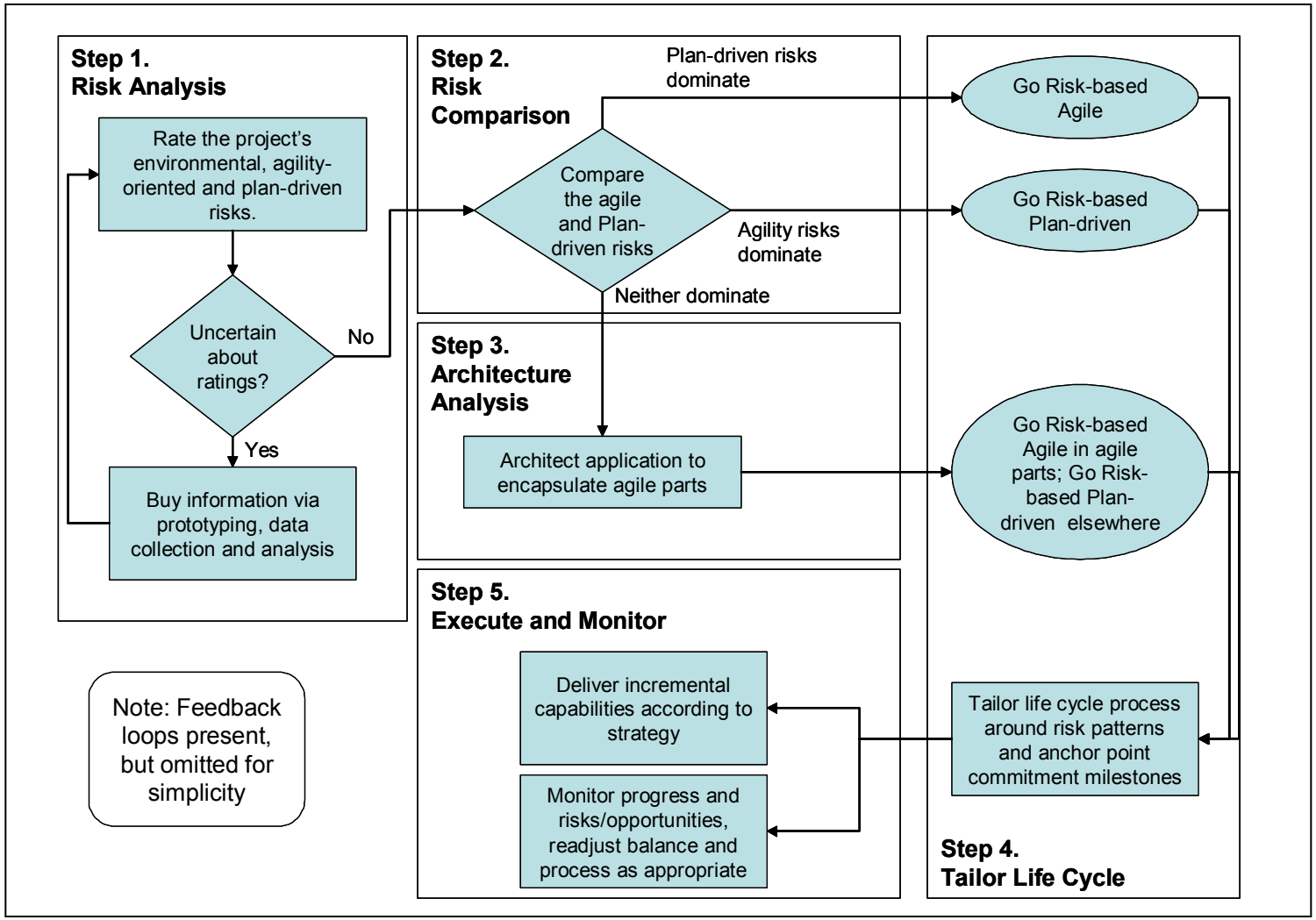
Agile Methods and Quality

- Responding to change over following a plan
  - Major source of software-induced rocket failures
- Small releases: It'll be fixed by next month
  - OK for discomfort; not for safety
- Test-driven development helps, but often leads to patching
  - Example: Ada compiler validation suite

Outline

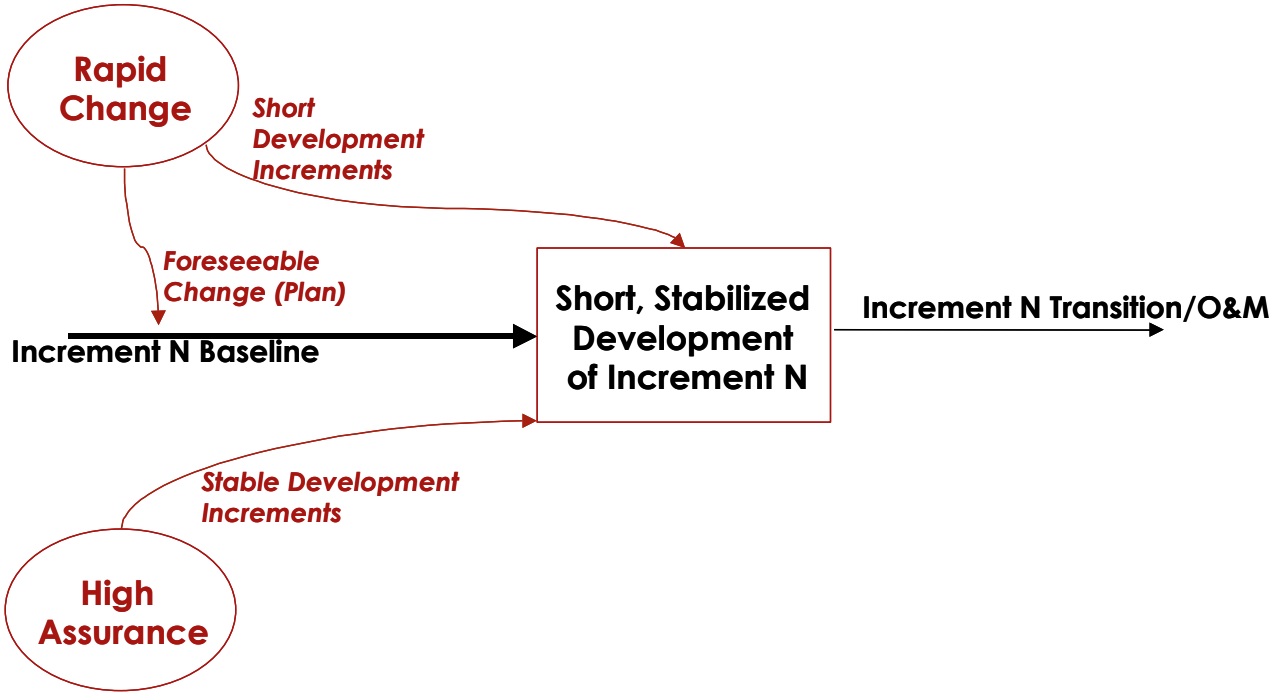
- Increasing importance of both agility and quality
- Challenges of achieving both agility and quality
- Approaches for achieving both agility and quality
- Case studies and critical success factors
- Conclusions

Using Risk to Balance Discipline and Agility - Overview



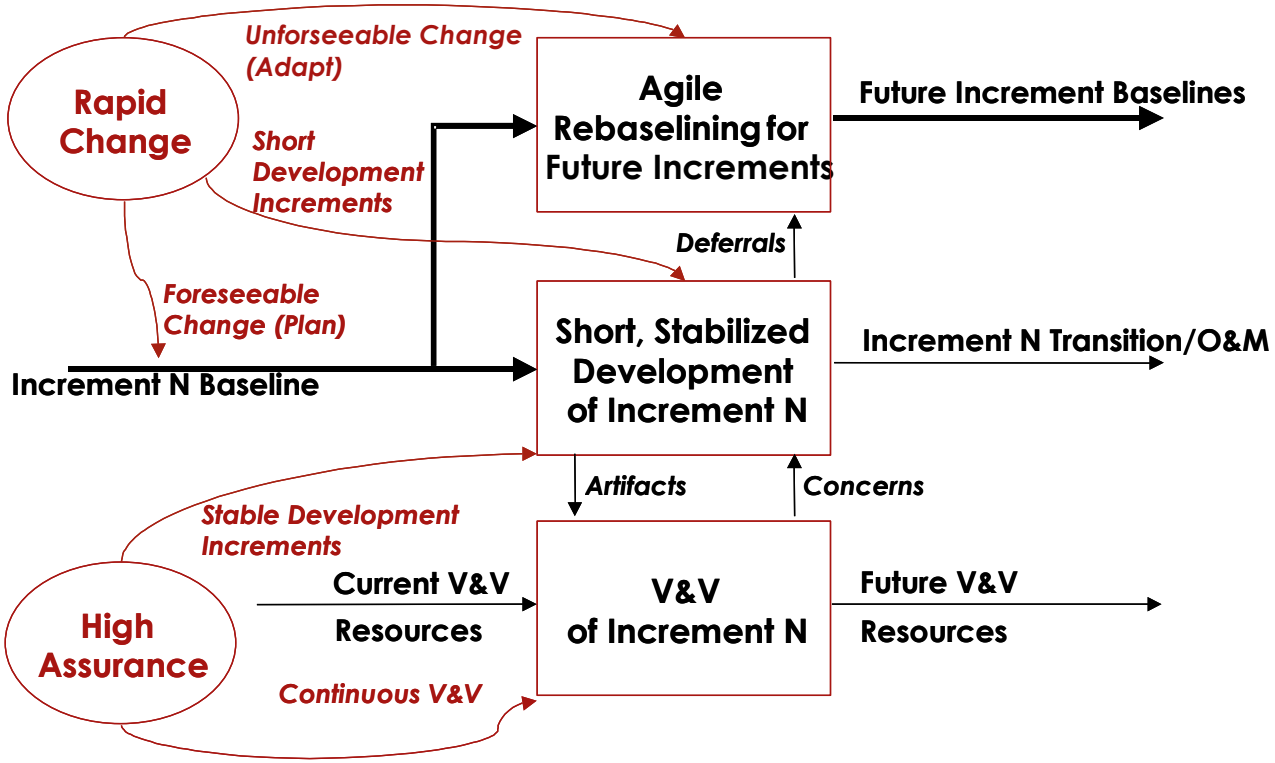
Incremental Commitment Model:

Single Increment View

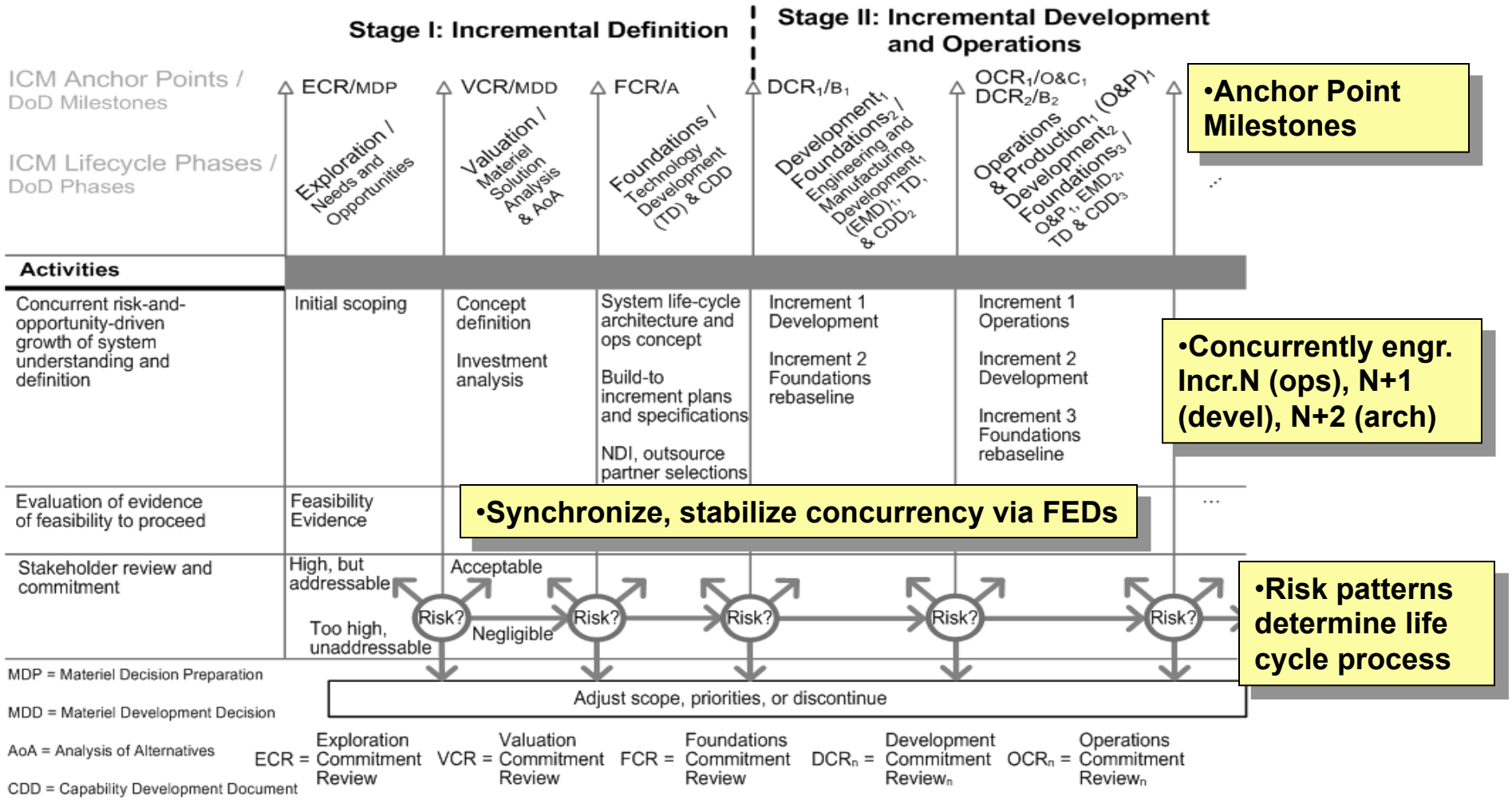


Incremental Commitment Model:

Single Increment View

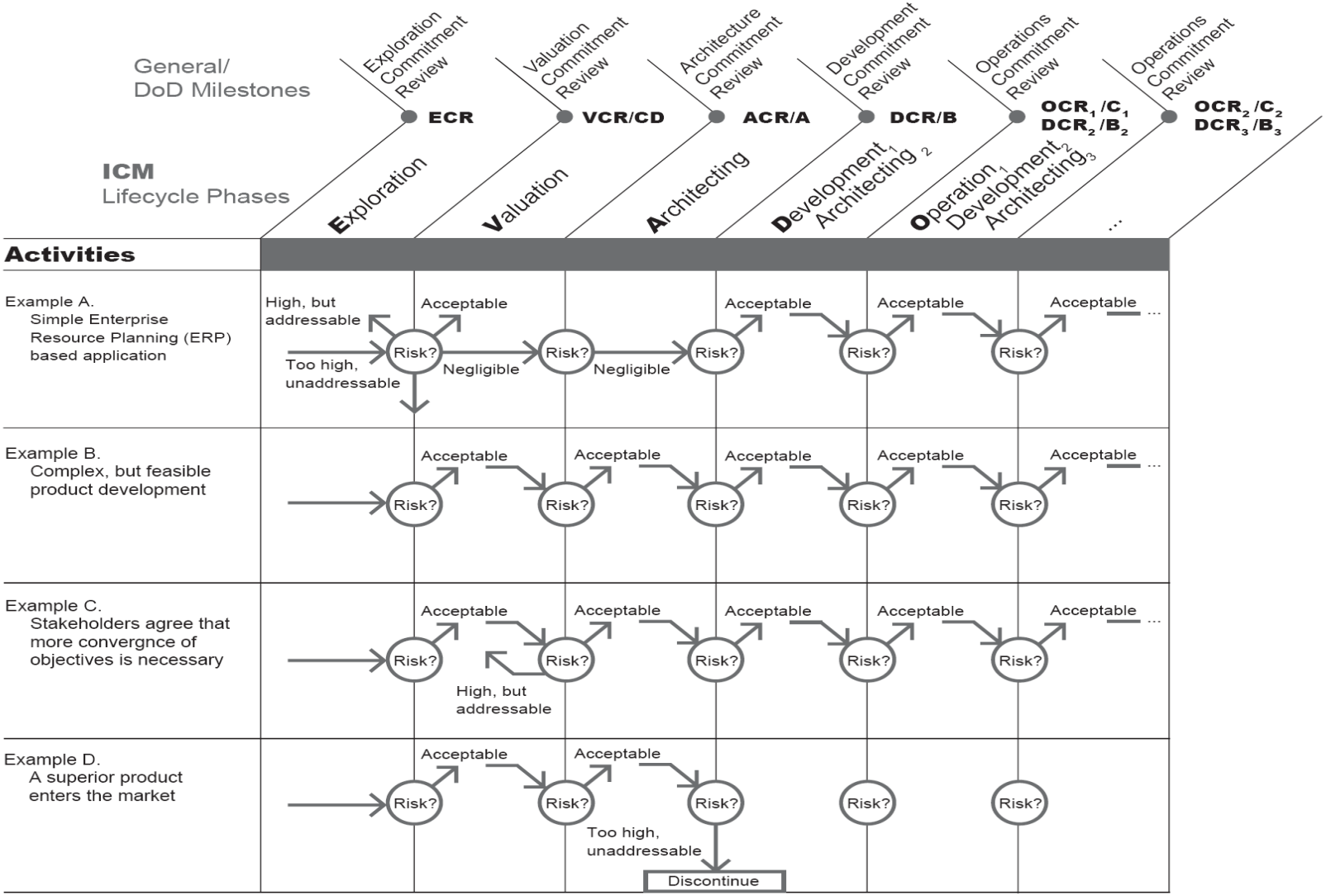


The Incremental Commitment Life Cycle Process: Overview



- **Anchor Point Milestones**
- **Concurrently engr. Incr.N (ops), N+1 (devel), N+2 (arch)**
- **Synchronize, stabilize concurrency via FEDs**
- **Risk patterns determine life cycle process**

Different Risk Patterns Yield Different Processes



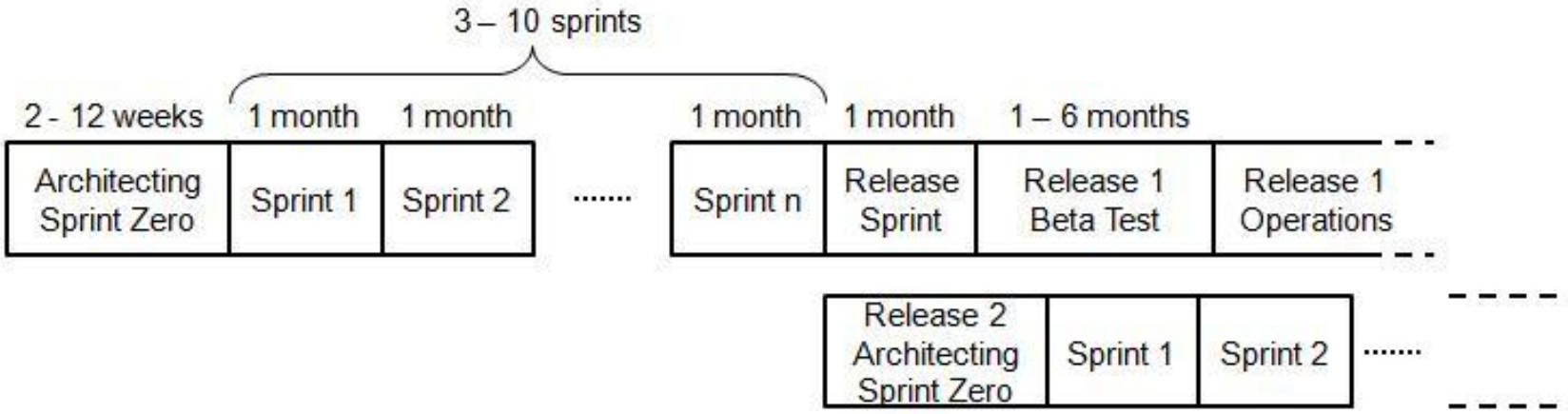
Outline

- Increasing importance of both agility and quality
- Challenges of achieving both agility and quality
- Approaches for achieving both agility and quality
- Case studies and critical success factors
- Conclusions

Architected Agile Corporate Transformations

- A US medical services company
  - Problems
    - Global (1,000) software developers
    - Slow, error-prone, and incompatible software applications and process
  - Solutions
    - Team leaders from all major sites to architect the framework using architected-agile process approach.
    - Using Scrum of Scrums in a collocated pilot project to build information framework
    - Team leaders returned, led the transformation in their regions
    - Using scrum with XP, dedicate team rooms, daily virtual meeting support

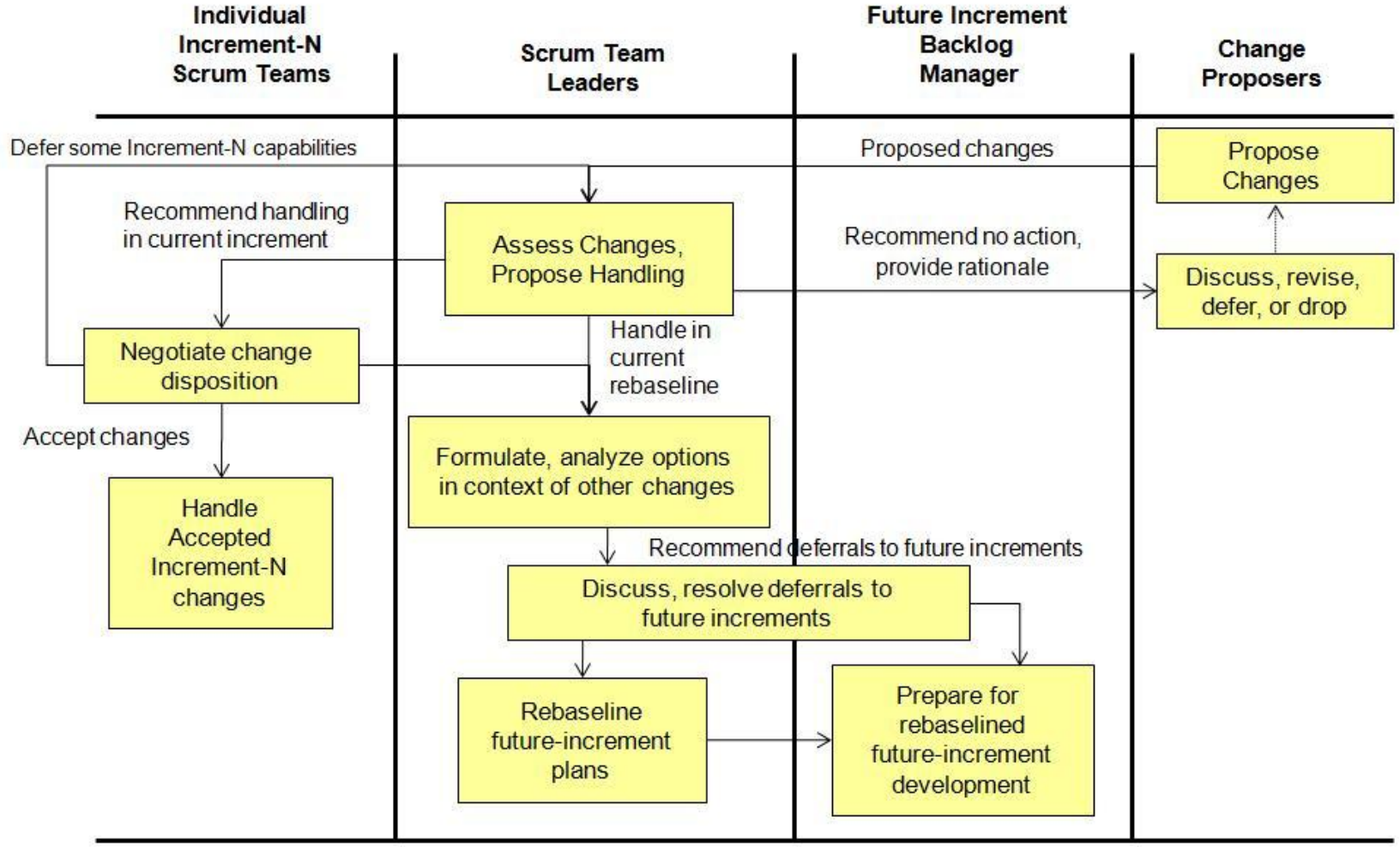
Example of Architected Agile Process



Architected Agile Corporate Transformations (Case No. 2, 3)

- A world-100 European company with global sites
- A large European IT company with major development centers in Europe
- Key Problems
  - A continuing stream of asynchronous change requests to accommodate new technology, environments
- Solution:
  - Applied key principles of ICM
  - Implement agile change management
  - Architecture resolved

Agile Change Process and Rebaselining



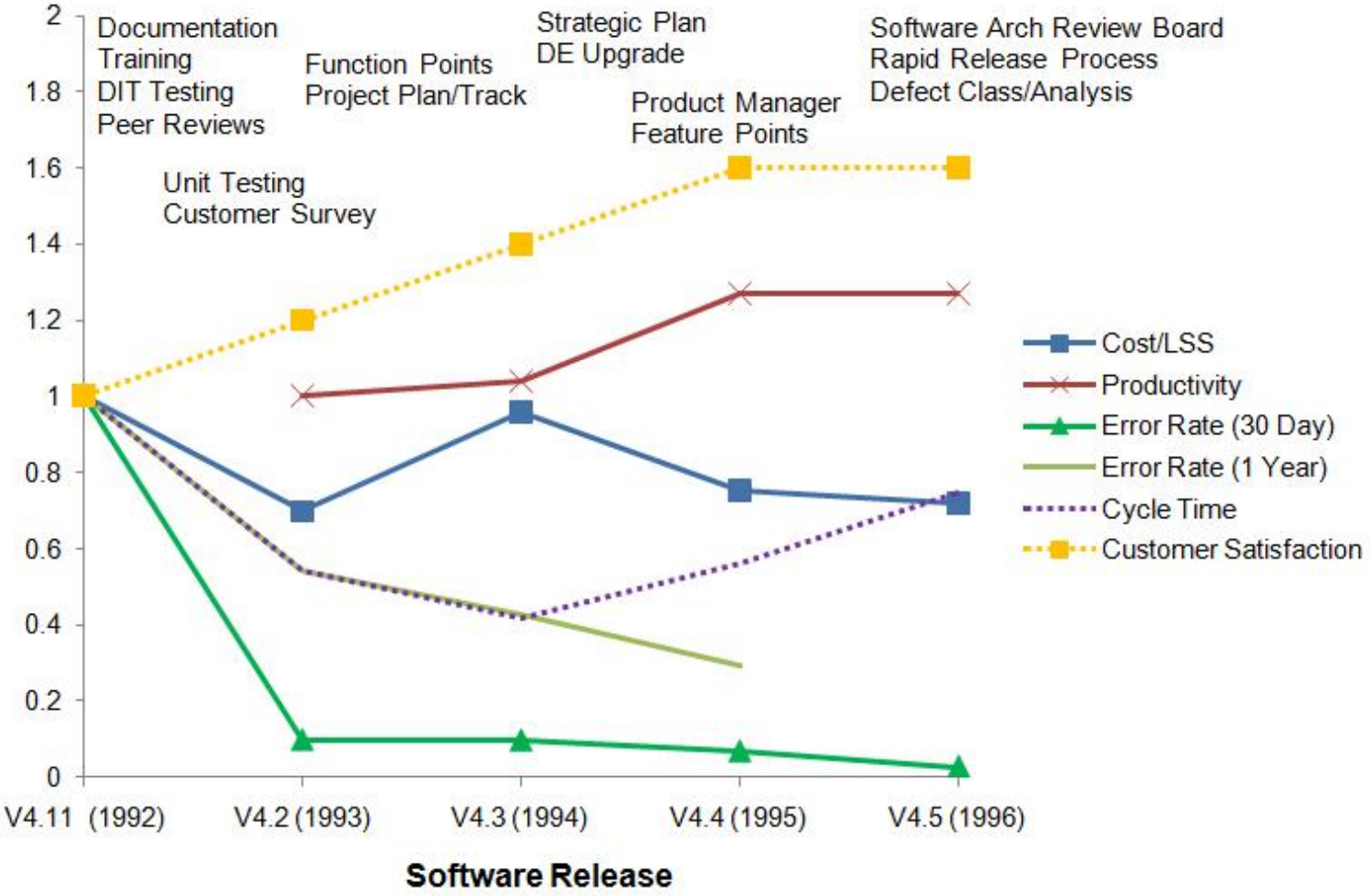
Automated Maintenance Support System

- A major aerospace company
- Using net-centric capabilities for anomaly analysis
- Problems
  - Using scrum of scrums
  - Facing numerous coordination challenges among multi-mission, multi-owner vehicle versions
- Solutions
  - A decentralized scrum-based approach, governed by product framework group (PFG)
  - PFG steer the teams by using Incremental, Iterative and concurrent approach

Major Health Care System

- A major DoD contractor
- Responsible for maintenance, upgrade, and installation of health care system at over 700 sites
- Problems
  - Too much time/costs required for major upgrades
  - Schedule pressures leading to acceptable defect levels
- Solutions
  - Agile analysis of incoming change requests
  - Architecture team to manage/evolve system architecture/ database structure
  - Early stabilization of next release to be deployed
  - Concurrent engineering of future releases
  - Committed stakeholders working with functional area teams
  - Continuous V&V

Results of Incorporating Process Changes Related to Architected Agile ICM Principles



Common Critical Success Factors

- Management commitment, with incremental feasibility checkpoints
  - Clear message about objectives, scope, and strategy
  - Involve top people from stakeholder organizations
  - Build in growth to expansion sites
  - Lead through early successes
- Thoroughly prepare the ground
  - Infrastructure, policies, practices, roles, training
  - Customer buy-in and expectations management
  - Get help from experts
- Make clear what's essential, optional
  - Most frequently, Scrum plus organizational essentials
  - Precede Development Sprints by Architecting Sprint
    - Follow by Release Sprint, beta testing
  - Where needed, work compliant mandate interpretations
- Monitor, reflect, learn, evolve

Conclusions

- Success-critical to achieve both agility and quality
- Hybrid architected-agile methods emerging
  - Incremental commitment framework
  - Early development, validation of scalable architecture
  - Concurrent engineering with synchronization milestones
  - Scrum plus organizational essentials
- Success stories emerging
  - Management commitment to objectives and strategy
    - With incremental feasibility checkpoints
  - Strong core team of technical and management leaders
  - Thorough preparation of organizations, people, infrastructure
    - Involvement, architecture, policies, practices, plans, training
  - Continuous change monitoring and adaptation