



Scaling Up Agility: The Architected Agile Approach

Barry Boehm, USC

JAOO 2009

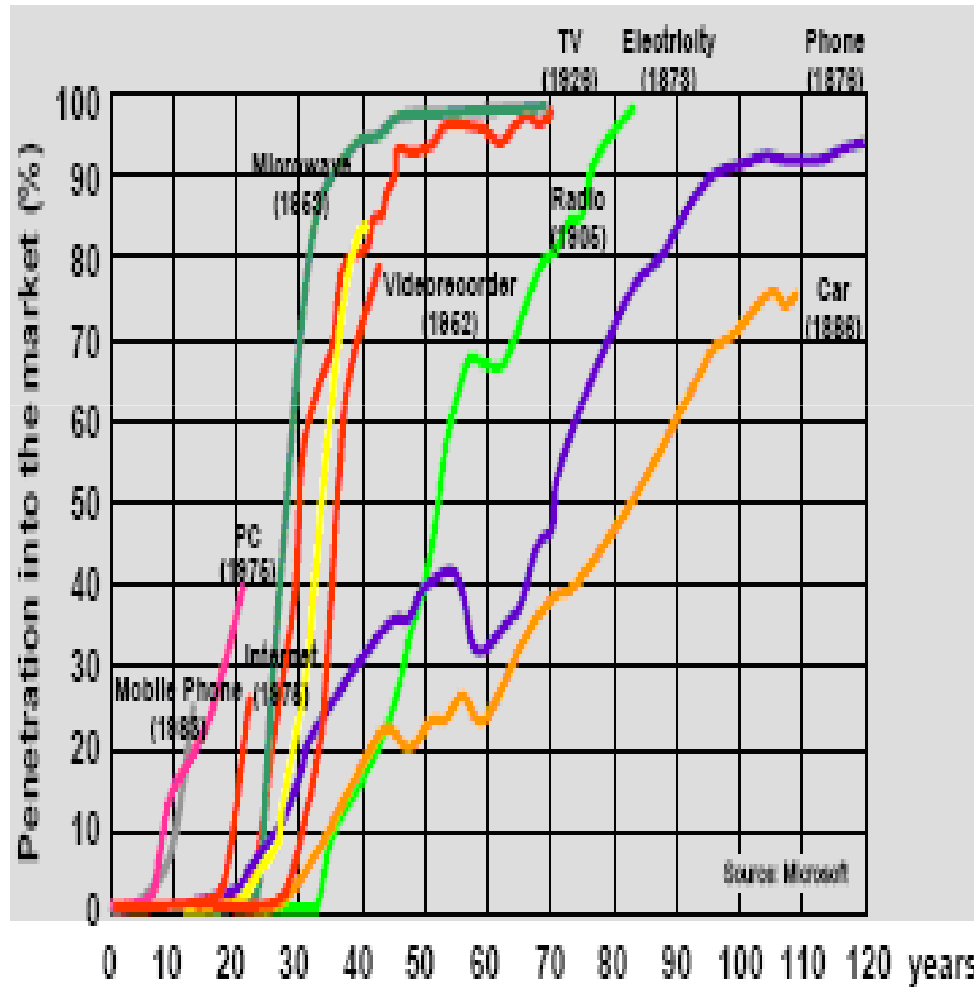
October 5, 2009



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 even more 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...

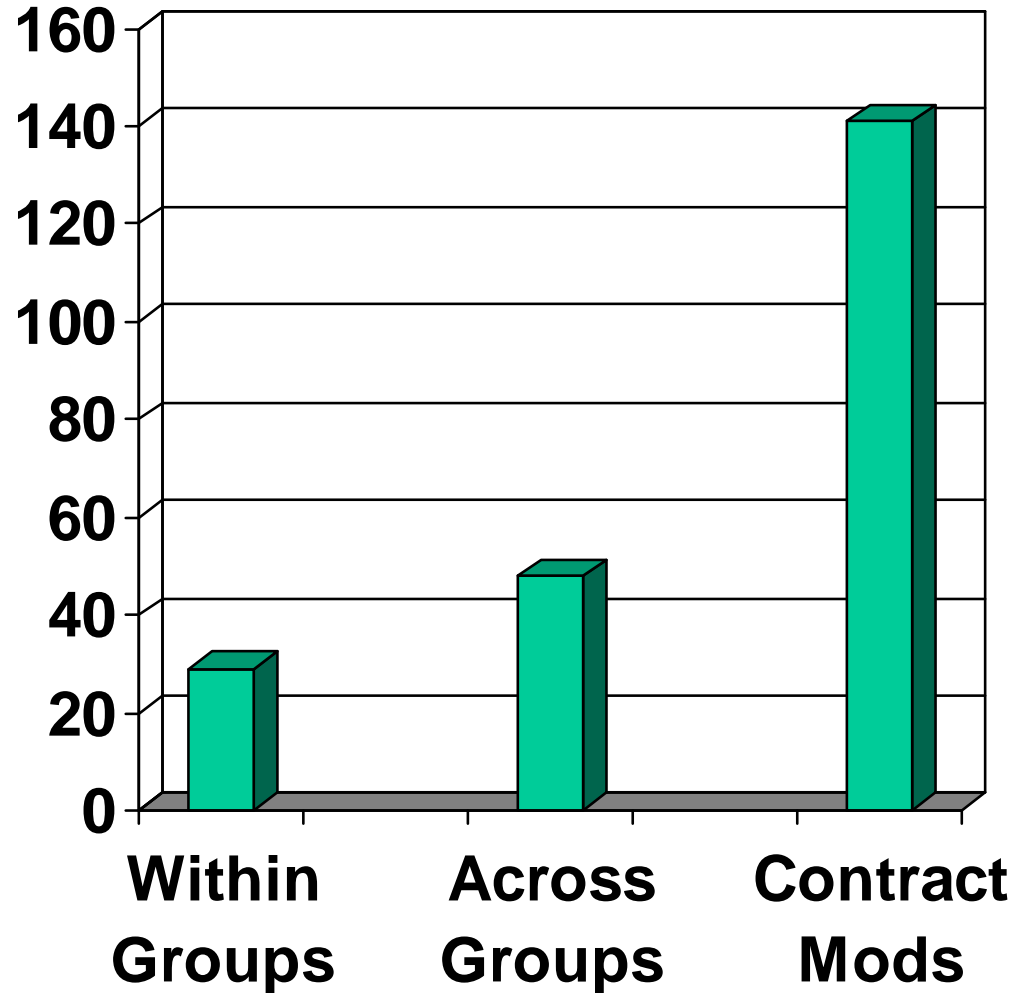
Processes and tools over individuals and interactions
Comprehensive documentation over working software
Contract negotiation over customer collaboration
Following a plan over responding to change

An Auditor Manifesto?



Average Change Processing Time: 2 Systems of Systems

- Average number of days to process changes





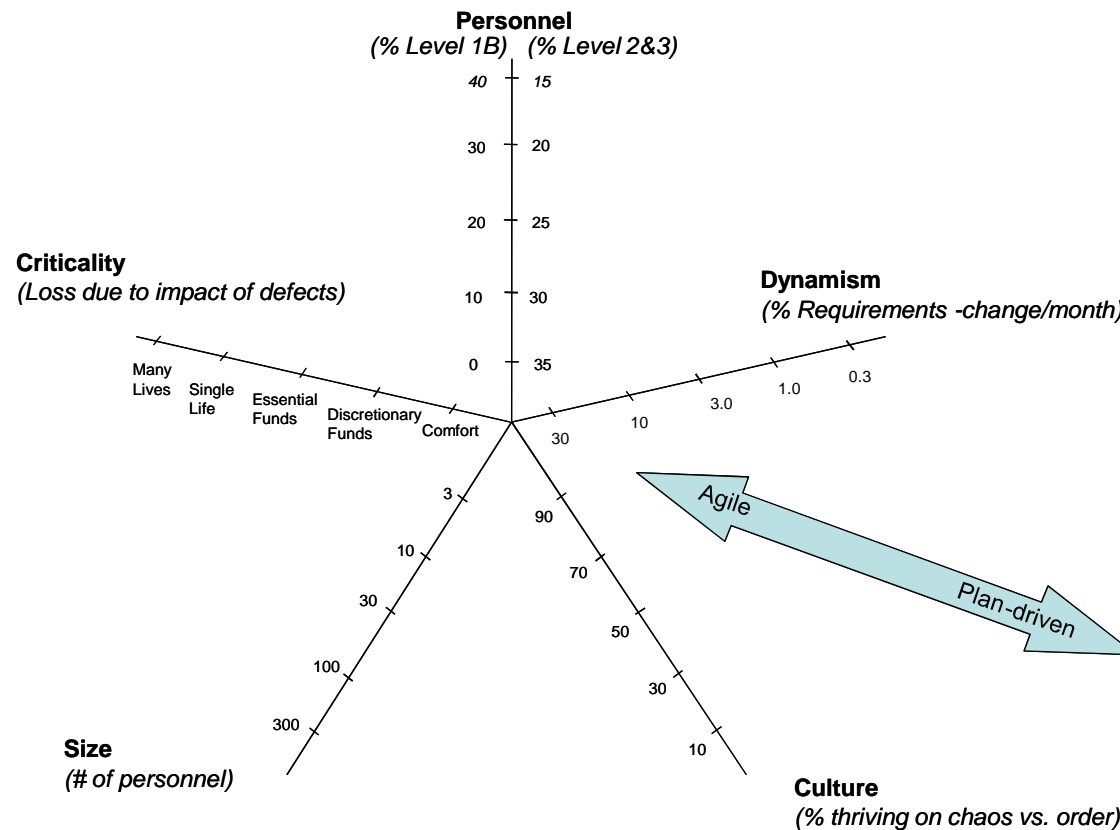
Agile Methods and Quality

- **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



Agile and Plan-Driven Home Grounds: Five Critical Decision Factors

- Size, Criticality, Dynamism, Personnel, Culture

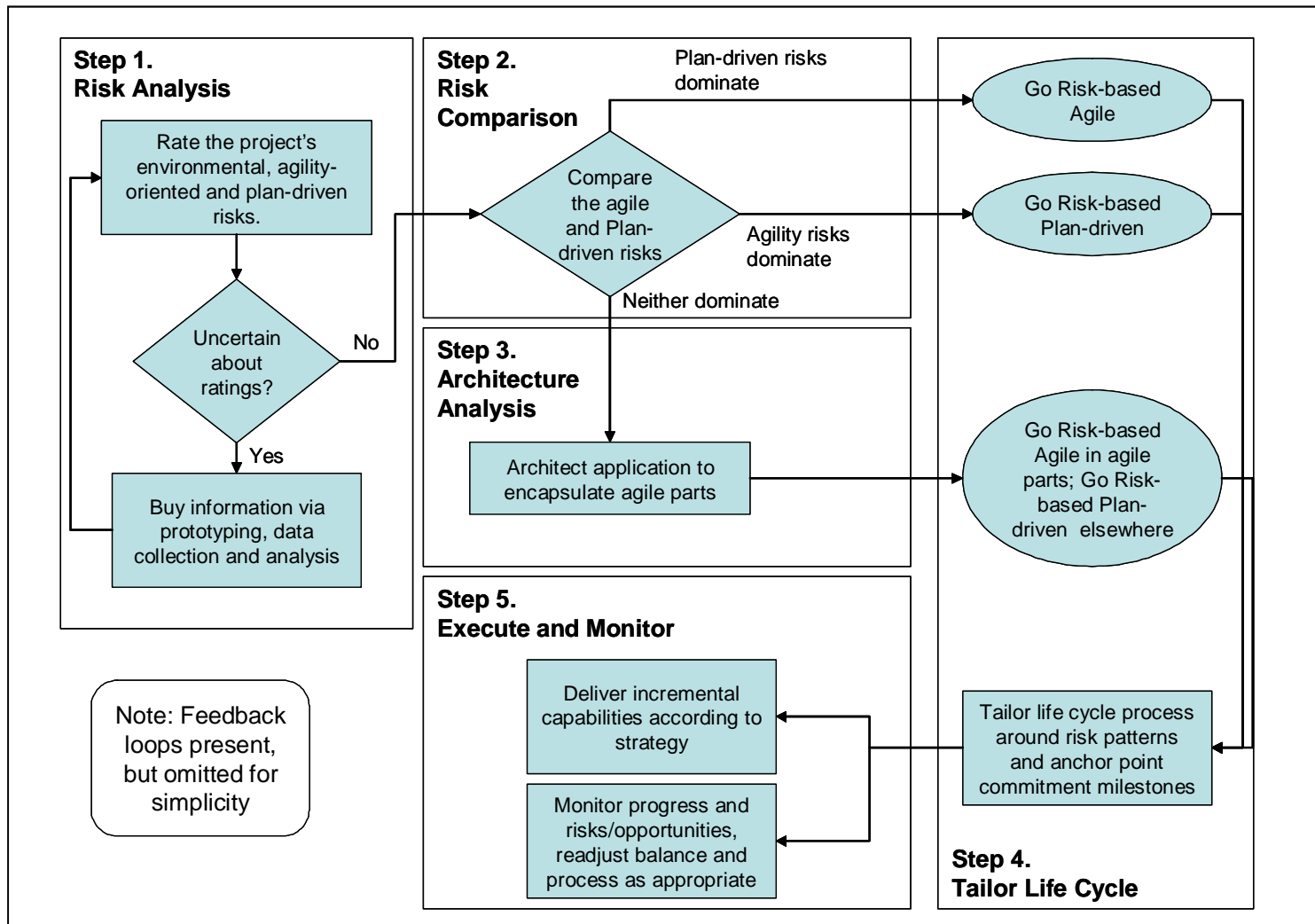




Outline

- **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**

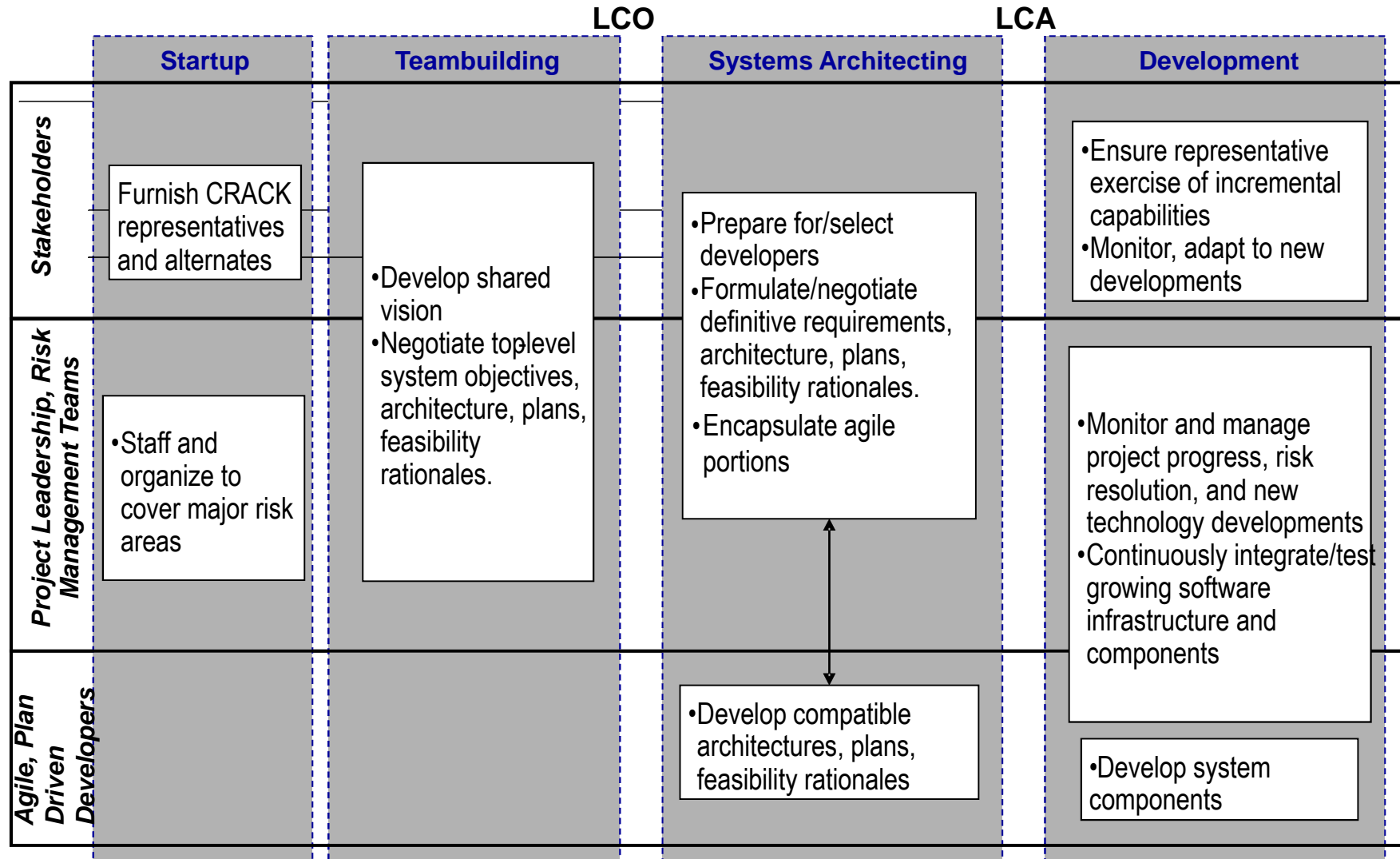
Using Risk to Balance Discipline and Agility - Overview





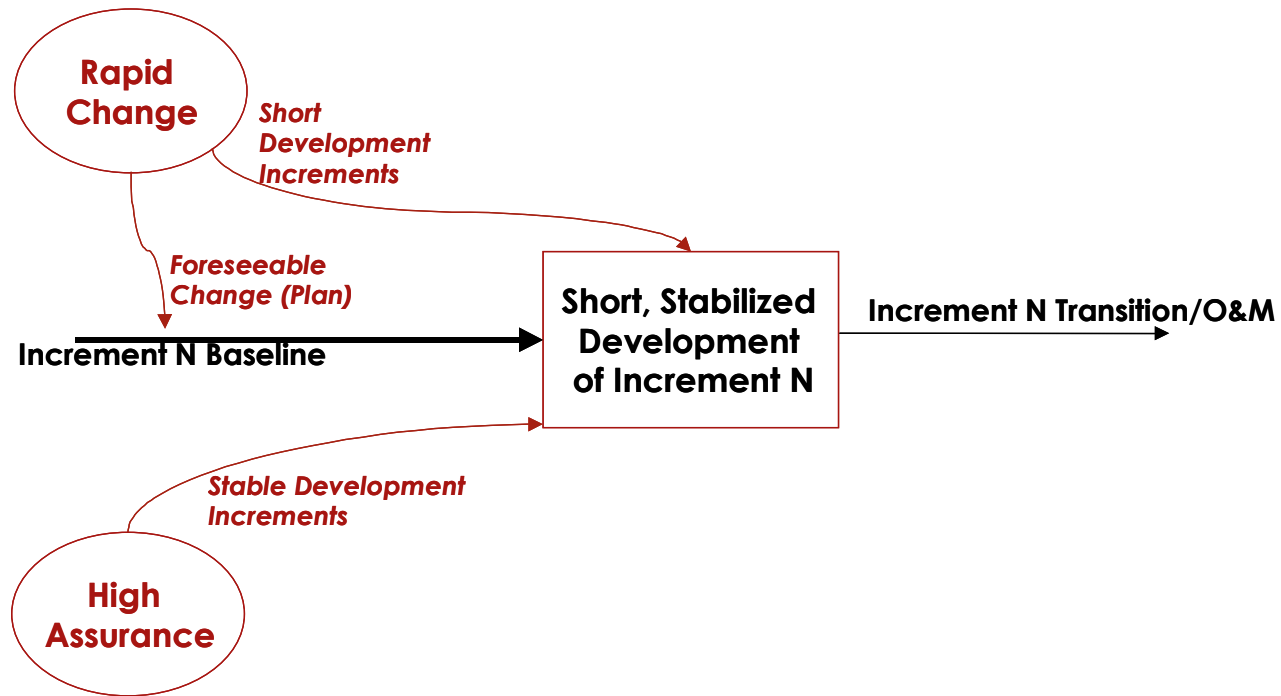
Hybrid Agile/Plan-Driven Strategy

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



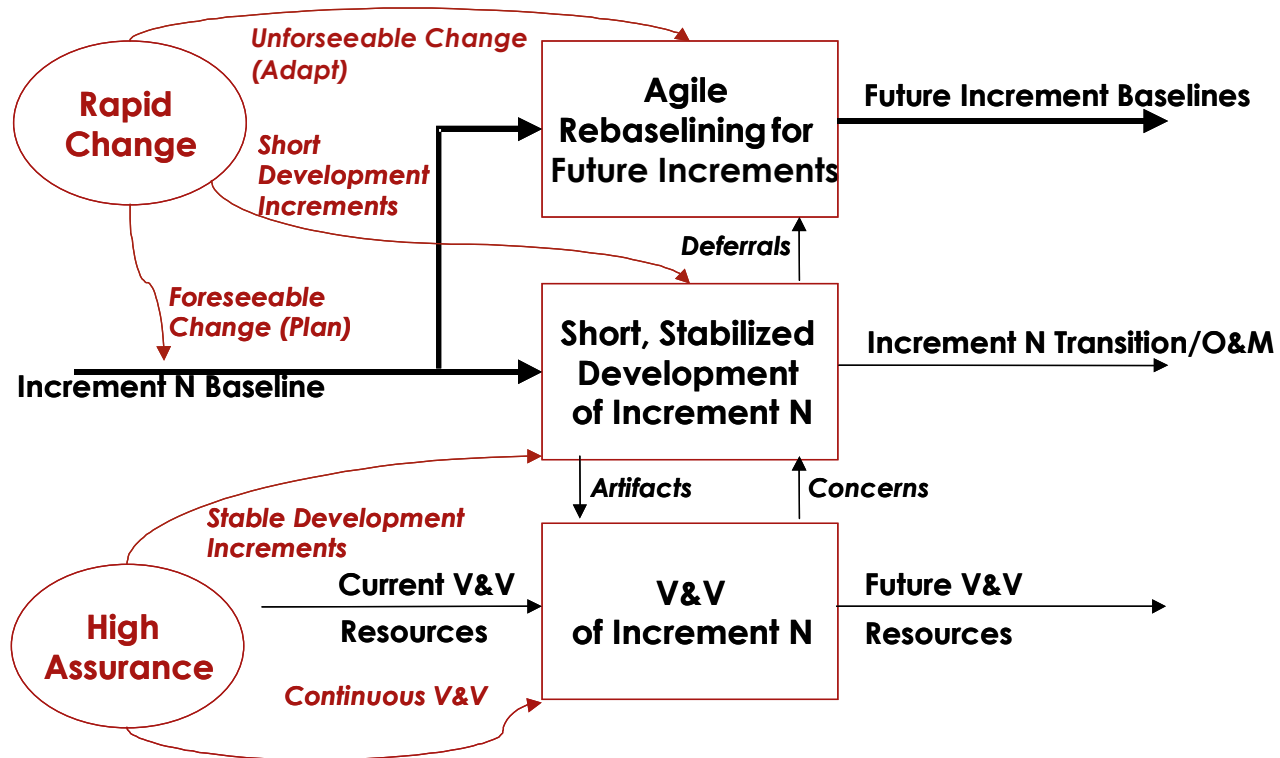


Incremental Commitment Model: Single Increment View





Incremental Commitment Model: Single Increment View





Milestone 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**
 - **Generate a viable return on investment**
 - **Generate satisfactory outcomes for all of the success-critical stakeholders**
- **All major risks resolved or covered by risk management plans**
 - **Serves as basis for stakeholders' commitment to proceed**



Case Studies and Critical Success Factors

- **Diversified, USA (AgileTek)**
- **Medical, USA**
- **Enterprise Resource Planning, Germany**
- **Enterprise Infrastructure, Germany**



Agile Tek and *Agile+*

- Agile+ is XP + ...
 - + Business Process Analyses (BPAs)
 - + Story “Actors”
 - + Delphi-STE Estimation
 - + Risk-Based Situation Audits (RBSAs)
 - + Componentized Architecture
 - + Wall Gantts 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 quality 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

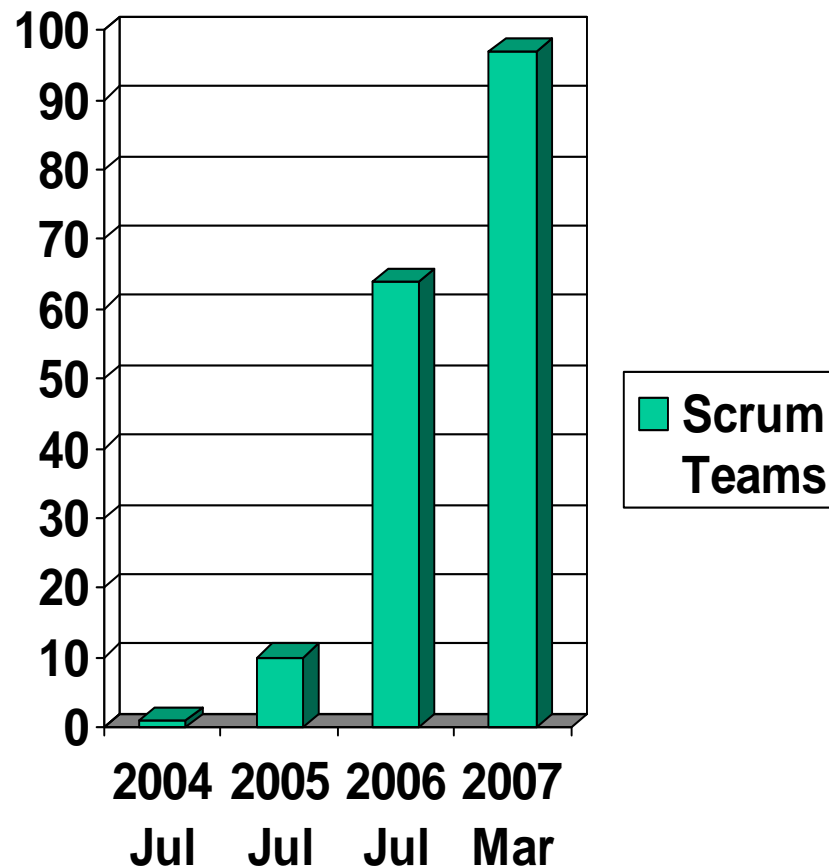


Medical Case Study -- USA

- **1400 software people; 7M SLOC; 7 sites**
 - 4 in Europe, 2 in India
- **500 medical applications; 500 financial; others**
- **Survivability-critical software problems**
 - Reliability, productivity, performance, interoperability
 - Sarbanes-Oxley requirements
 - Management receptive to radical change
- **Some limited experimental use of agile methods**
 - Led by top software technologist/manager
- **Committed to total change around Scrum and XP**



Medical-USA Adoption Profile



- **July 2004 - July 2005**
 - Recruit top people from all sites into core team(s)
 - Get external expert help
 - Develop architecture
 - Early Scrum successes with infrastructure
 - Revise policies and practices
 - Train, reculture everyone
 - Manage expectations
- **July 2005 – July 2006**
 - Begin full-scale development
 - Core teams as mentors



Key Practices – USA Medical

- **Include customers and marketers**
 - New roles; do's/don'ts/opportunities; CRACK personnel; full collaboration and teamwork; expectations management
- **Scrum; most XP practices; added company practices**
 - 6-12 person teams with team rooms, dedicated servers
 - Hourly smoke test; nightly build and regression test
 - Just-in-time analysis; story-point estimates; fail fast; detailed short-term plans; company architecture compliance
 - Embrace change in applications and practices
 - Global teams: wikis, daily virtual meetings, act as if next-door
- **Release management**
 - 2-12 week architecting Sprint Zero; 3-10 1-month Sprints; Release Sprint; 1-6 month beta test
 - Next Sprint Zero concurrent with Release Sprint
- **Initiative manager and team**
 - Define practices; evolve infrastructure; provide training; guide implementation; evaluate compliance/usage; continuous improvement

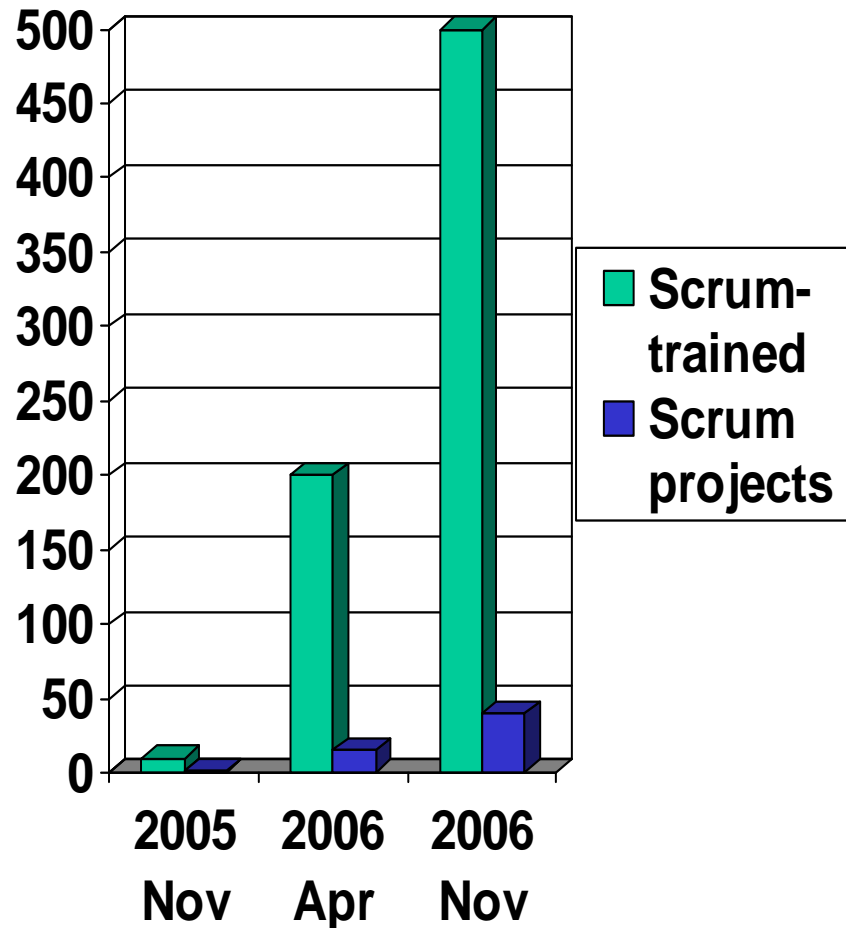


ERP Case Study -- Germany

- **35,000 employees; 32,000 customers worldwide**
 - Major development centers in India, Israel
- **Need to improve software development productivity, adaptability of Product Innovation Life Cycle**
 - Invent, Define, Develop, Deploy, Optimize
- **Proposed agile development cycle**
 - Scrum; tailored corporate practices; strong, >70%-time Product Owner and Scrum Master
 - 10-person teams best; up to 3-team Scrum of Scrums
 - Release cycle similar to Medical-USA
- **Strong, highly experienced agile initiative leader**
 - With top management support



ERP-Germany Adoption Profile



- **Two large projects**
 - 8 teams, 2 countries
 - 5 teams, 5 Product Owners, 3 countries
- **Mentoring new teams**
 - Pre-training
 - Experienced Scrum Master
 - Mentor first sprint
 - Coach second sprint
- **Benefits: visibility, communication, productivity**



Challenges and Lessons Learned

- **Challenges**
 - Keeping roles straight
 - Setup for large projects
 - Rebaselining, reprioritizing backlog
 - Cross-cultural training, jelling
 - Team learning culture
- **Lessons learned**
 - Need strong Product Owner and Scrum Master
 - Training for Product Owner, other stakeholders
 - Especially for scaling up to multiple teams
 - Reinforce, evolve common corporate Scrum process
 - Can't neglect CM, version control, change management
 - Of applications and process



Corporate Infrastructure -- Germany

- **Fortune World 100 company**
- **Global development**
 - Especially US, India, China
- **Need to rebaseline corporate infrastructure**
 - Business processes, services, IT infrastructure
 - Multi-platform portability, always on
 - With worldwide participation and buy-in
 - Strategy: RUP/Spiral architecting; Scrum-based development
- **Began with multi-site core team of top personnel**
 - Led by strong, results-oriented project/technology leader
 - With top management support and backing
- **Grown to 4 sites, 250 people, 24 teams in 2.5 years**
 - Largest project: 10 teams of 10-person Scrums



Corporate Infrastructure: Principles

- **Service-oriented architecture**
 - **Business processing: IBM, SAP, Microsoft**
 - **Generic applications: phone, Web, user interface**
 - **Infrastructure: servers, gateways, networks**
- **Learn form others**
- **Select, protect best team**
- **Inclusive stakeholder communication and involvement**
- **Plan for early wins**
- **Corporate product and process framework with explicit tailoring areas, continuous improvement**



Development Practices

- **RUP/Spiral startup**
 - 2 Inception, 3 Elaboration, 4 Construction cycles
 - Proposal bay with wall stickies for risk prioritization
 - 30 top people from all 4 sites
 - NetMeeting for remote office involvement
 - SharePoint vs. heavy documentation
 - Dedicated specialists (architecture, performance, test)
- **Initial Operational Capability development**
 - Timeboxed sprints with prioritized requirements
 - 30% of initial requirements dropped to admit new features
 - Use backlog as management instrument
 - Post-IOC release sprint for documentation, installation, training
 - Followed by field test, concurrent detailed expansion planning, return of offsite core team members to lead distributed-development scaleup



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

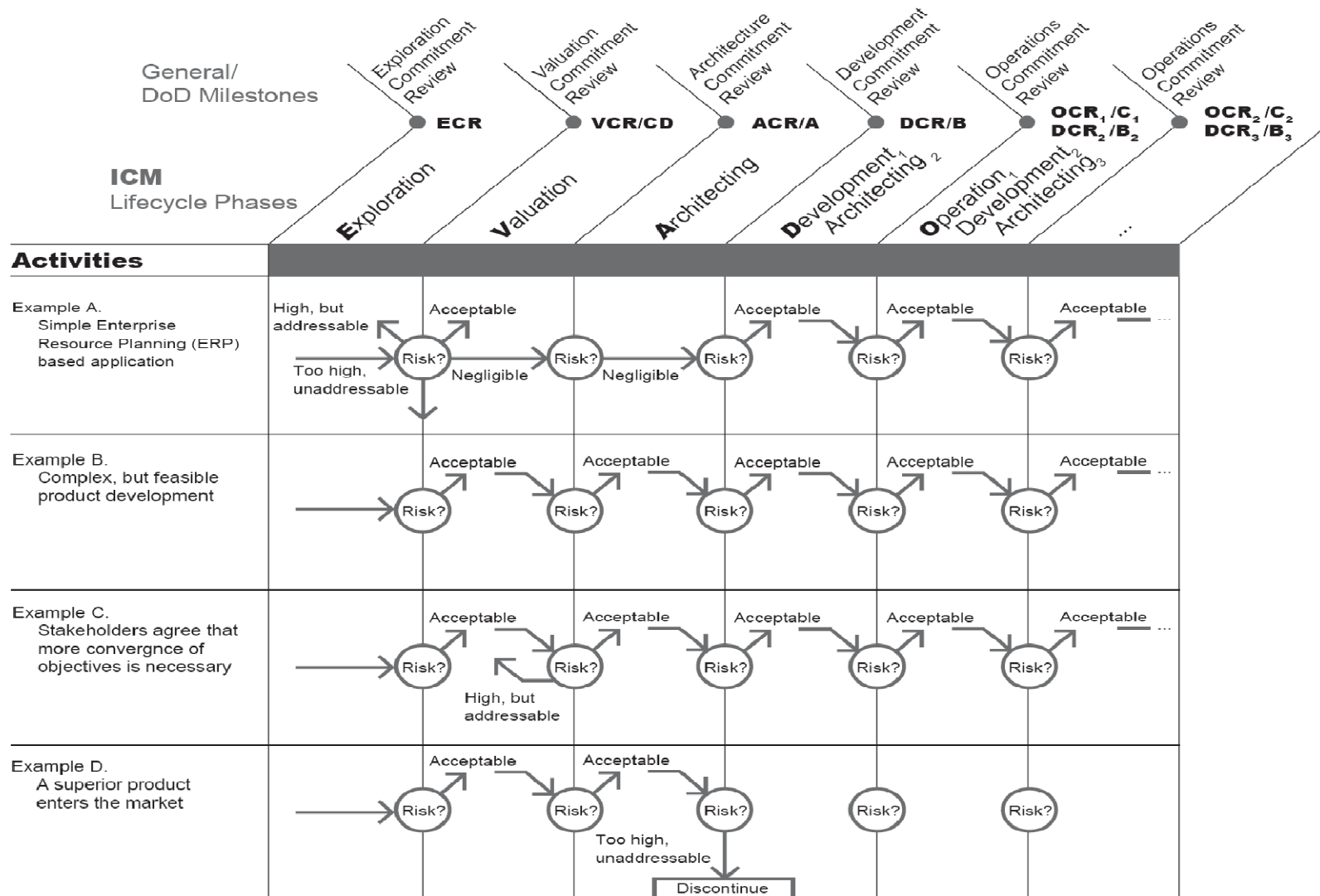
- **Success-critical to achieve both agility and quality**
- **Hybrid agile/plan-driven methods emerging**
 - Incremental commitment framework
 - Concurrent engineering with synchronization milestones
 - Scrum plus organizational essentials
- **Success stories emerging**
 - Management commitment to objectives 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



Backup Charts



Different Risk Patterns Yield Different Processes





ICM HSI Levels of Activity for Complex Systems

