IBM Software Group

Agile Strategies for Enterprise Architects

IBM agility@scale™

Rational. software

Purpose of this Module

- This module overviews our thinking around agile strategies for enterprise architecture
- It includes results from a recent EA survey
- This can be stand alone or part of a larger training offering
- Use this for both internal and customer-facing purposes
- This is a hidden slide, feel free to remove it

Agenda

- Some industry statistics
- Disciplined agile delivery
- Agile architecture strategies
- Agile enterprise architecture strategies
- Scaling agile
- Parting thoughts

Agenda

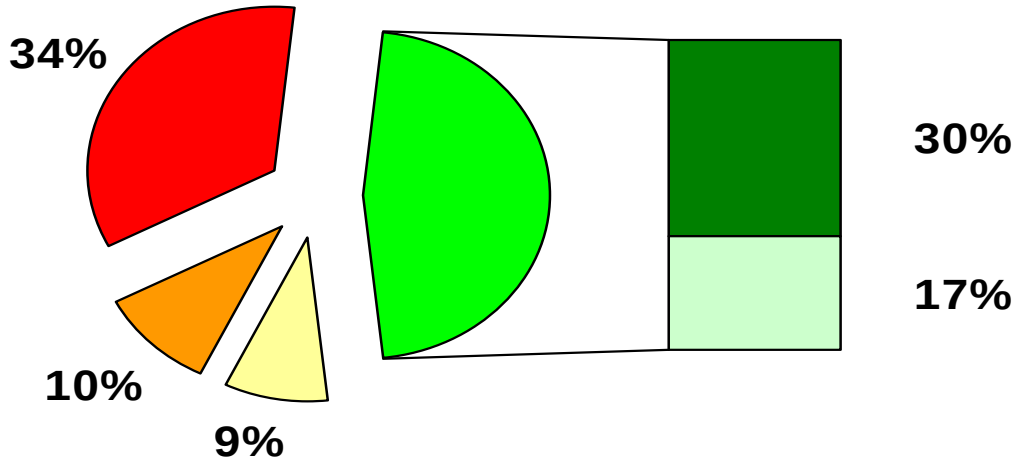
- Some industry statistics
- Disciplined agile delivery
- Agile architecture strategies
- Agile enterprise architecture strategies
- Scaling agile
- Parting thoughts

Dr Dobb's January 2010 State of the IT Union Survey

- Last week of January and all of February 2010
- Survey link included in:
  - January 2010 DDJ Agile Newsletter
  - Jon Erickson's blog at www.ddj.com
  - www.ambysoft.com/surveys/ page
  - Posting to ambysoft@yahoogroups.com
- Data, summary, and slides downloadable from www.ambysoft.com/surveys/
- 374 respondents
  - 38% were developers, 27% were in management
  - 80% had 10+ years in IT
  - 28% worked in orgs of 500+ IT people
  - 66% North American, 21% European, 10% Asia Pacific

Source: Dr Dobb's January 2010 State of the IT Union Survey

What best describes the current state of your EA program?



Enterprise architecture isn't as widespread as we would hope.

What are YOU going to do about that?

Source: Dr Dobb's January 2010 State of the IT Union Survey

What are/were the goals for the EA Program? (Multiple selections allowed)

- 53% Promote common technical infrastructure
- 51% Business efficiency/transformation
- 50% Reduce operating costs
- 49% Support system integration
- 48% Improve technical integrity
- 47% Improve enterprise decision making
- 44% Improve IT governance
- 41% Improve data integrity
- 33% Improve risk management
- 32% Reduce technical complexity
- 31% Ensure continuity of organizational knowledge
- 30% Reduce waste
- 29% Improve business governance
- 16% Increase effectiveness of audit compliance
- 11% Support multi-vendor projects
- 10% Support outsourcing initiatives

Each organization has unique goals. One EA process does not fit all.

Source: Dr Dobb's January 2010 State of the IT Union Survey

The artifacts (being) produced by the EA program include (Multiple selections allowed)

- 67% Definition of business goals/drivers/objectives
- 65% An inventory/list of existing systems
- 64% Architecture principles for development teams
- 55% Development guidelines
- 44% Reference architectures (examples)
- 38% Current state models
- 33% "To be" models
- 29% White papers/position papers

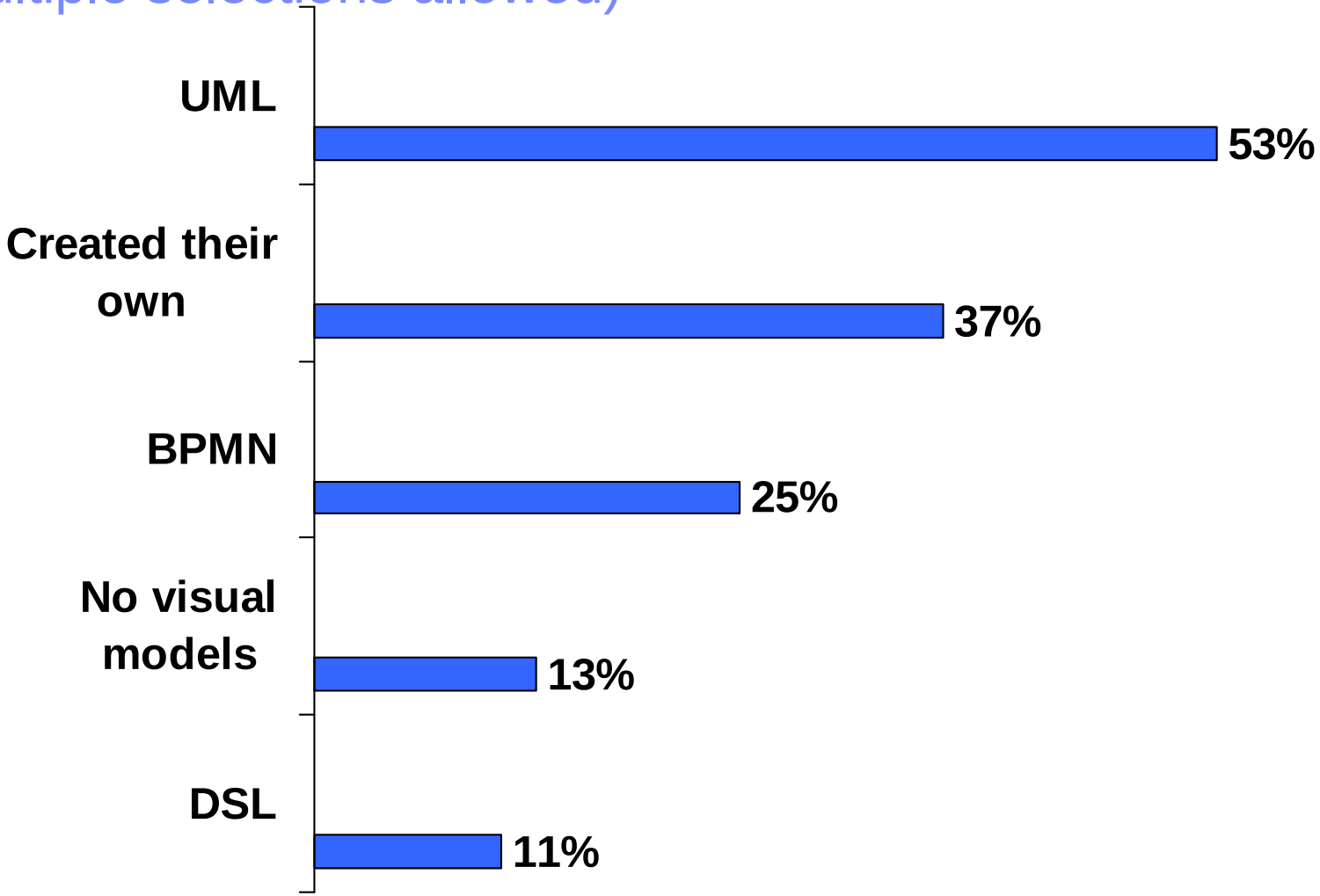
Source: Dr Dobb's January 2010 State of the IT Union Survey

The types of models (being) produced by the EA program include (Multiple selections allowed)

- 65% Business architecture model
- 56% High-level conceptual data model
- 51% Enterprise business process model
- 48% Deployment models
- 45% Component model
- 35% Security models
- 35% Detailed enterprise data model (EDM)
- 30% Enterprise use case model

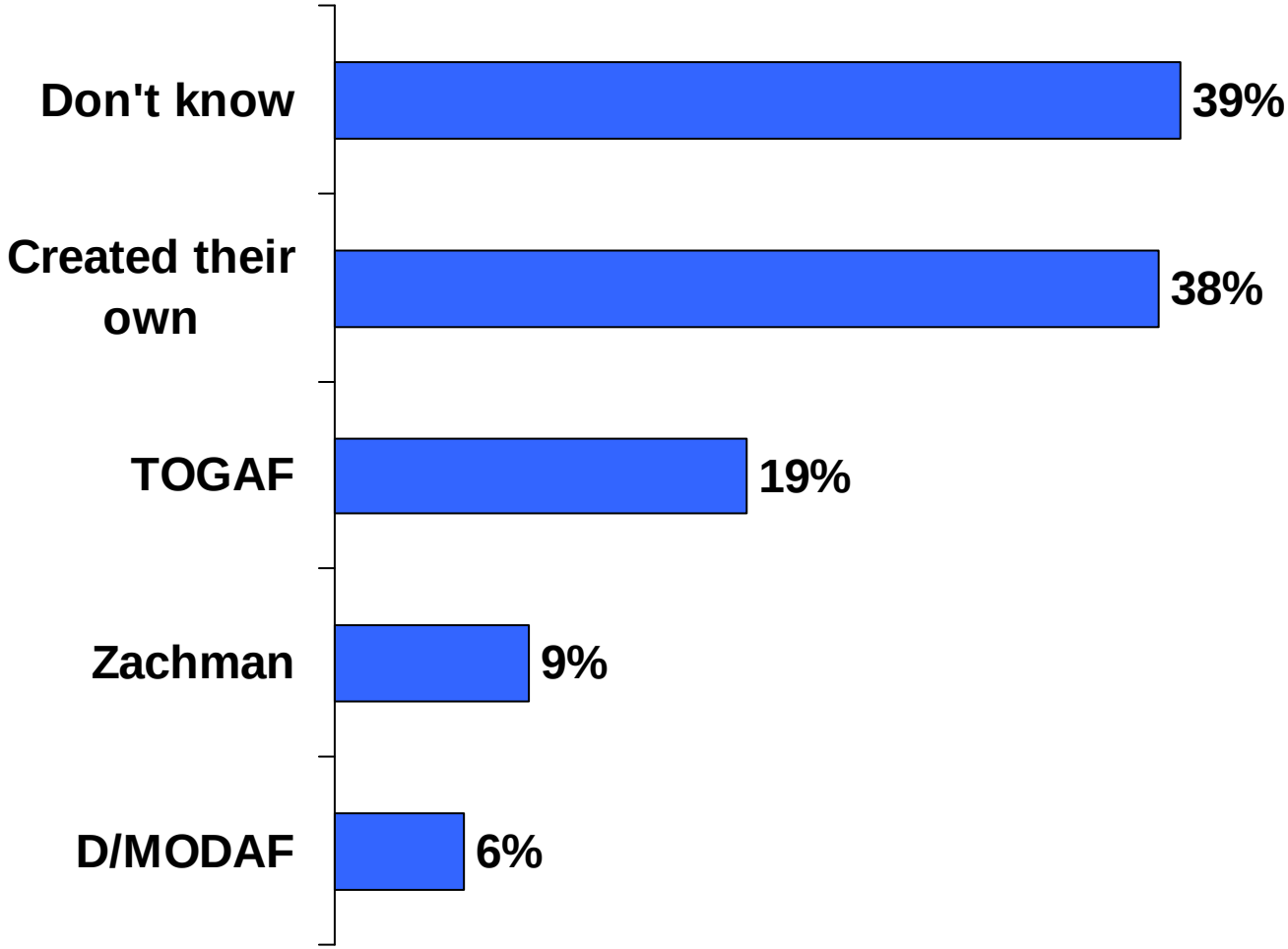
Source: Dr Dobb's January 2010 State of the IT Union Survey

Which modeling notations does/did the EA apply? (Multiple selections allowed)



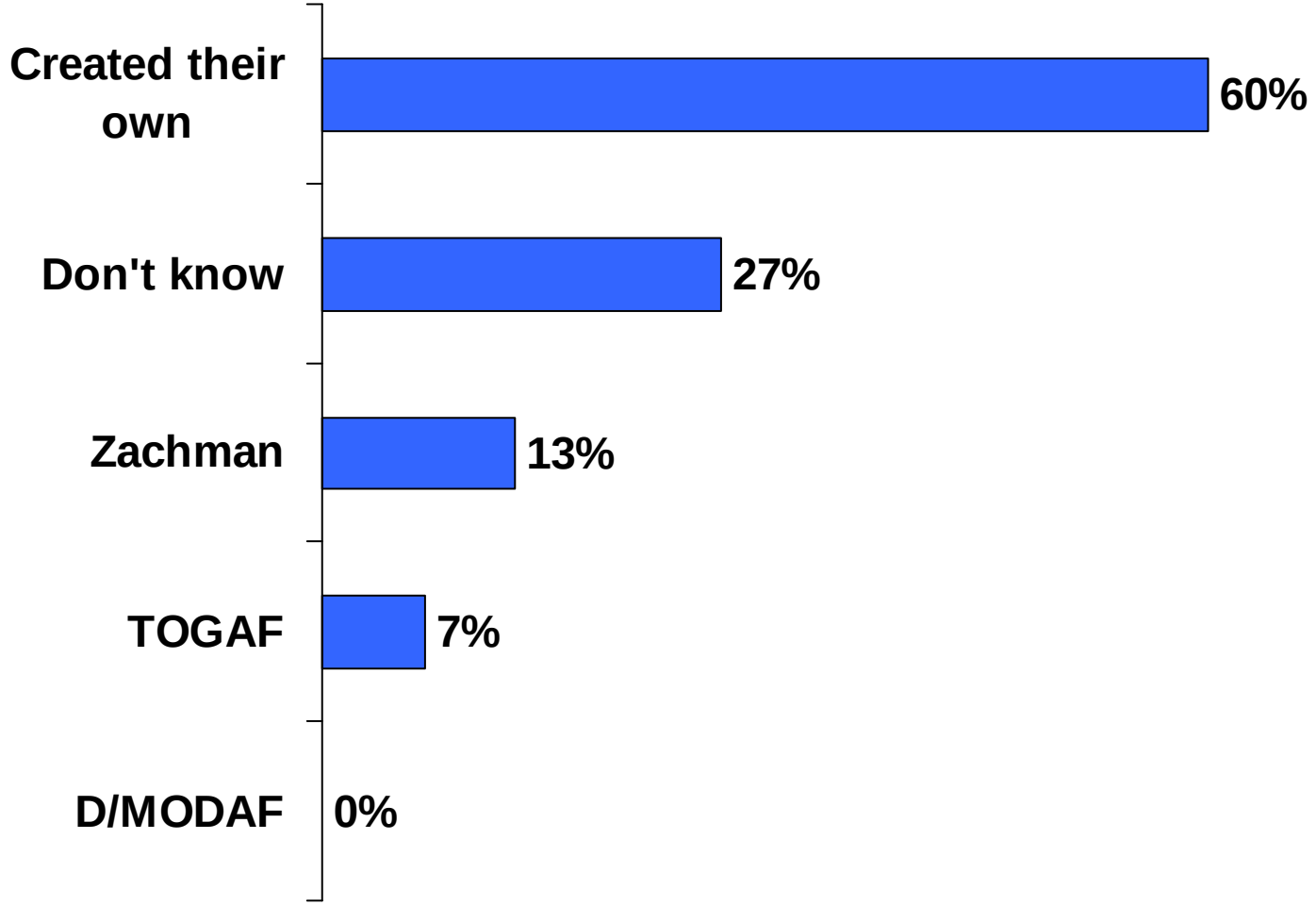
Source: Dr Dobb's January 2010 State of the IT Union Survey

Which EA frameworks, if any, did your (successful) EA program apply? (Multiple selections allowed)



Source: Dr Dobb's January 2010 State of the IT Union Survey

Which EA frameworks, if any, did your (unsuccessful) EA program apply? (Multiple selections allowed)



Source: Dr Dobb's January 2010 State of the IT Union Survey

The technology strategy captured by the EA includes (Multiple selections allowed)

- 65% Service Oriented Architecture (SOA)
- 55% Common Frameworks
- 52% Business process management (BPM)
- 43% Components
- 37% Software as a Service (SAAS)
- 31% Product Line Architecture
- 22% Cloud Computing
- 14% Semantic Architecture

Are we a fashion industry?

- Ivar Jacobson

Source: Dr Dobb's January 2010 State of the IT Union Survey

For existing Enterprise Architecture (EA) programs, what has improved? (Rating between -10 and +10)

1. System integration (3.6)
2. IT governance (3.3)
3. Team follows common technology infrastructure (3.3)
4. Business efficiency (3.2)
5. Data integrity (3.2)
6. Continuity of organizational knowledge (3.0)
7. Business governance (3.0)
8. Audit compliance (2.9)
9. Risk management (2.9)
10. Technical integrity (2.8)
11. Operating costs (2.5)
12. Enterprise decision making (2.5)
13. Reduction of waste (2.3)
14. Support for multi-vendor projects (1.8)
15. Outsourcing initiatives (1.3)
16. Reduction of technical complexity (0.8)

The EA reality doesn't seem to match the EA rhetoric

Source: Dr Dobb's January 2010 State of the IT Union Survey

For existing Enterprise Architecture (EA) programs, what were the importance of success factors/strategies? (Rating between -10 and +10)

1. Active involvement of business leaders (5.8)
2. Active involvement of IT leaders (5.7)
3. Enterprise architects are active participants on project teams (5.5)
4. Enterprise architects are trusted advisors of the business (5.5)
5. Flexible enterprise architects (5.1)
6. Having a business case for EA efforts (4.5)
7. Continuous improvement/evolution of EA artifacts (4.5)
8. Architecture reviews (4.1)
9. Appropriate governance (4.1)
10. Cost reduction (3.5)
11. Master data management (MDM) (2.8)

"People issues" appear to be the primary drivers of success

Source: Dr Dobb's January 2010 State of the IT Union Survey

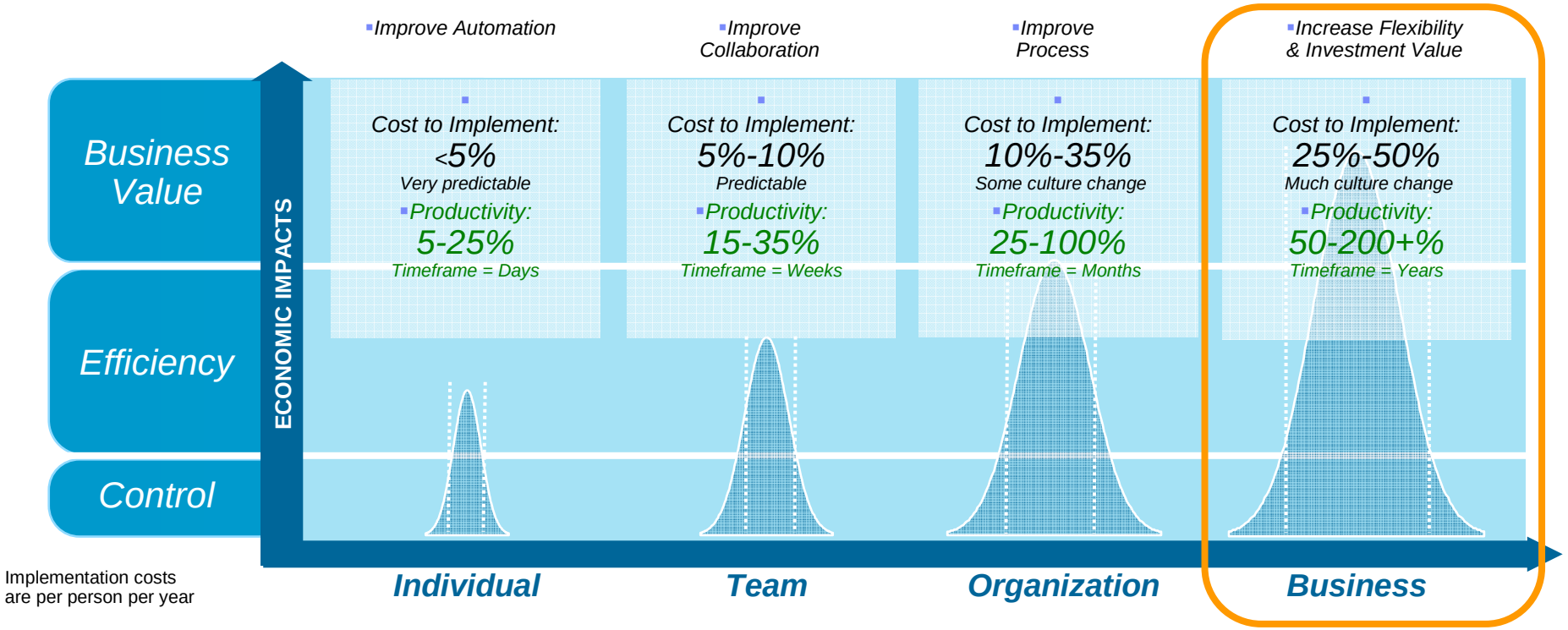
For cancelled Enterprise Architecture (EA) programs, why was it ended? (Rating between -10 and +10)

1. Insufficient time provided (3.3)
2. Project teams didn't take advantage of the EA (3.2)
3. Too difficult to measure benefits (2.5)
4. Enterprise architects perceived as "ivory tower" (2.5)
5. Development teams couldn't wait for enterprise architects (2.5)
6. No perceived benefit of EA program (2.0)
7. No executive endorsement (1.7)
8. Enterprise architects weren't sufficiently flexible (1.5)
9. Enterprise architects perceived as impediment to success (1.5)
10. Insufficient funding (1.5)
11. EA perceived as not viable (0.0)
12. Cancelled due to political issues (-0.6)
13. EA program successful but terminated (-1.9)

EA failure is often due to “overpromising and under-delivering” or to “people issues”

Source: Dr Dobb’s January 2010 State of the IT Union Survey

Invest across the spectrum of improvement to manage risks and optimize business outcomes



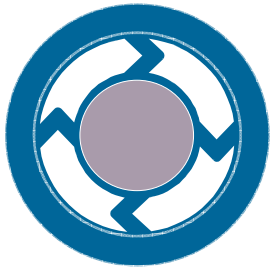
Focusing on business outcomes offers the greatest return on investment

Source: IBM Rational

Agenda

- Some industry statistics
- Disciplined agile delivery
- Agile architecture strategies
- Agile enterprise architecture strategies
- Scaling agile

Agile Scaling Model (ASM)



Core Agile Development

- Focus is on construction
- Goal is to develop a high-quality system in an evolutionary, collaborative, and self-organizing manner
- Value-driven lifecycle with regular production of working software
- Small, co-located team developing straightforward software

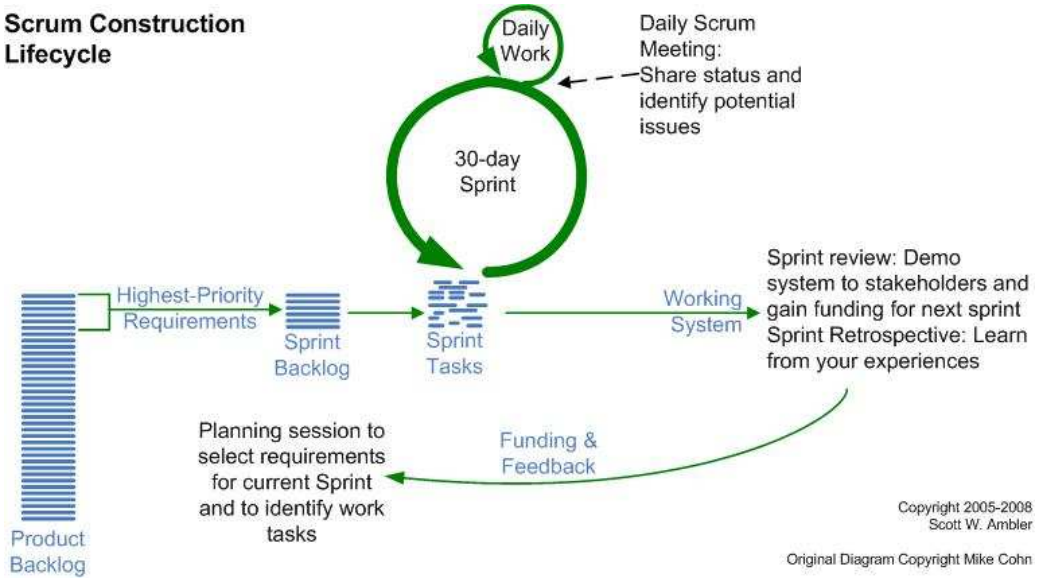
Disciplined Agile Delivery

- Extends agile development to address full system lifecycle
- Risk and value-driven lifecycle
- Self organization within an appropriate governance framework
- Small, co-located team delivering a straightforward solution

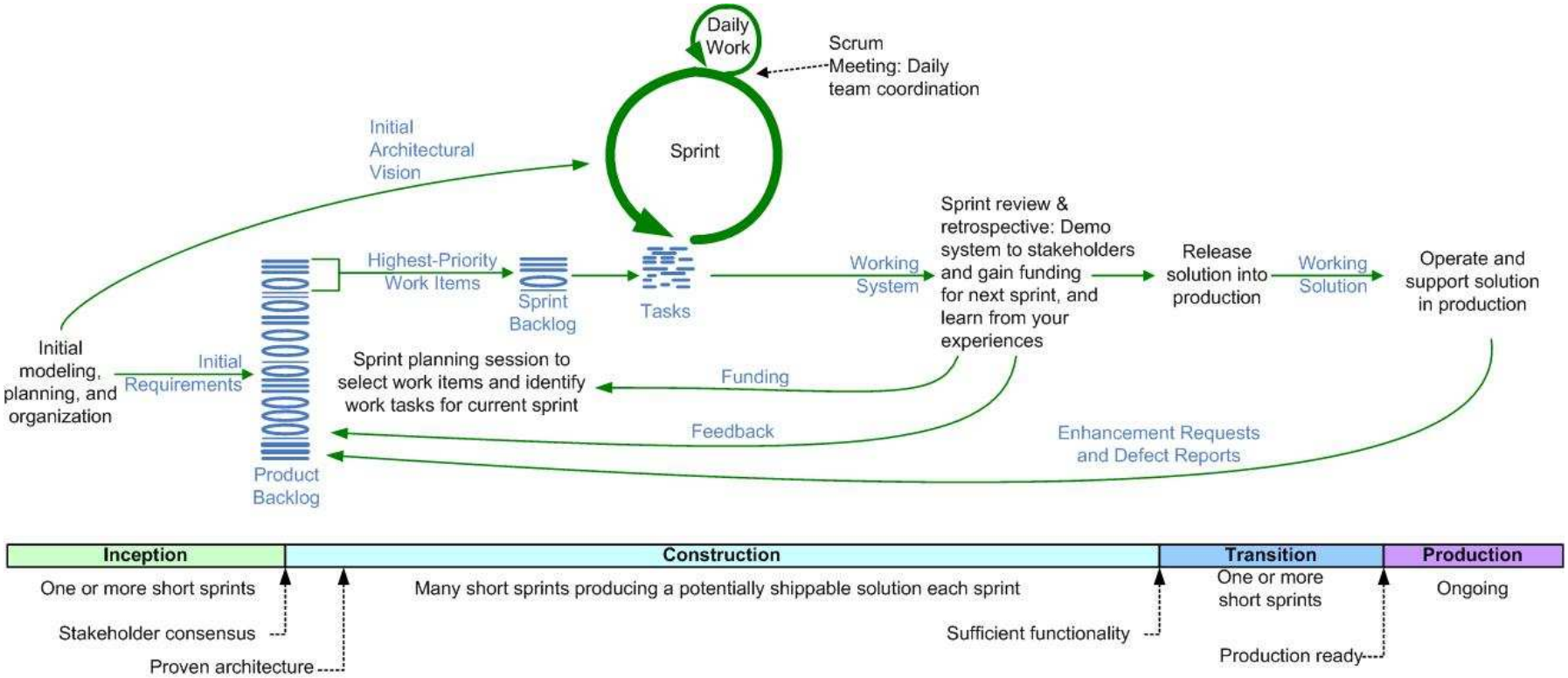
Agility at Scale

- Disciplined agile delivery and one or more scaling factors applies

The agile construction lifecycle



The disciplined agile delivery life cycle



What is disciplined agile delivery (DAD)?

Disciplined agile delivery is an evolutionary (iterative and incremental) approach that regularly produces high quality solutions in a cost-effective and timely manner via a risk and value driven lifecycle.

It is performed in a highly collaborative, disciplined, and self-organizing manner within an appropriate governance framework, with active stakeholder participation to ensure that the team understands and addresses the changing needs of its stakeholders.

Disciplined agile delivery teams provide repeatable results by adopting just the right amount of ceremony for the situation which they face.

Core Principles

- "Fits just right" process
- Continuous testing and validation
- Consistent team collaboration
- Rapid response to change
- Ongoing customer involvement
- Frequent delivery of working solutions

Agenda

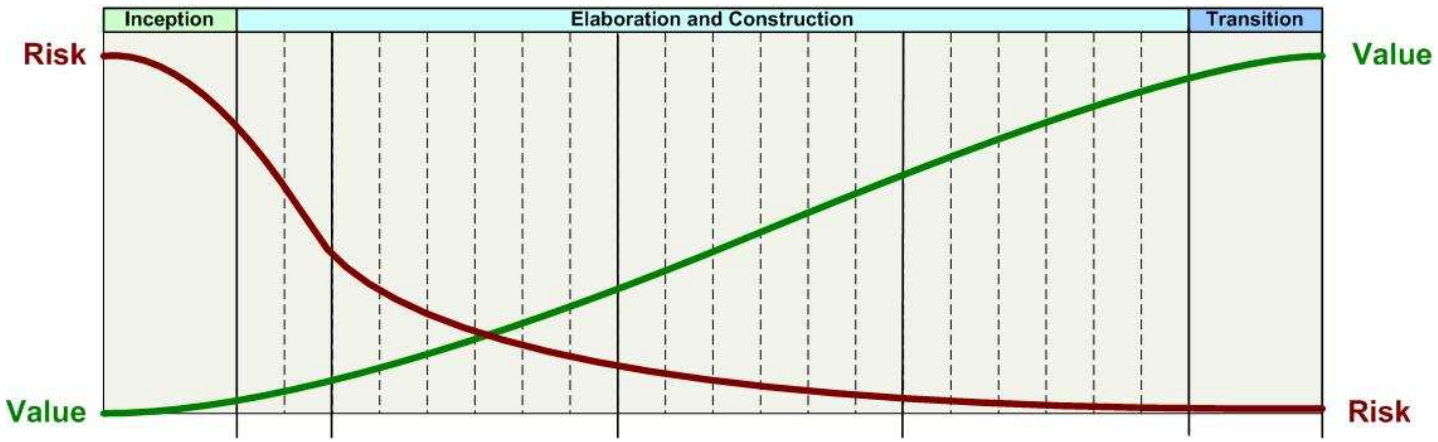
- Some industry statistics
- Disciplined agile delivery
- Agile architecture strategies
- Agile enterprise architecture strategies
- Scaling agile
- Parting thoughts

Look Beyond Technology

- You need to understand the business
  - Architecture must be based on requirements, otherwise you are hacking
  - There are two aspects to architecture, business and technical
- Individuals and interactions
  - Supplying models and documents isn't sufficient
  - Support project teams
  - Roll up your sleeves and work closely with the teams
  - Architecture comes from teams, not individuals

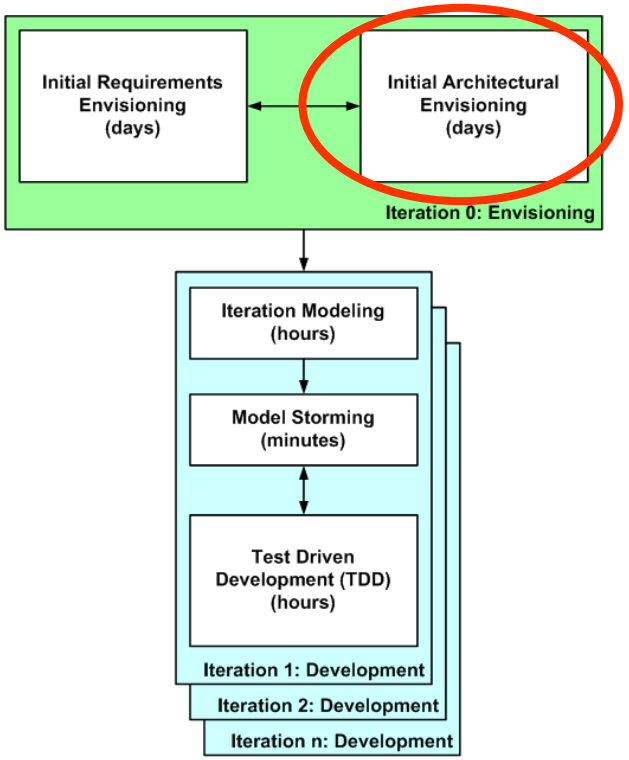
Prove the Architecture With Working Code

- Everything looks like it will work on whiteboards or pretty architecture diagrams
- It's not until you've built a working end-to-end skeleton of the system which addresses your major technical risks do you know that your architecture really works
- The Unified Process's Elaboration phase explicitly focuses on reducing technical risk on a project by proving the architecture with code



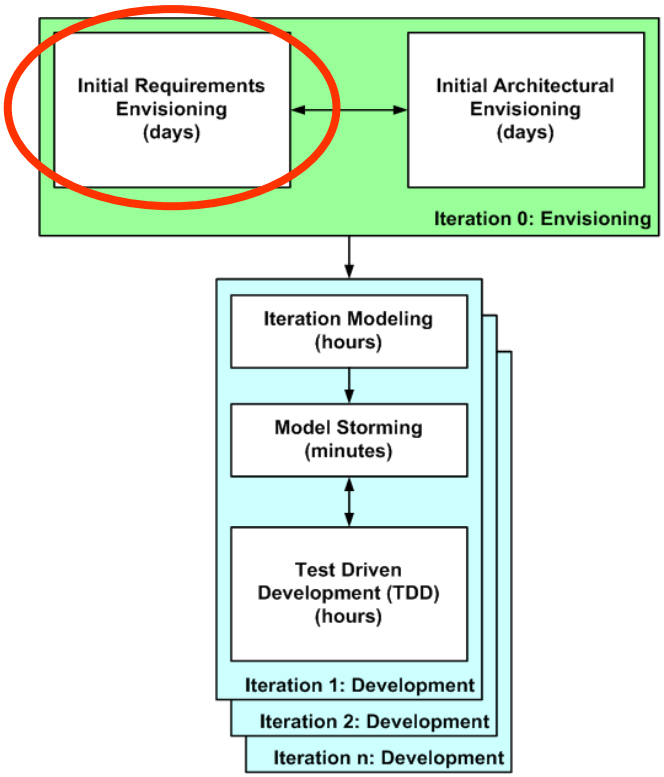
Initial Architecture Envisioning

- Your goals are to:
  1. Identify and agree to a potential initial architecture of your system
  2. Provide sufficient technical vision for estimating and scheduling concerns
- Critical models for business application development:
  - Some form of deployment diagram
  - A free-form “technology stack” diagram
  - A UI flow diagram



Base Your Architecture on Requirements

- Your goals are to:
  1. Identify and agree to the initial scope of your project
  2. Develop the initial stack of requirements
  3. Gather enough information to address initial scheduling and estimating concerns
- Critical models for business application development:
  - Some sort of usage model (use cases, user stories, …)
  - Conceptual/domain model
  - Some UI sketches



Architectural “spikes”

- Sometimes your team will work with a technology they are unfamiliar with
- You want to gain experience with that technology
  - You want to make informed decisions as to its application
  - There are always tradeoffs
  - There are always usage patterns which are effective and some which are not
  - The technology may not work well in your environment, for a variety of technical and cultural reasons
- Perform an architectural spike:
  - Write just enough code to explore the technology
  - This is typically “throw away” code
  - This is typically hours or days of effort
- Consider “bake offs”
  - If you’re considering several competing technical options, spike them all in parallel

Think About the Future, But Wait to Act

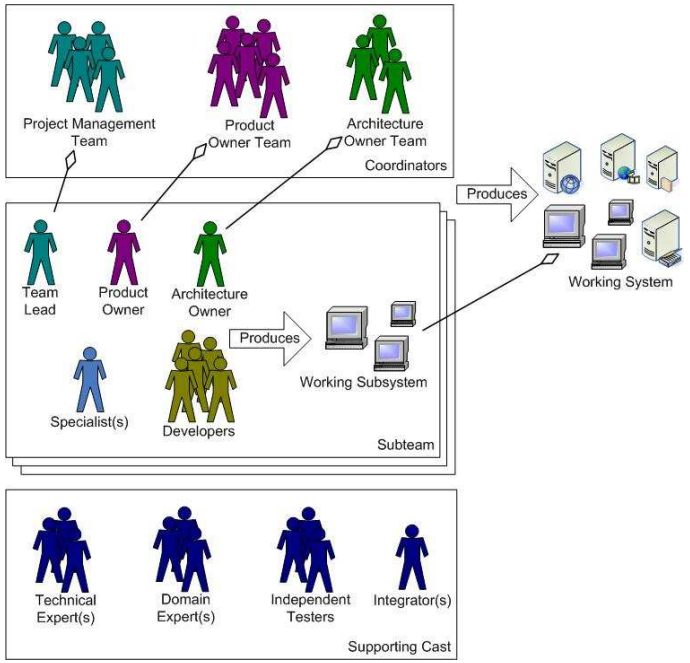
- Teams that focus on building frameworks, reusable components, … and other architecturally important foundations run the risk of being cancelled because they're not providing direct value to the business stakeholders
- The value of architectural envisioning is that it helps you to think through technical risks and provide a viable technical direction for your team
- Just because you've modeled it doesn't mean you need to build it right away
- By writing high-quality code, and by keeping it of high quality through refactoring, and by regular regression testing, it is safe to wait until you actually need an architectural feature to build it

Architects also code

- Agile software development has moved away from the traditional strategy of overly specialized people handing off artifacts to other overly specialized people to one of close collaboration between “generalizing specialists”
- A generalizing specialist is between the extremes of specialists, someone who knows a lot about a narrow topic, and generalists who know a little about a wide range of topics
- Architects that don’t code run the risk of:
  - Not understanding the underlying technologies
  - Not being respected by, or followed by, developers
  - Injecting serious defects which often prove costly to fix into a system
- But….
  - People with architecture-level skills may be rare
  - You may need to initially assign these people to multiple teams, clearly not an ideal strategy, and motivate them to focus on just architectural work and skills transfer

Architecture Owners, not Architects

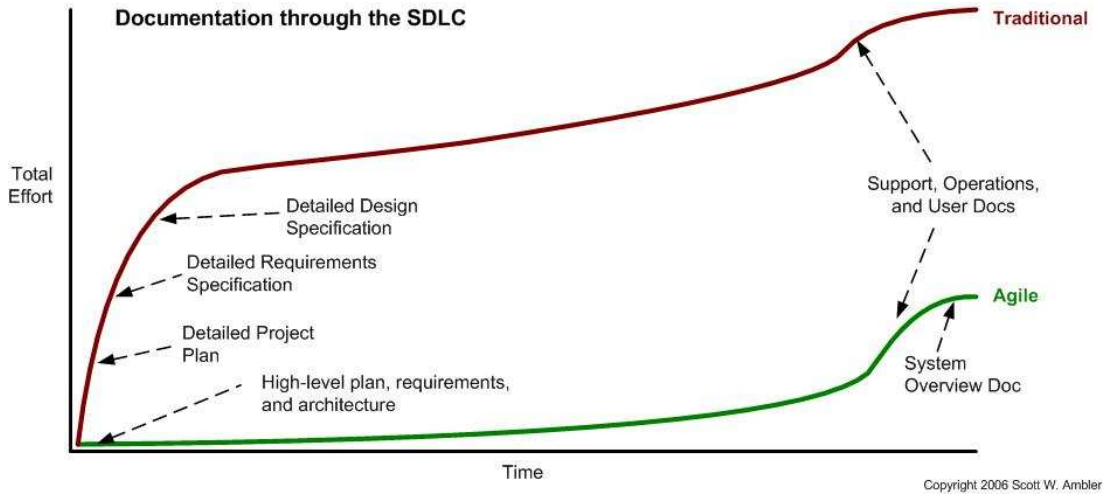
- An architecture owner is responsible for the architecture of the system or subsystems that the team is working on
- This person mentors and guides the developers in architectural issues, and leads them through technical issues
- This person understands the architectural direction and standards of their organization and helps to ensure that the team adheres to them appropriately
- This person is not solely responsible for the architecture, but is the technical leader of the team
- This person will have the final say regarding technical decisions, but tries to avoid dictating the architectural direction in favor of a collaborative, team-based approach



- www.agilemodeling.com/essays/architectureOwner.htm

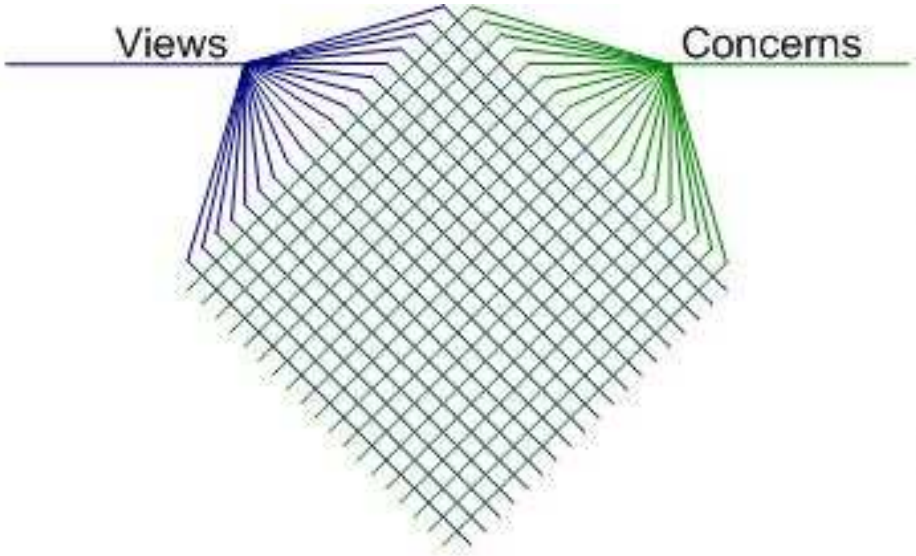
Travel Light

- Every artifact that you create, and then decide to keep, will need to be maintained over time
- Detailed specifications early in the lifecycle increase project risk by:
  - Motivating you to make significant decisions earlier in the lifecycle than they actually need to be made
  - Motivating you to stick to questionable decisions because it's too onerous to rework all of the artifacts
  - Increasing the cost of making changes
- Strive to get the benefit out of modeling which is to think things through without taking on the risk of unnecessary documentation



Take a Multi-View Approach

Code Distribution
Data Storage
Data Transmission
Deployment
Function/Logic/Services
Events
Hardware
Network
System Interface
User Interface
Usage



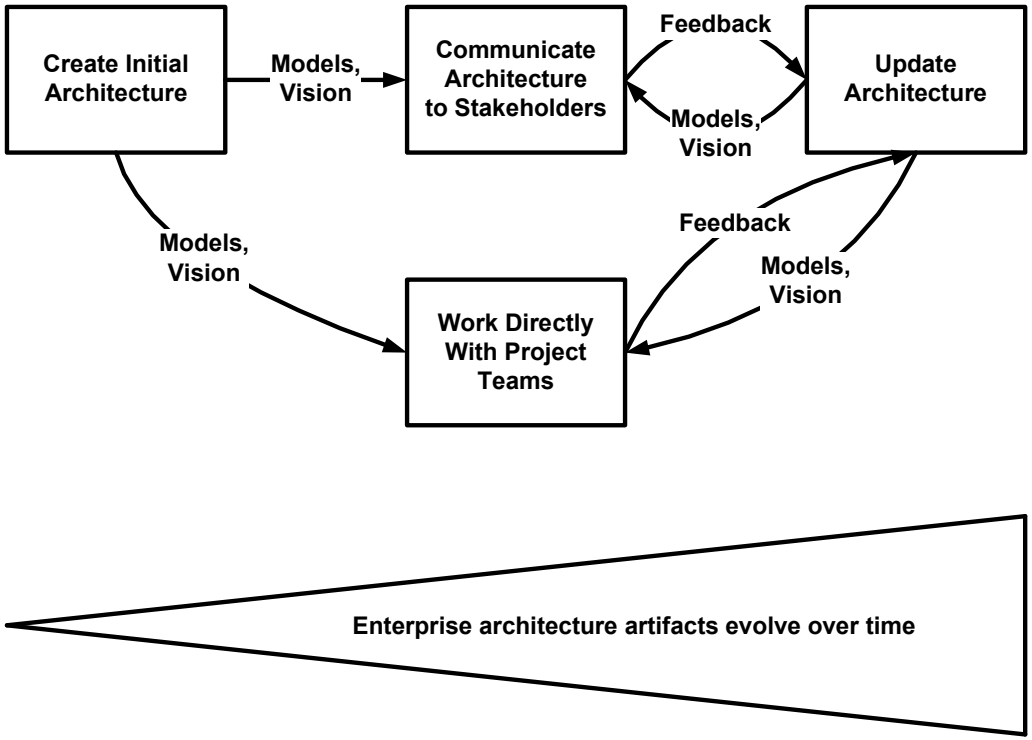
Accuracy
Availability
Concurrency
Consumability
Customization Points
Internationalization
Layering/Partitioning
Maintenance
Operations
Quality
Performance
Regulations
Reliability
Reuse
Security
Serviceability
Support
Timeliness
Usability
Validation

Agenda

- Some industry statistics
- Disciplined agile delivery
- Agile architecture strategies
- Agile enterprise architecture strategies
- Scaling agile
- Parting thoughts

Agile Enterprise Architecture

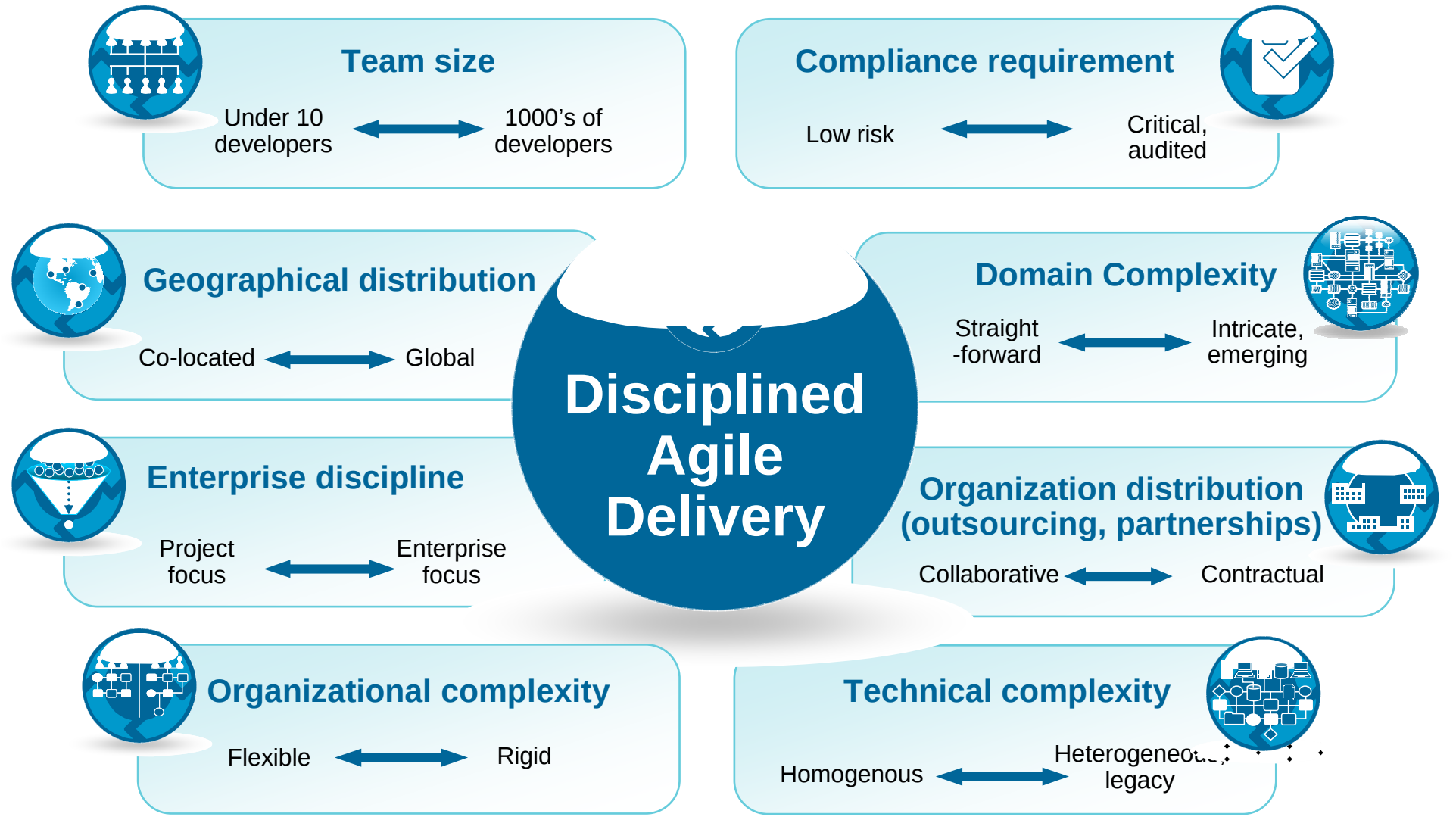
- Create slim models at first
- Get actively involved with teams
  - Mentor them
  - Lead the technical effort
- Work with stakeholders
- Evolve enterprise architecture artifacts over time



Agenda

- Some industry statistics
- Disciplined agile delivery
- Agile architecture strategies
- Agile enterprise architecture strategies
- Scaling agile
- Parting thoughts

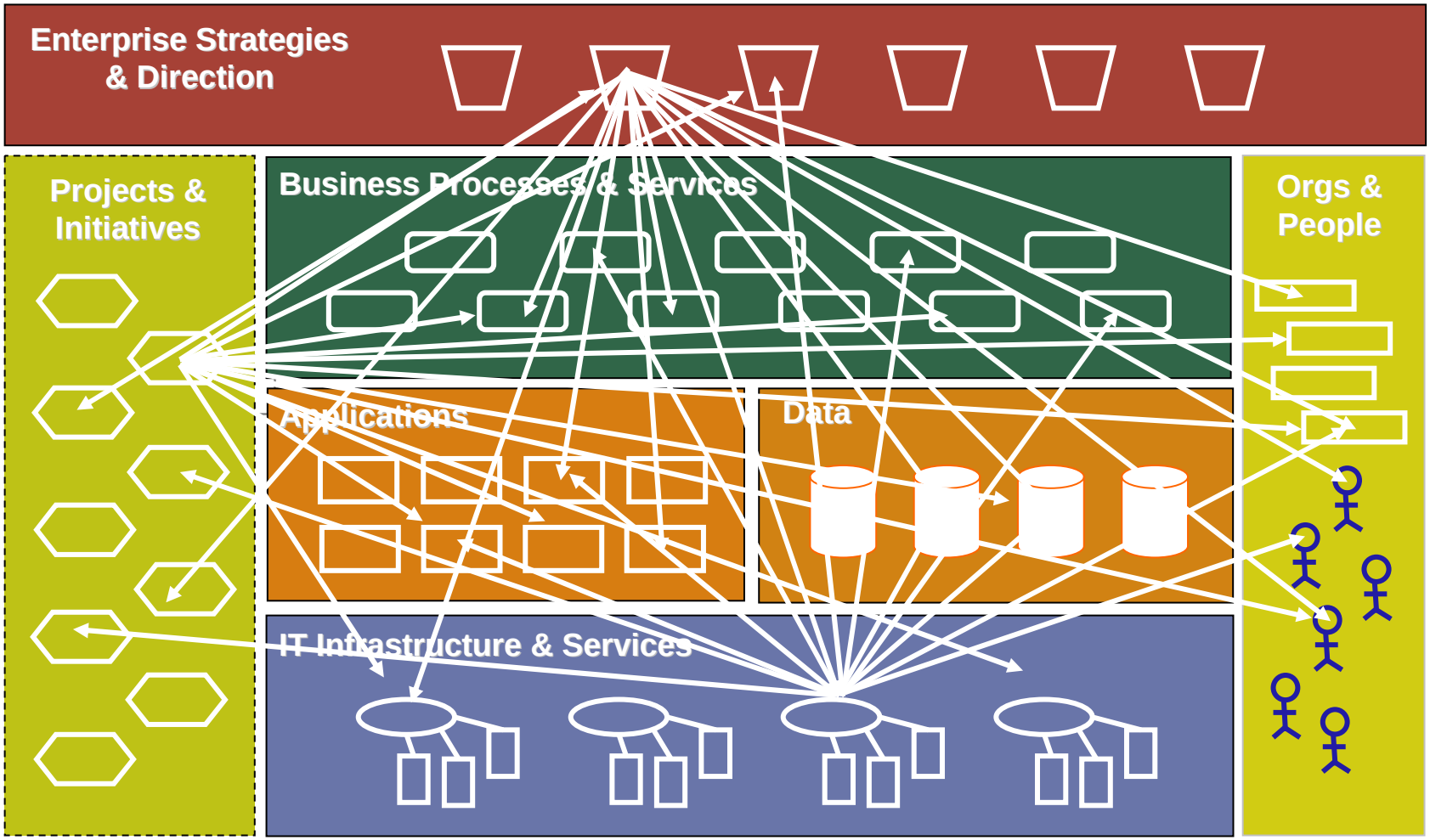
Agile scaling factors



Agenda

- Some industry statistics
- Disciplined agile delivery
- Agile architecture strategies
- Agile enterprise architecture strategies
- Scaling agile
- Parting thoughts

EA helps minimize risk associated with change



Understand Enterprise Strategies & Their Implementation
Understand How Infrastructure Changes Impact the Business
Understand Projects' Dependencies and Impacts on the Organization

How can architecture(s) help?

Understand What You Have
- Current state analysis
- Your IP and Assets

Improve What You Have
- Find incremental improvements
- Manage business transformation

Satisfy Mandated Compliance
- Regulatory or contractual
- DoDAF, TOGAF, FEA (iRMA), etc.

Pass an Audit
- Architecture & business transparency
- Repeatable, documented

Manage the Portfolio
- Applications/Products
- Improved reuse across organization

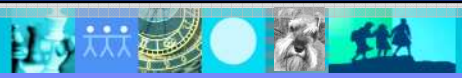
Return on Assets
- Leverage elements across subsystems and product lines

Visualize and Communicate
- Beyond basic drawing tools

Common Project Starting Points
- Initiate new projects from a common starting point based on an EA model

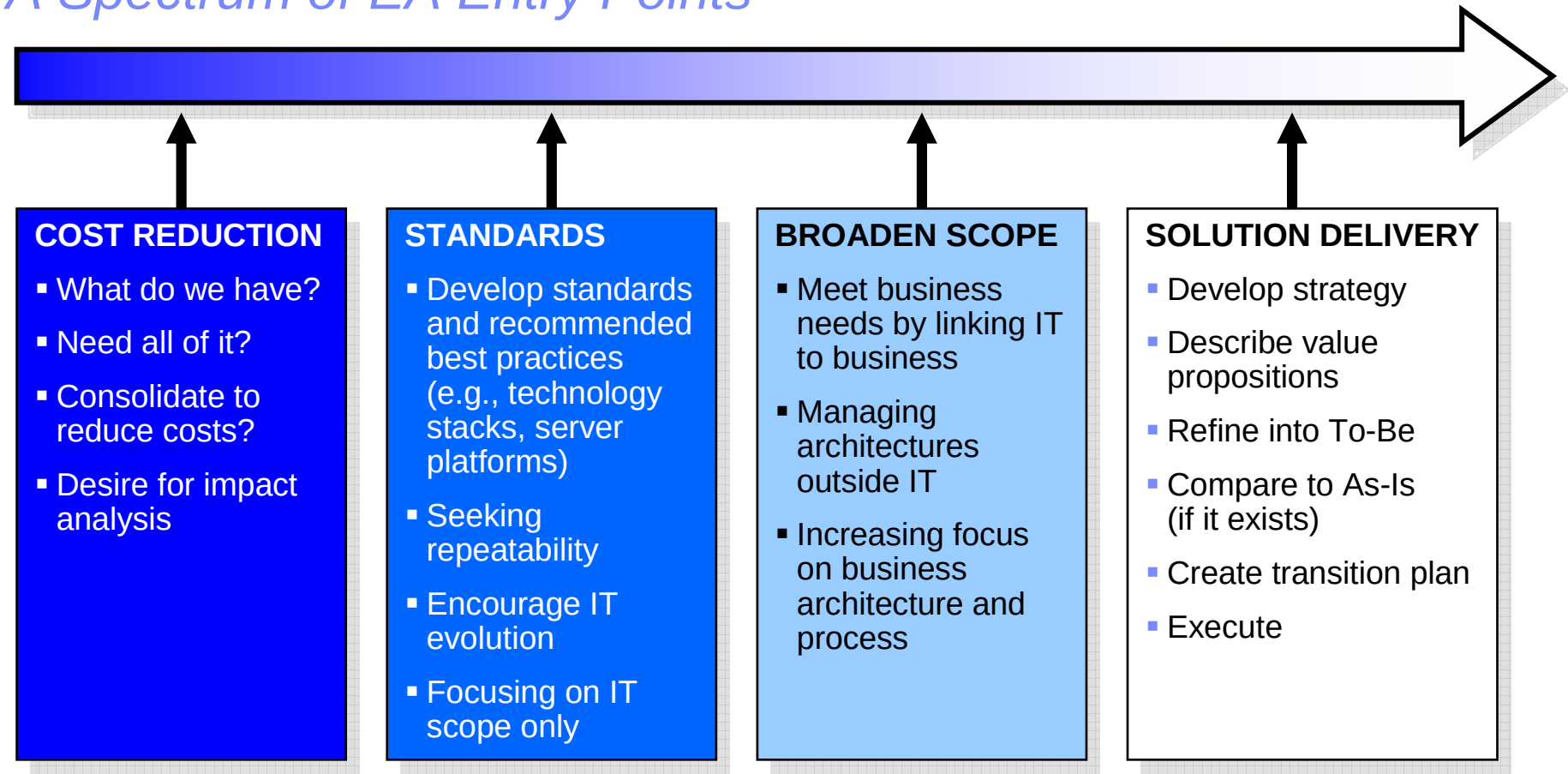
Manage Outsourcing
- Customer: Req’ts, specs, testing
- Vendor: Actual architectures

Manage Packaged Applications
- Integrate with rest of architecture



Is there a single approach?

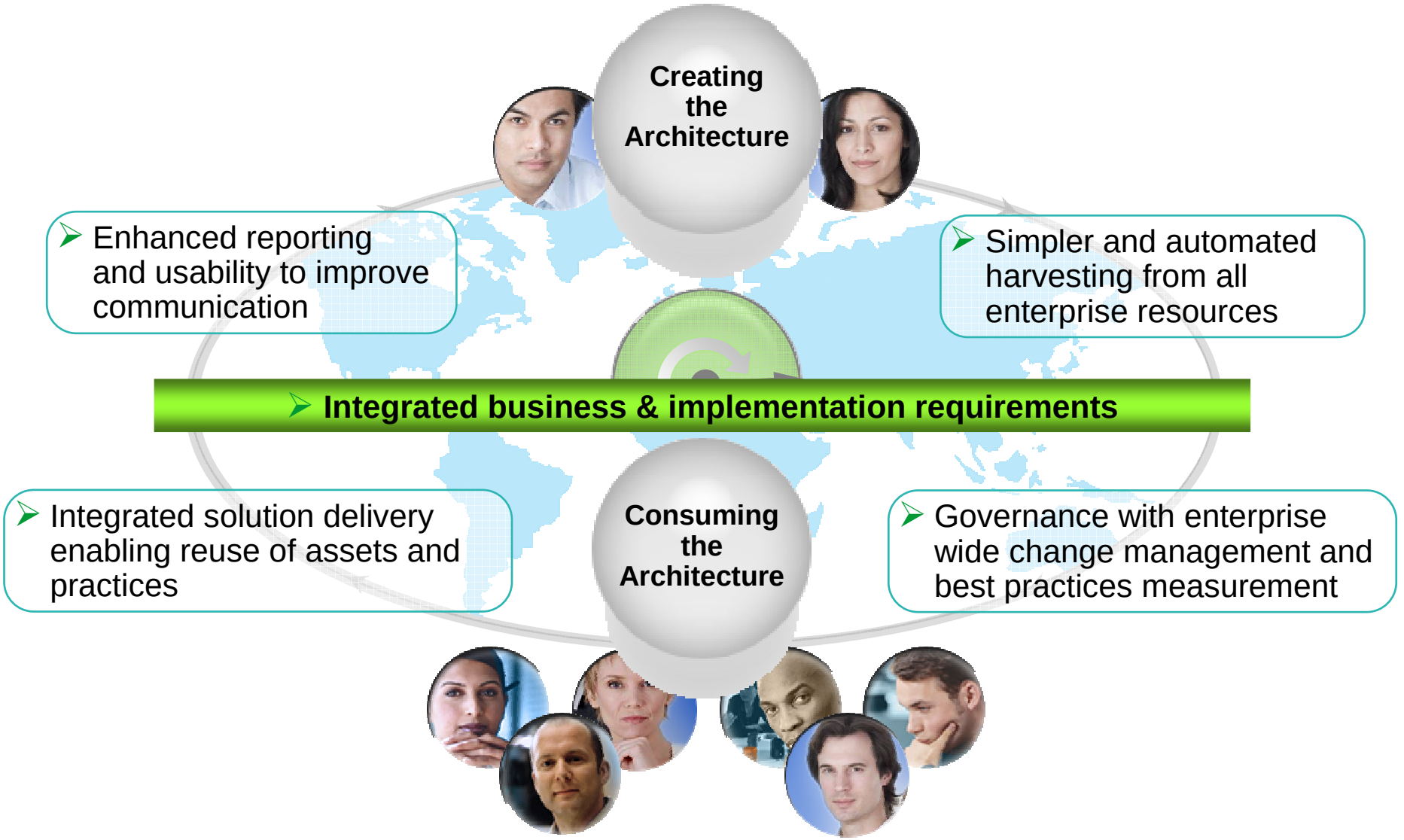
A Spectrum of EA Entry Points



IBM'S EA approach allows:

- Multiple entry points to more quickly realize value
- The ability to manage upstream to downstream process flows

IBM's vision for enabling broader adoption of EA

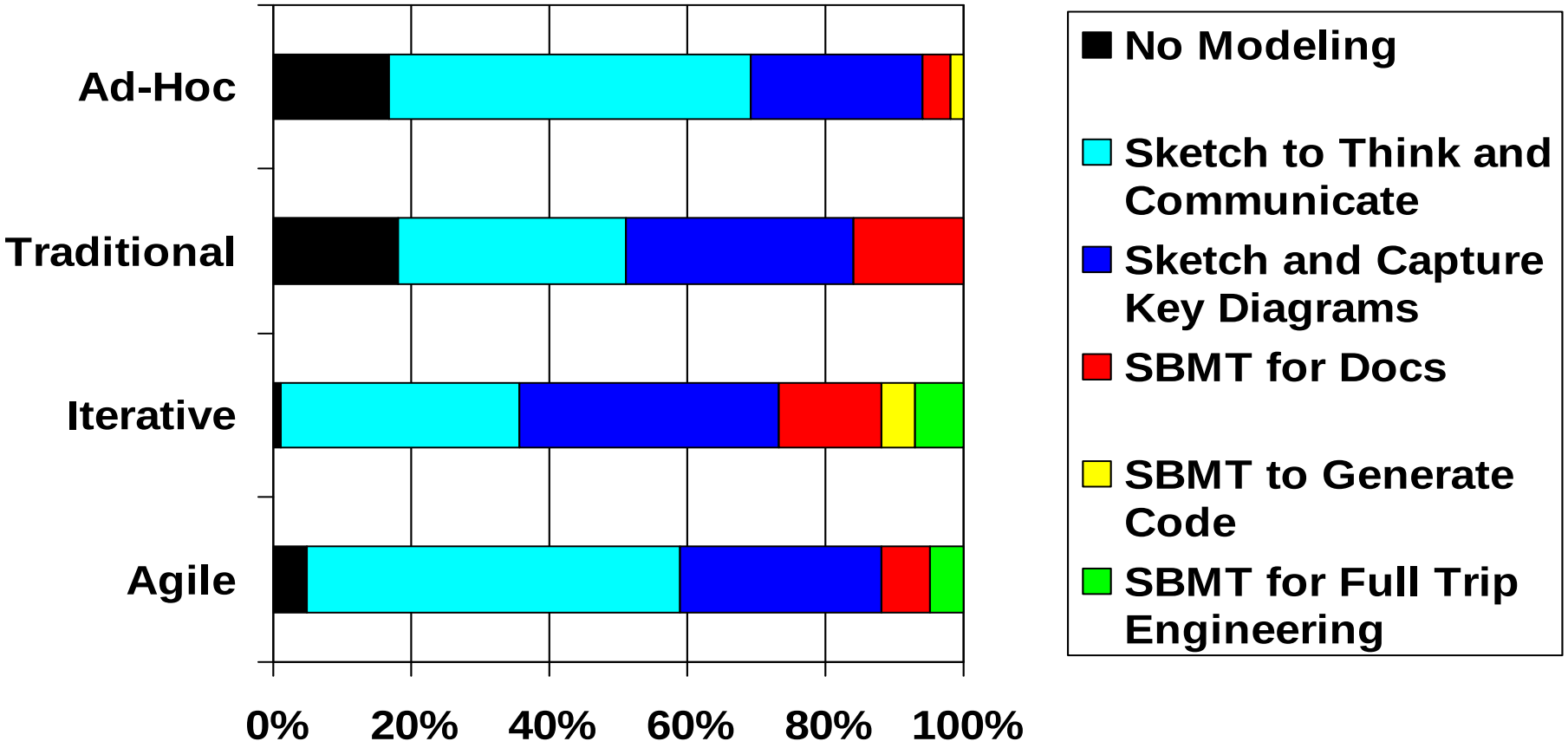


Backup slides

IBM agility@scale™

Rational. software

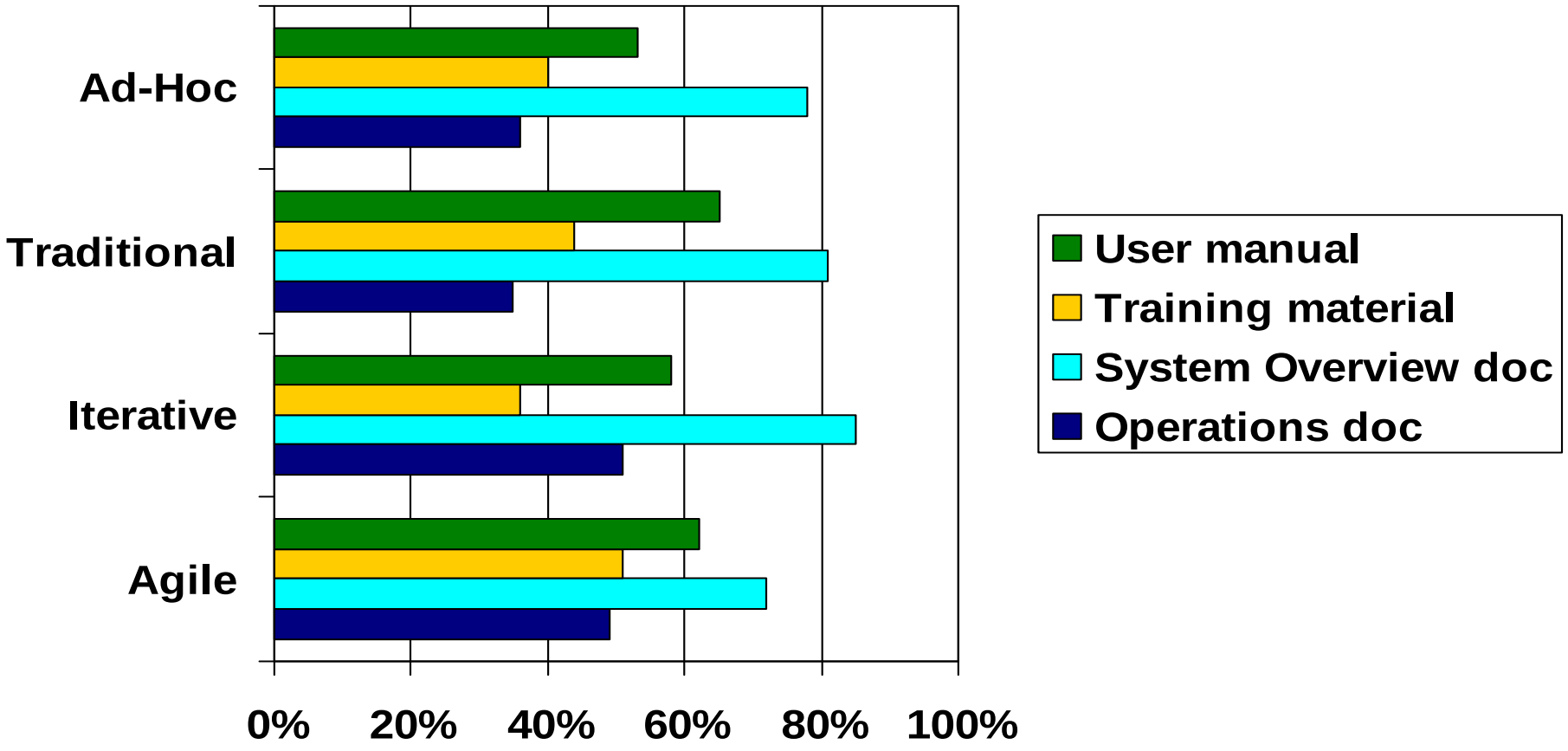
Primary Strategy for Modeling



Only the most disciplined development teams use software-based modeling tools (SBMTs) in practice

Source: Dr Dobb's 2008 Modeling and Documentation Survey

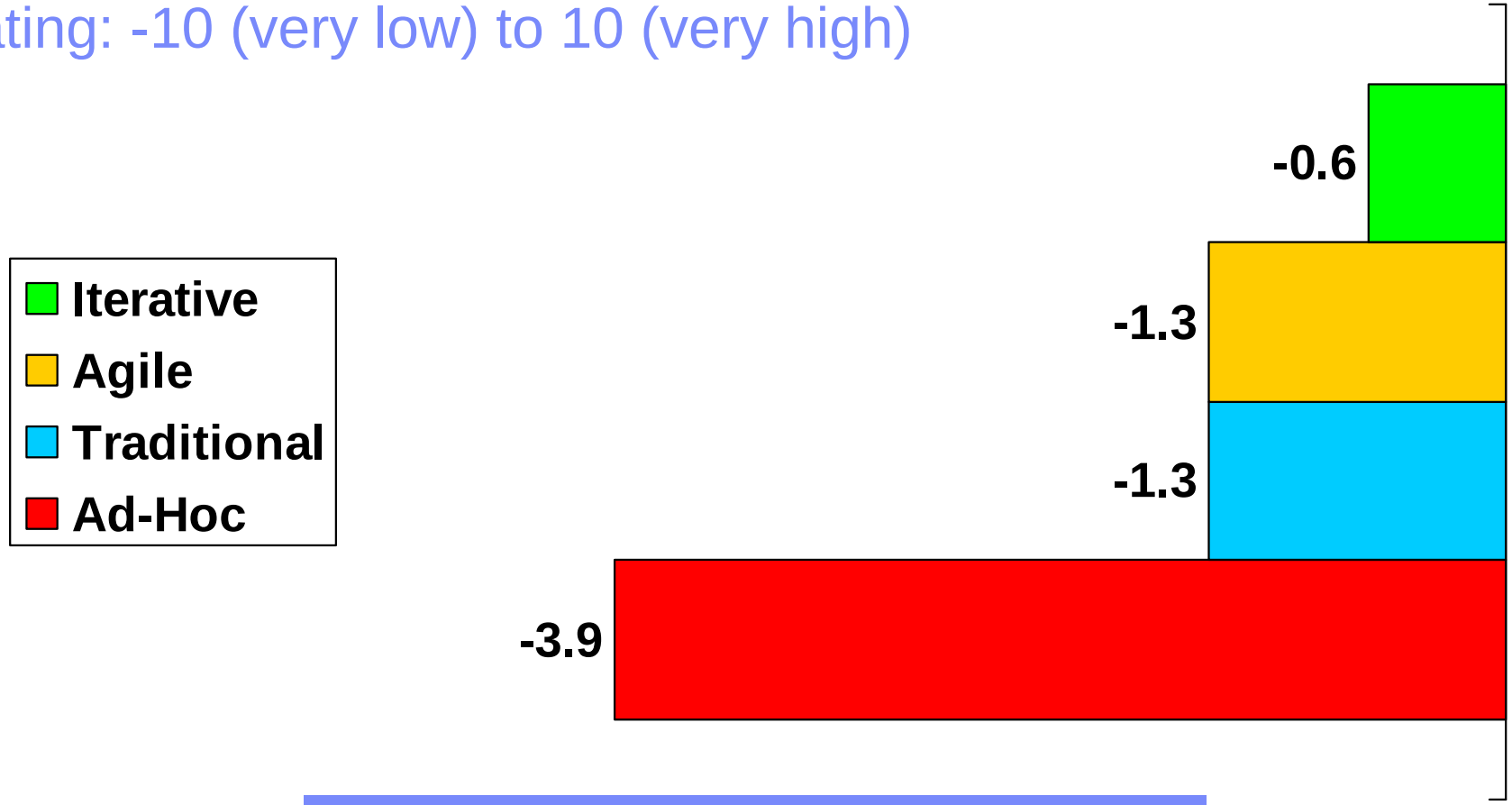
Percentage of Teams Creating Deliverable Documentation



Agile teams create deliverable documentation too!

Source: Dr Dobb's 2008 Modeling and Documentation Survey

What is the quality of the deliverable documentation produced by a development team?
Rating: -10 (very low) to 10 (very high)



And the “quality” of the documentation is the same.

Source: Dr Dobb’s September 2009 State of the IT Union Survey

Tooling for agile IT software teams

- Flagship Agile Products
  - Rational Team Concert (RTC) – Distributed agile development, project monitoring
- Primary Agile Products
  - Rational Application Developer (RAD) – Development
  - Rational AppScan – Web site security testing
  - Rational Build Forge (RBF) – Continuous integration, deployment
  - Rational Insight – Governance
  - Rational Project Conductor – Project Management
  - Rational Quality Manager (RQM) – Test management
  - Rational Requirements Composer (RRC) – Requirements modeling
  - Rational Software Analyzer (RSAR) – Static code analysis
- Extended Agile Products
  - Other products are potential candidates for scaling purposes

Tooling for agile embedded software teams

- Flagship Agile Products
  - Rational Team Concert (RTC) – Distributed agile development, project monitoring
- Primary Agile Products
  - Rational Insight – Governance
  - Rational Performance Tester
  - Rational Project Conductor – Project Management
  - Rational Quality Manager (RQM) – Test management
  - Rational Rhapsody – Modeling
  - Rational Software Analyzer (RSAR) – Static code analysis
  - Rational Test RealTime – Testing
- Extended Agile Products
  - Other products are potential candidates for scaling purposes