Agile Requirements at Scale: Context Counts

Davyd Norris
Senior IT Architect , IBM Rational
dnorris@au1.ibm.com

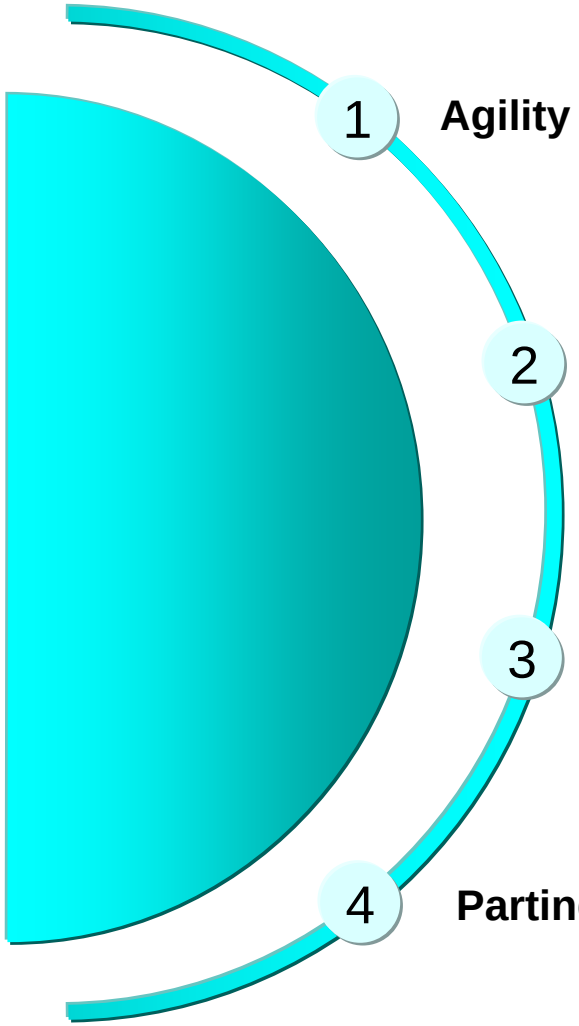
Rational. software

IBM agility@scale™

My Goals

- To challenge your beliefs regarding requirements practices
- To share industry data with you
- To explore how to scale agile requirements practices
- To show that context counts, that one “process size” does not fit all

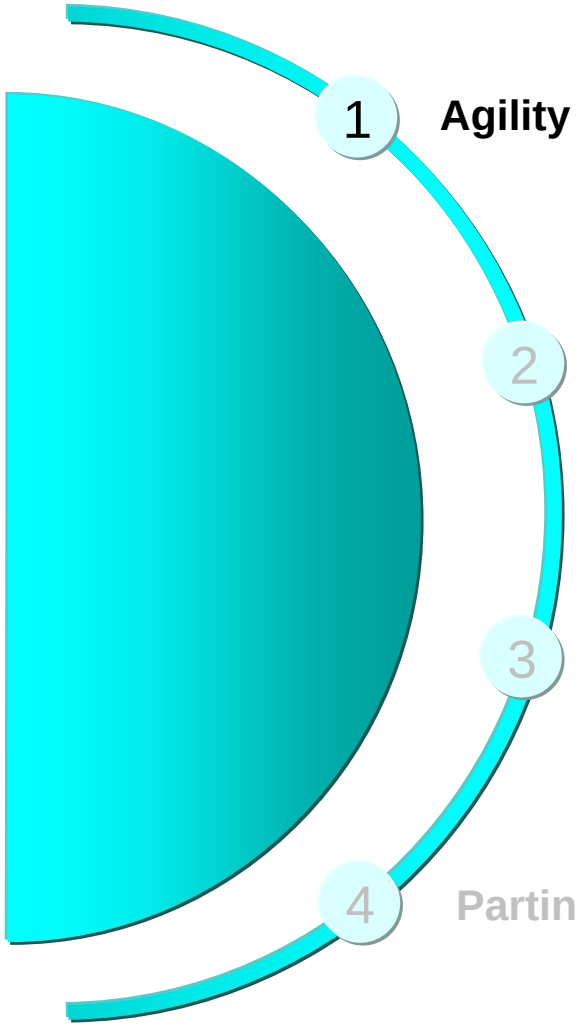
Agenda



1. **Agility and requirements**
2. **Introduction to agility@scale**
3. **Agile requirements at scale**
4. **Parting thoughts**

IBM agility@scale™

Agenda



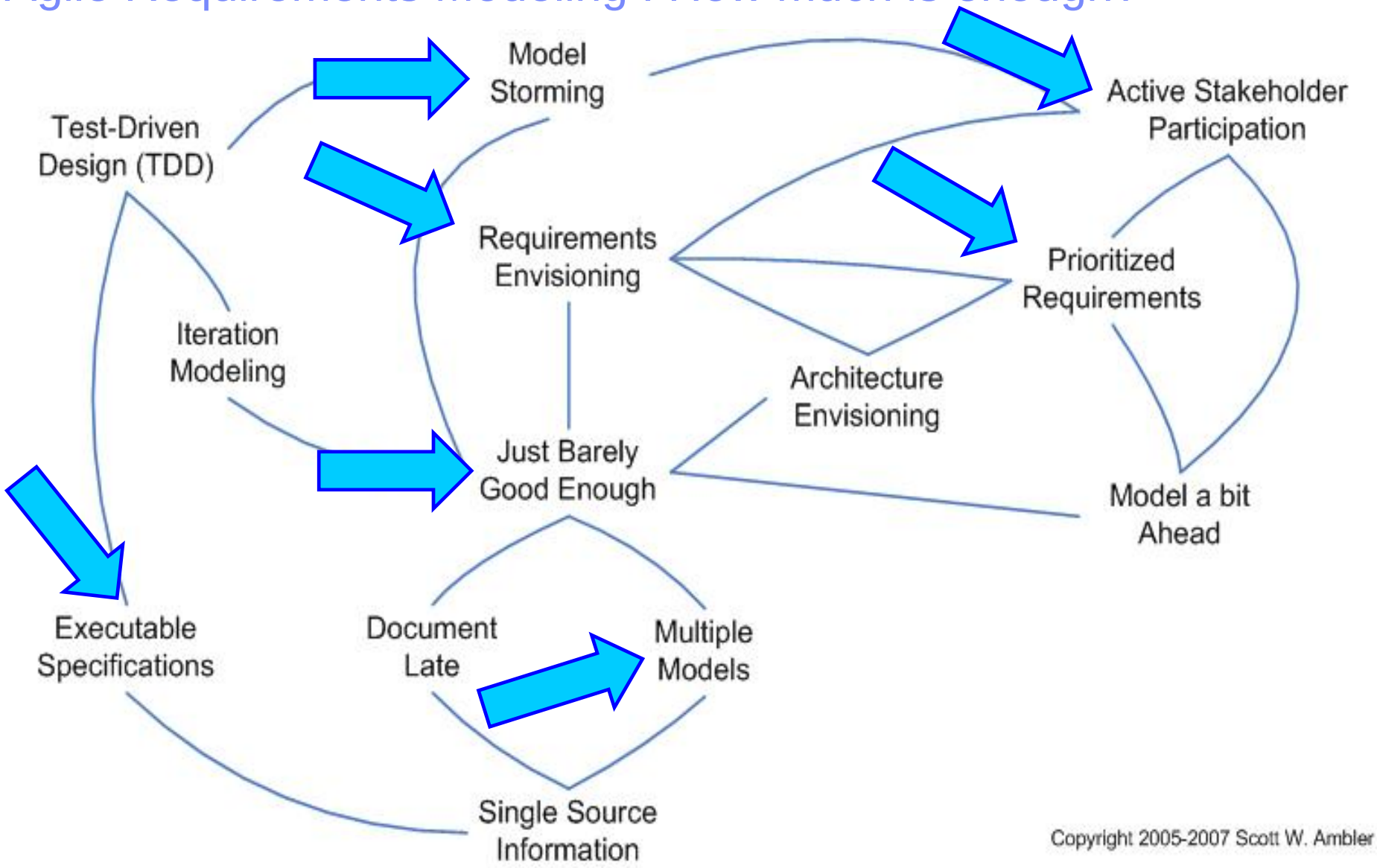
1. **Agility and requirements**
2. Introduction to agility@scale
3. Agile requirements at scale
4. Parting thoughts

IBM agility@scale™

Requirements strategies are changing

- Continual customer involvement
  - Product owner represents the stakeholders
- Shared vision
  - Understand business needs
  - Focus on stakeholders goals
- Requirements elicitations
  - Conversations, agile modeling, workshops
- Requirements analysis
  - Performed “just in time”
- Requirements documentation
  - User stories, storyboards, acceptance tests, agile models
  - Test your documentation effectiveness : ”CRUFT” Measure
- Formality
  - Improvised, more relaxed approach

Agile Requirements Modeling : How much is enough?

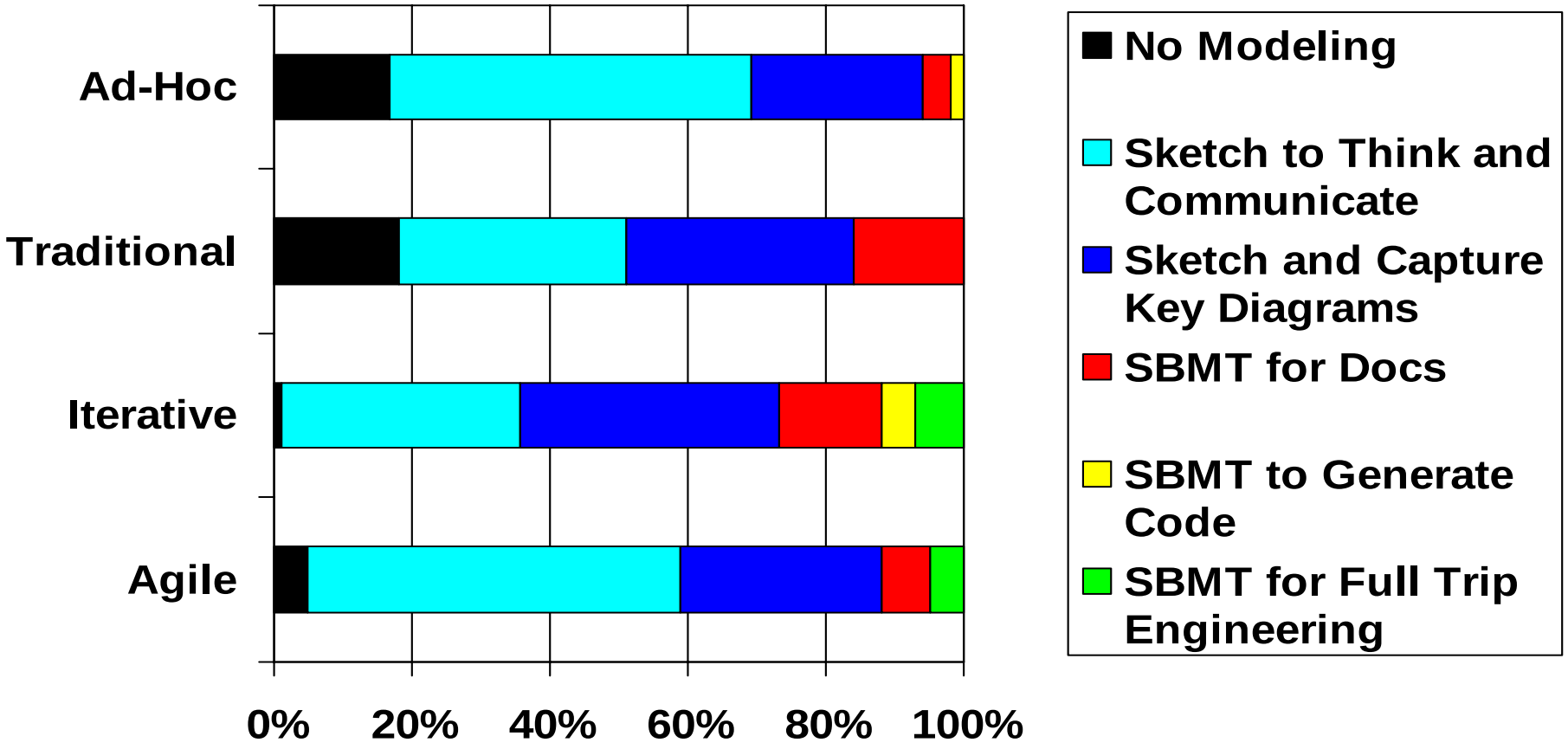


Agile Approach to Initial Requirements

76%	High-level initial requirements modeling
30%	Detailed initial requirements modeling
12%	Have initial requirements models supplied to them
10%	Use enterprise models as a reference
12%	Use industry models as a reference
88%	Do some sort of initial modeling or have initial models supplied to them
89%	Do some sort of initial modeling, or have initial models supplied, or leverage reference models

Source: Ambysoft 2009 Agile Project Initiation Survey

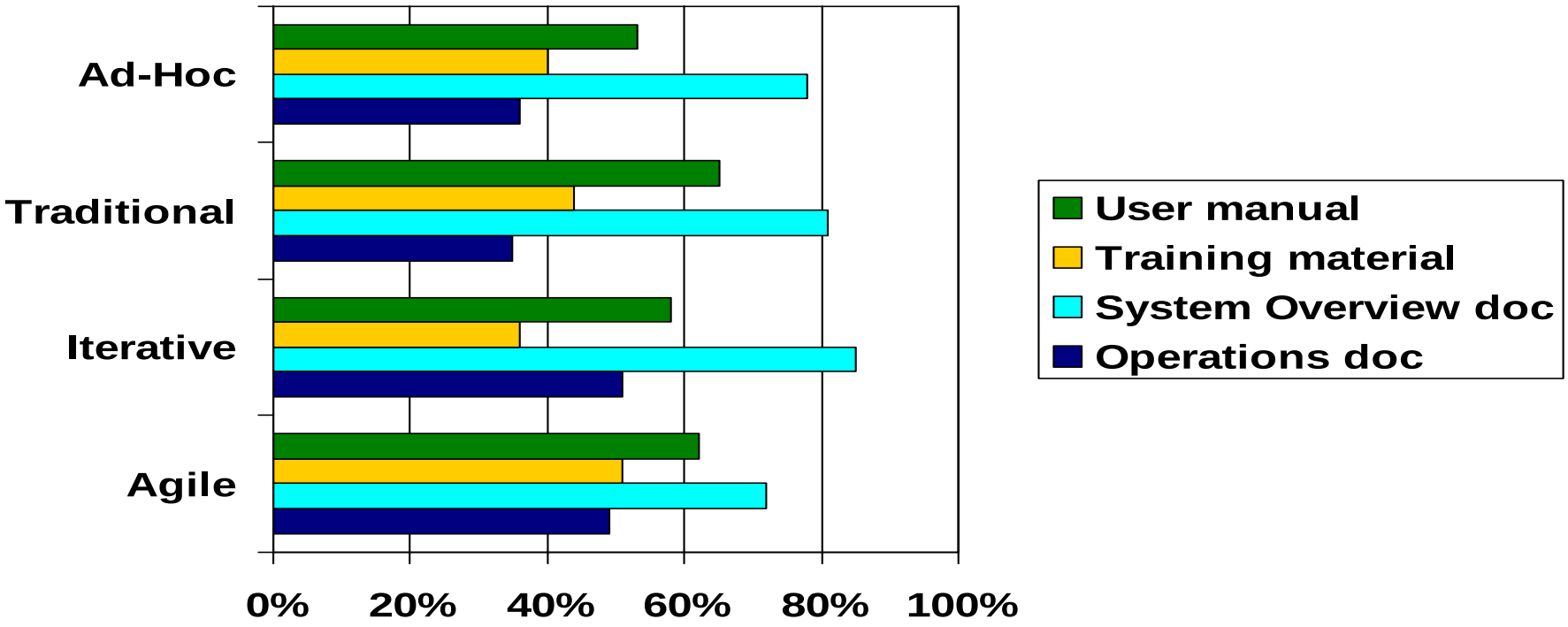
Agilists Model!
Their primary strategy for modeling



Source: Dr Dobb's 2008 Modeling and Documentation Survey

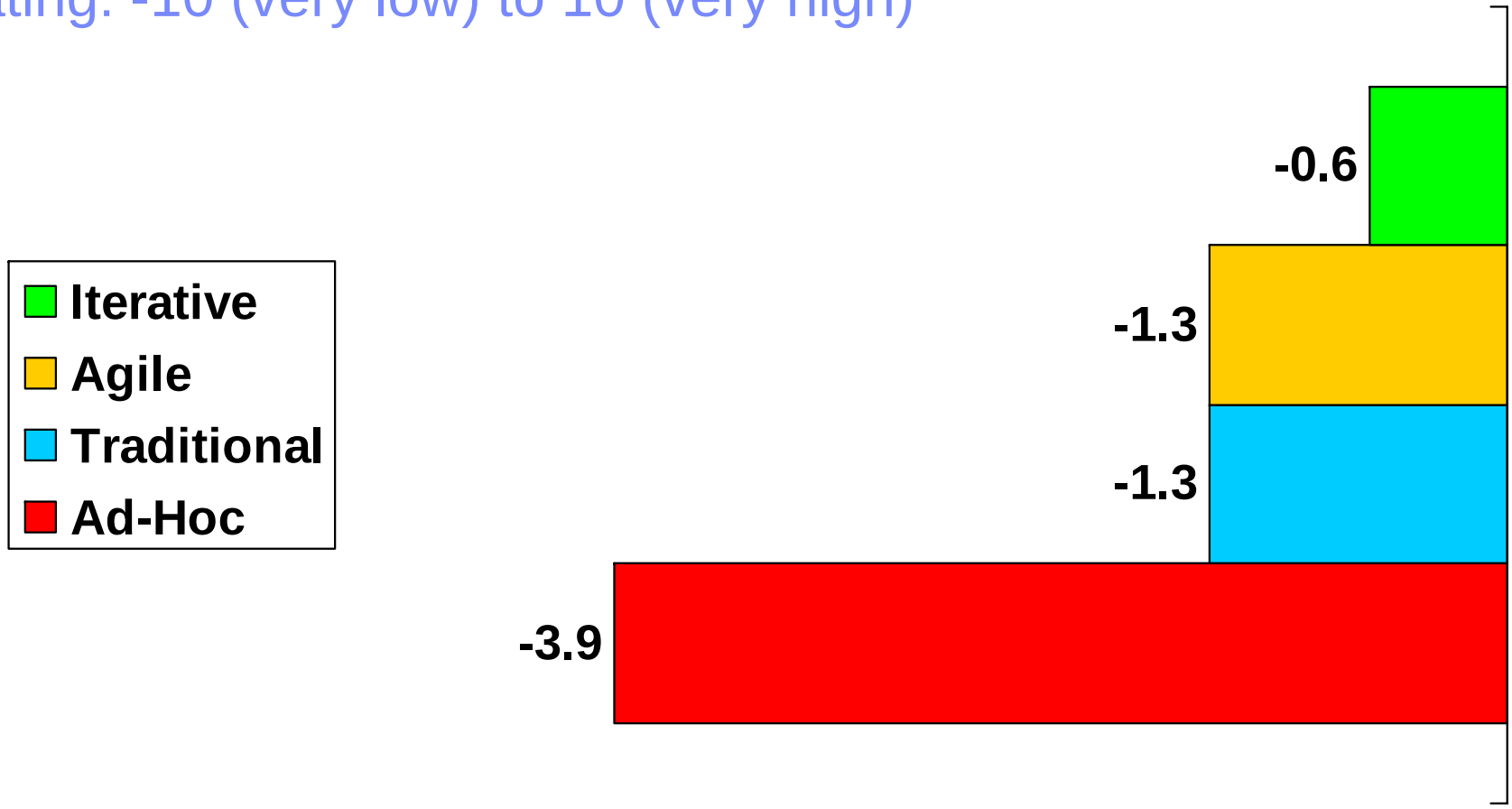
SBMT: Software-Based Modeling Tool

Agilists Write Documentation!
Percentage of teams creating deliverable documentation



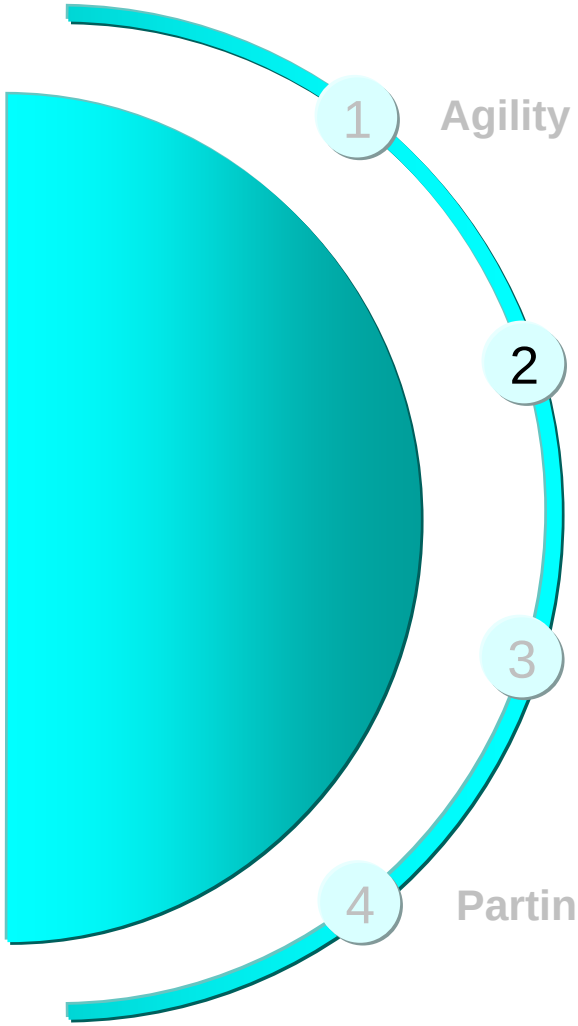
Source: Dr Dobb's 2008 Modeling and Documentation Survey

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



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

Agenda



1 Agility and requirements

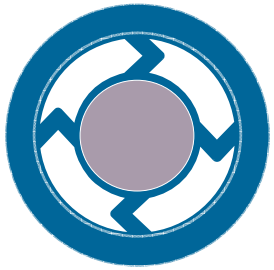
2 **Introduction to agility@scale**

3 Agile requirements at scale

4 Parting thoughts

IBM agility@scale™

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

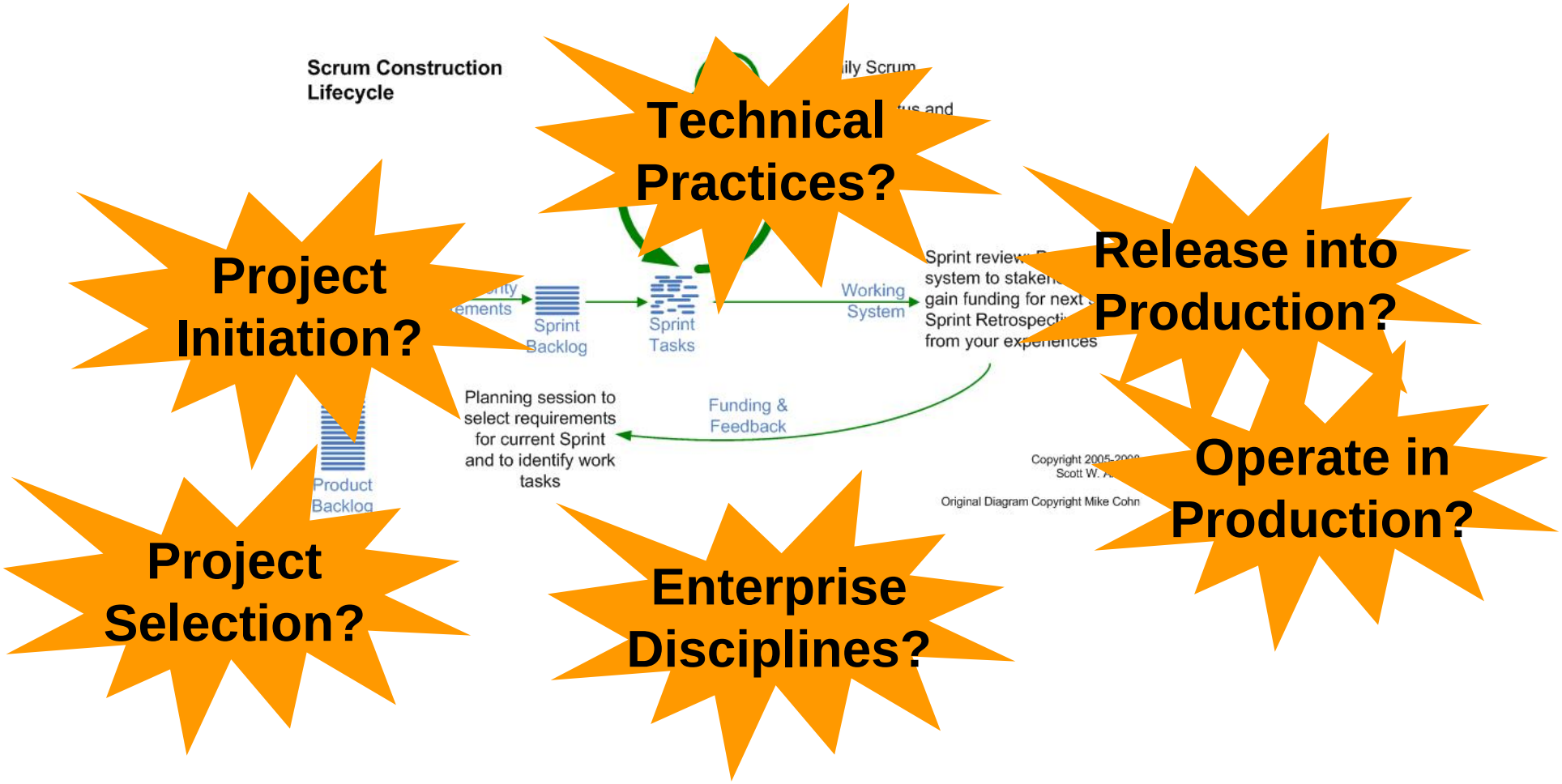
Disciplined Agile Delivery

- 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

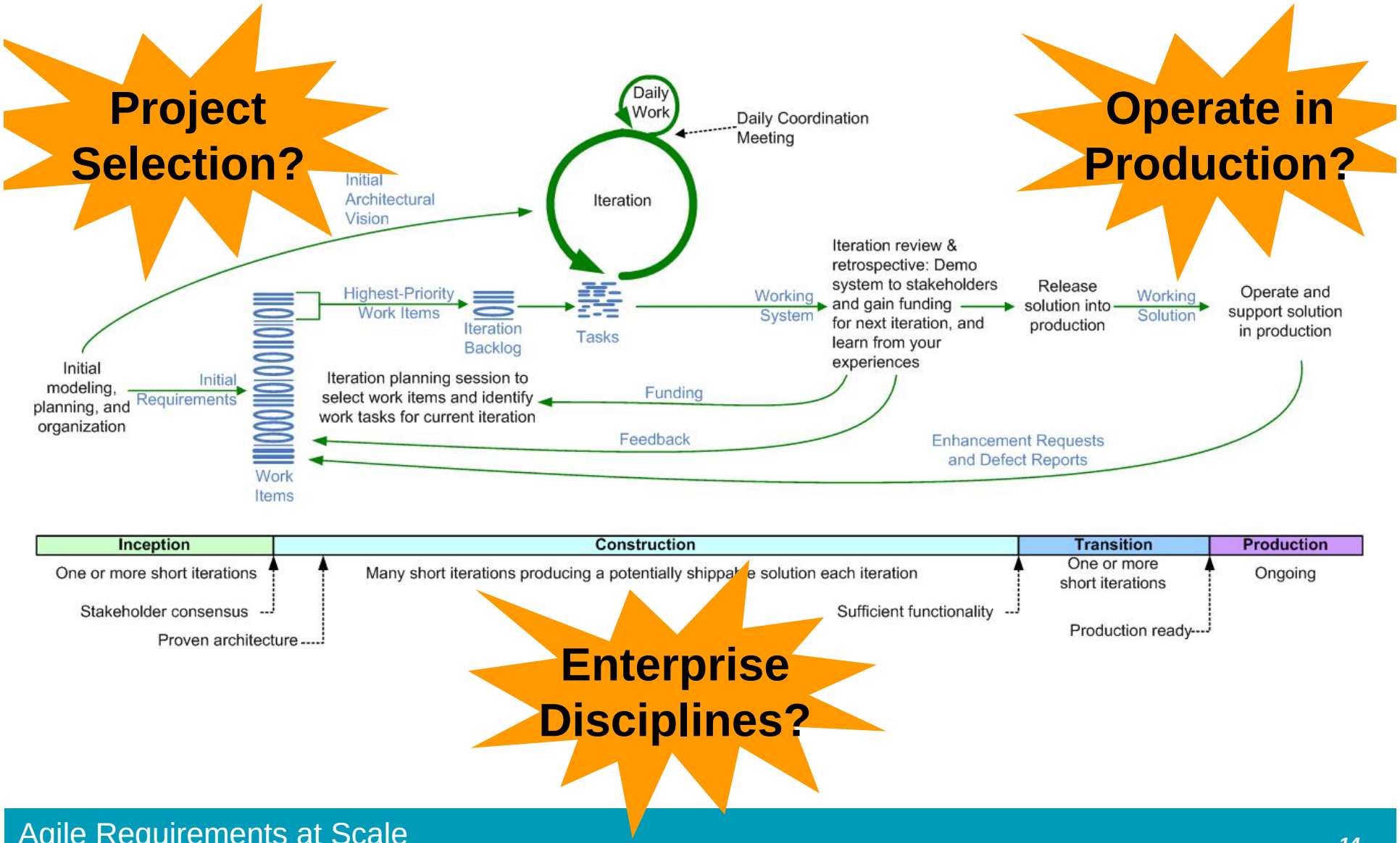
Agility at Scale

- Disciplined agile delivery and one or more scaling factors applies

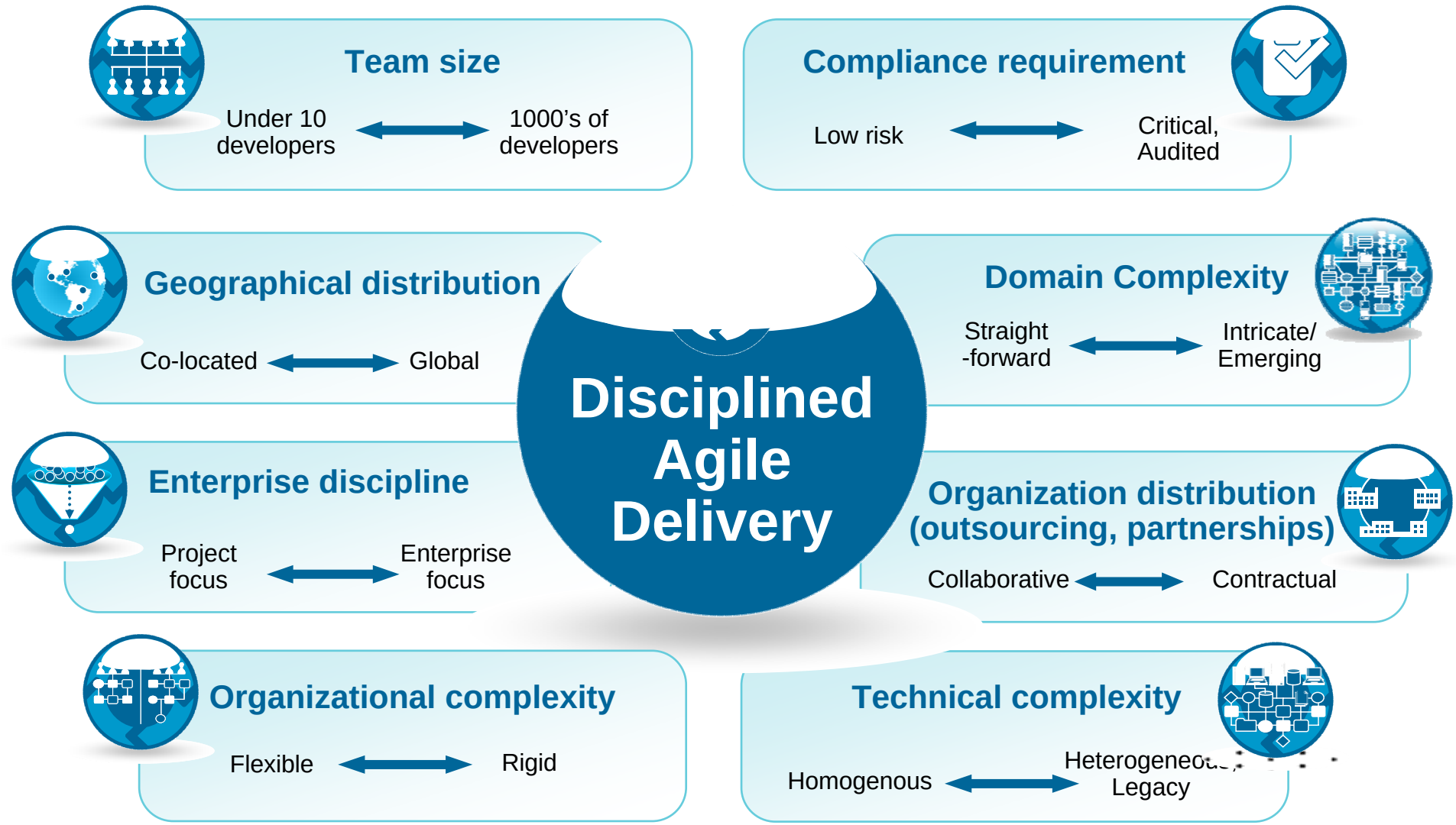
The Scrum construction lifecycle



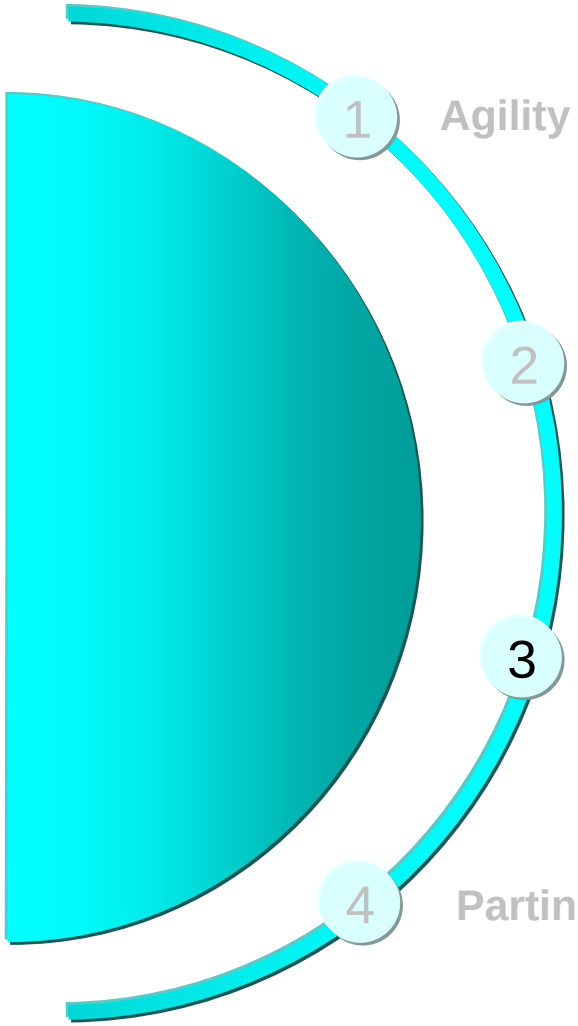
The disciplined agile delivery life cycle



Agile scaling factors



Agenda



1. Agility and requirements
2. Introduction to agility@scale
3. **Agile requirements at scale**
4. Parting thoughts

IBM agility@scale™

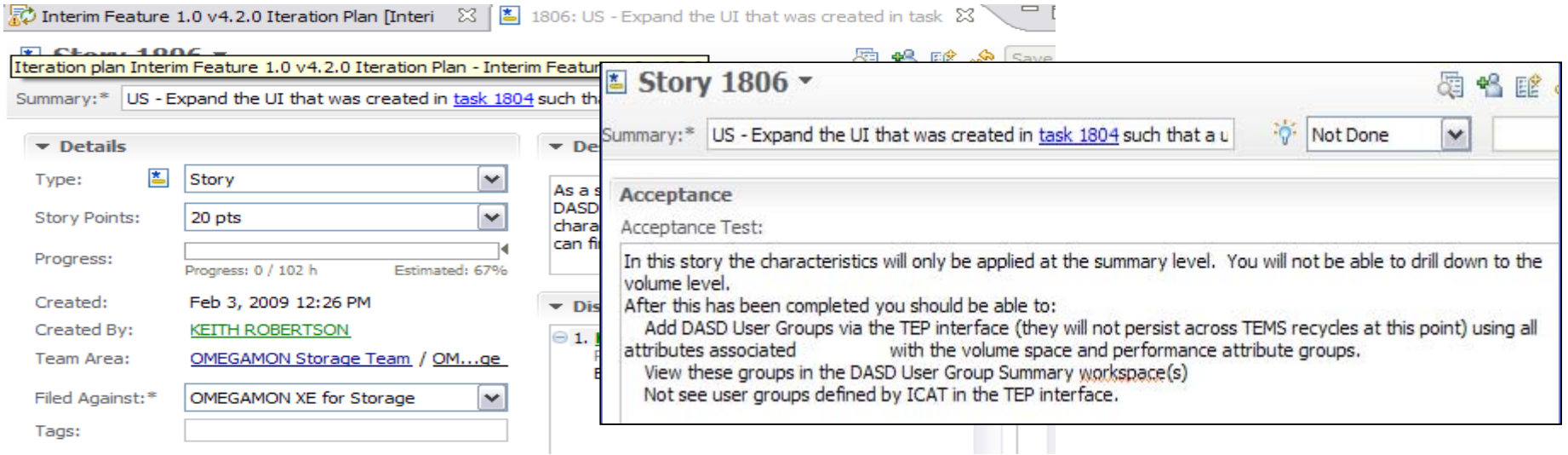
Requirements and disciplined agile delivery

- Small, co-located team developing a straightforward solution
- Capture requirements using inclusive tools (paper and whiteboards)
- Planning can be done by displaying the cards on a corkboard or table
- Prioritization can be done by keeping the cards in a stack

- Use simple tools and strategies for simple situations!

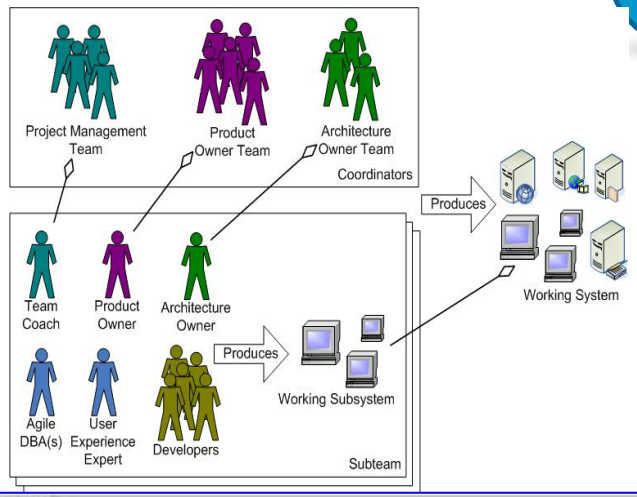
Agile requirements and large teams

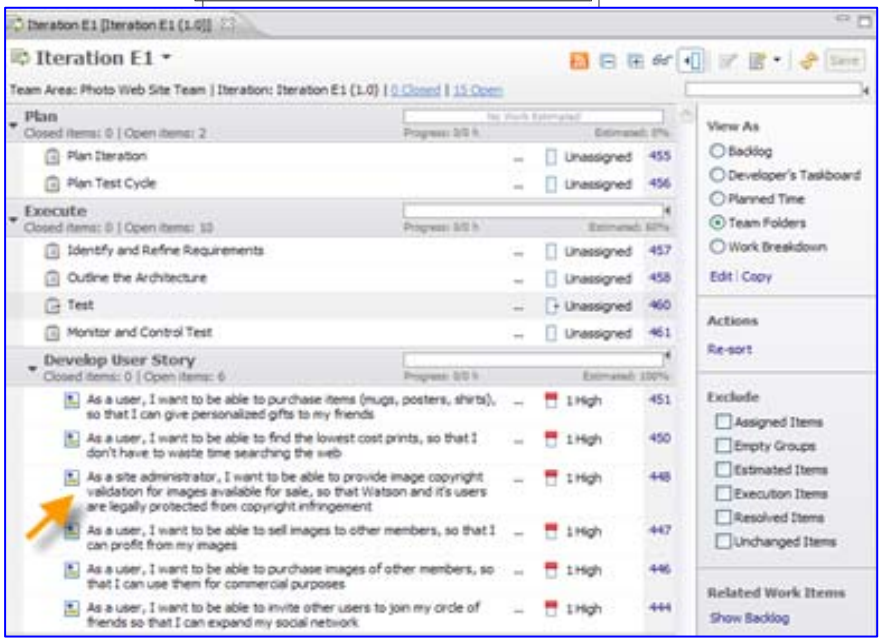
- Communication and coordination risk increases with large teams
- Initial requirements and architecture envisioning is critical
- Coordination of requirements between subteams is important
- Team organization, architecture, and requirements must reflect each other
- Re-enforce the usage of product backlog for scope management
- Use simple tools, apply some agile practices such as active participation of stakeholders



Agile requirements and geographically distributed development

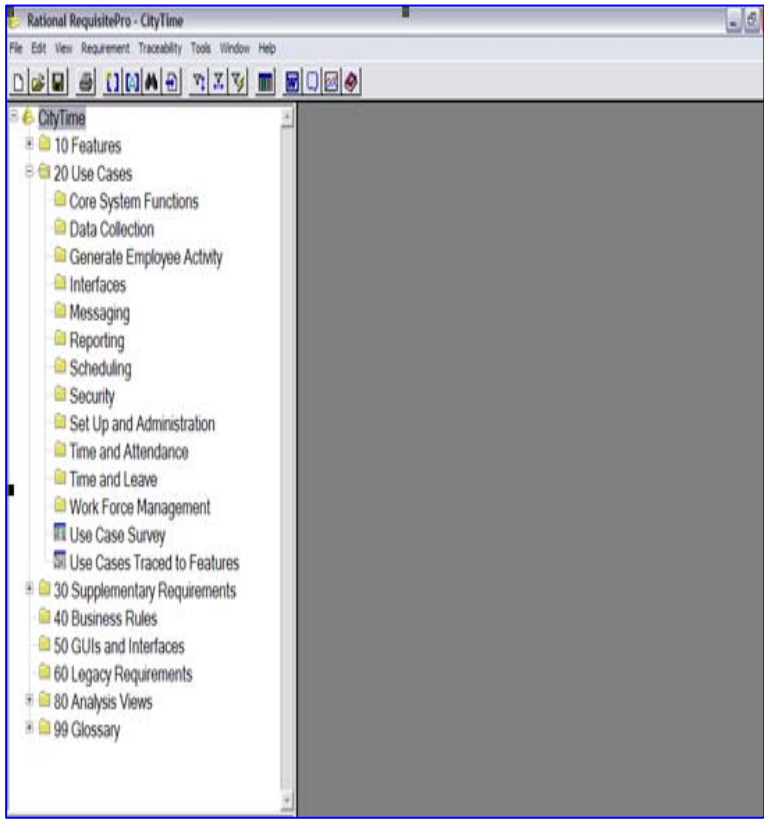
- Geographically distributed teams incur significant communication risk
- Need a more “disciplined” agile requirements approach
  - One that can address risks
  - Automation is a “must” for requirements traceability, version control and collaboration
  - Requirements dashboards and reporting on certain important measures become necessary
- Large team considerations apply





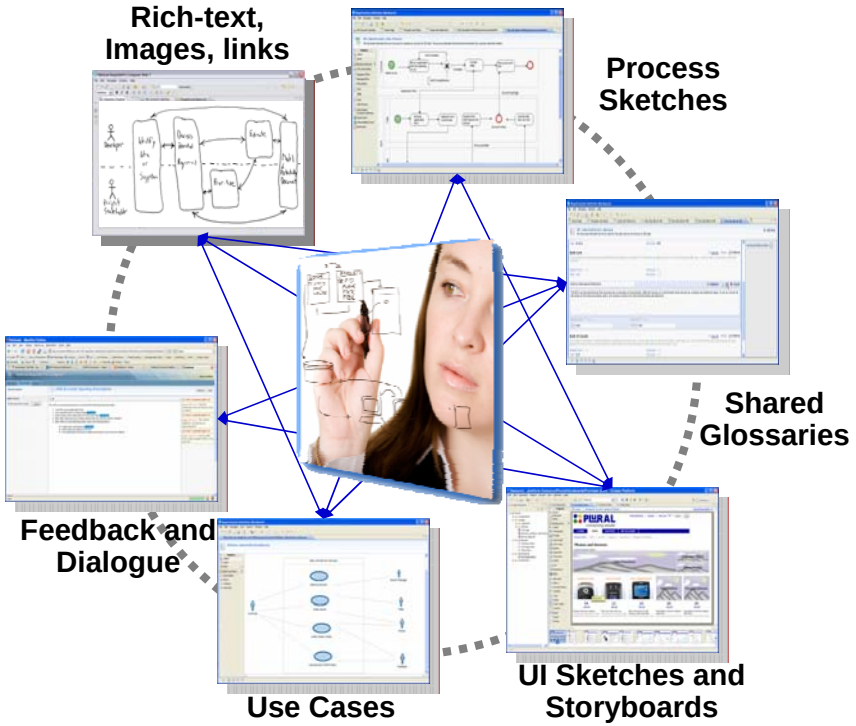
Agile requirements and regulatory compliance

- You may need to adopt other requirements strategies, such as use cases or formal System Requirements Specifications (SRSs)
  - BUT... read the regulations, because they likely don't specify how, nor when, to capture the requirements
- Traceability is often a secondary, but important, part of the regulation
  - BUT… read the regulations, because they likely don't specify the level of detail required
- You will likely need to write more documentation, particularly business rules and requirements pertaining to sensitive data
  - BUT… read the regulations, because you only need to do this to the extent of the risk of the project
- You may need to hold reviews
  - BUT… read the regulations, because they seldom require formal reviews



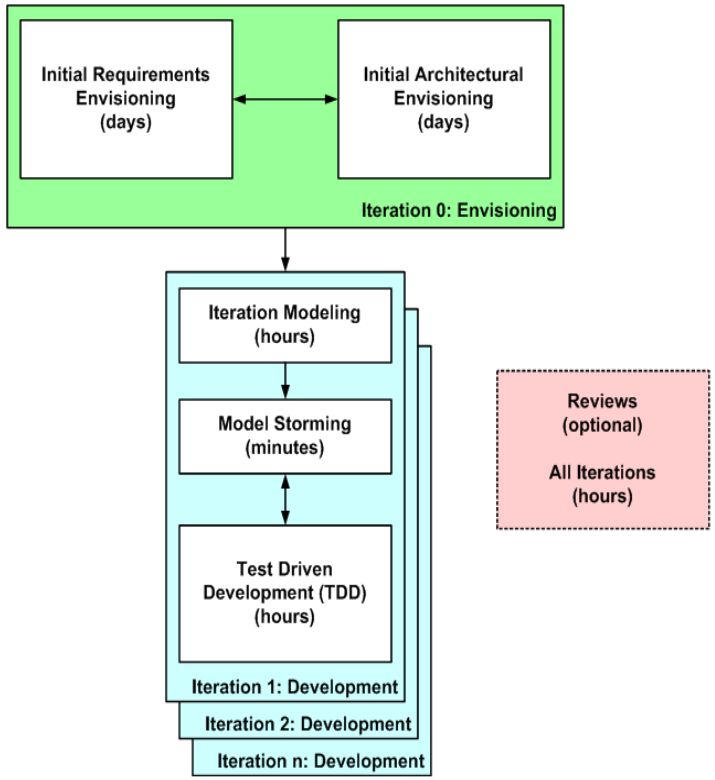
Agile requirements and domain complexity

- Business process sketching may help understand the complex domain environment
- Might want to consider light-weight use cases instead
- Will likely need to do more user interface (UI) prototyping
- Active participation of stakeholders throughout the life cycle is crucial for you to understand their changing needs
- Important: Complex domains don't imply that you need detailed requirements speculations



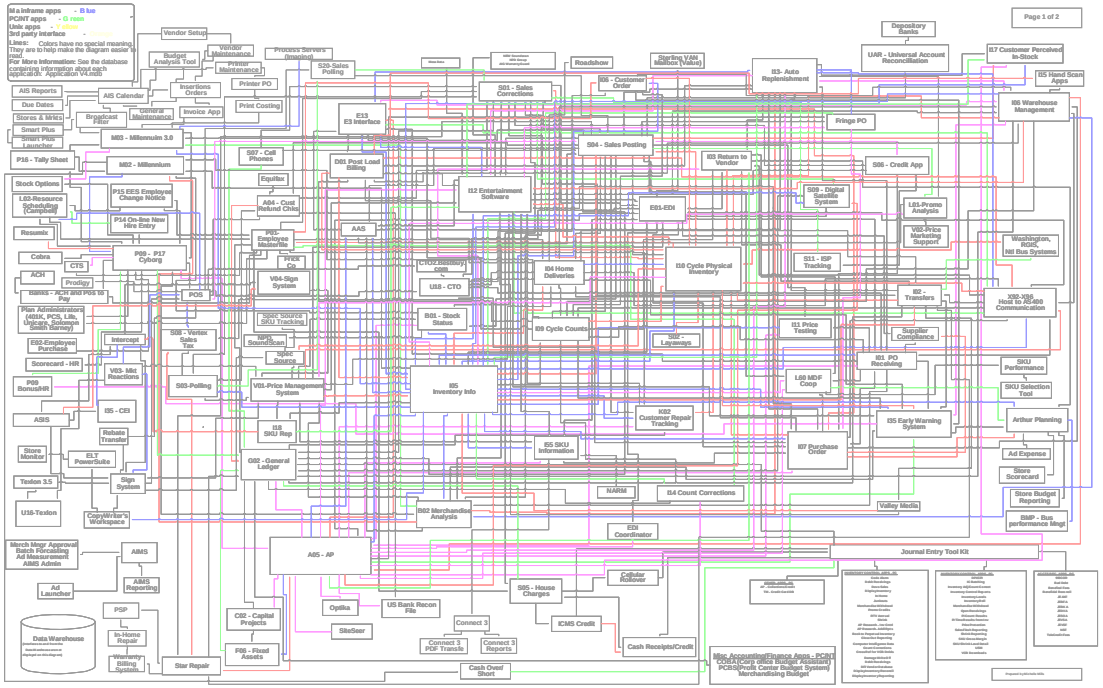
Agile requirements and organization distribution

- When multiple organizations are involved:
  - Access control to portions of the work may become critical
  - You will need to negotiate access rights, particularly in secure situations
  - Intellectual property (IP) rights need to be negotiated up front
- Initial requirements and architecture envisioning are required to organize the work between the organizations



Agile requirements and technical complexity

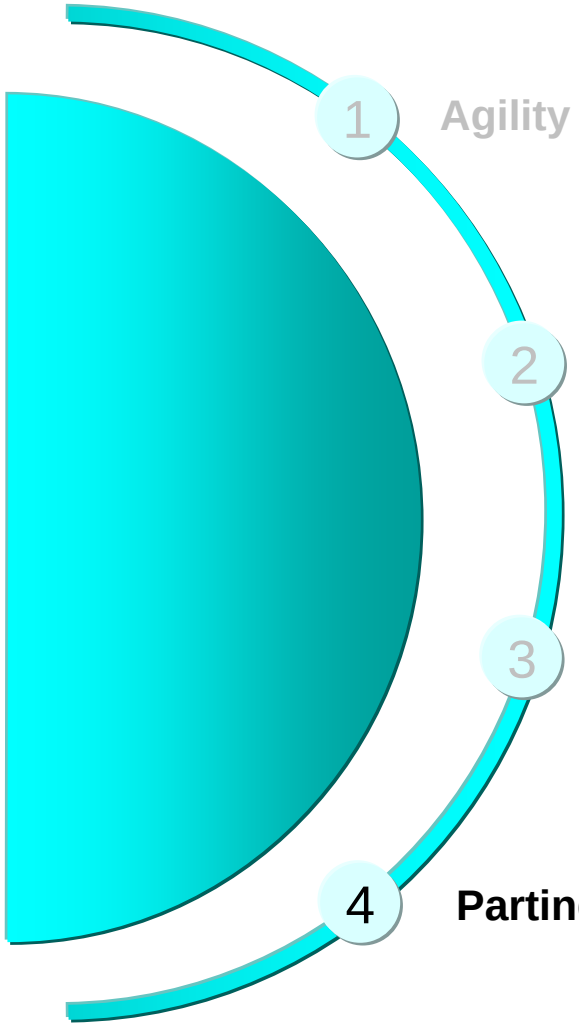
- May want to consider more sophisticated strategies, such as light-weight use cases, instead of user stories
- Sizing of the individual requirements becomes difficult if the development teams doesn't understand the technology
- Adopt a Risk-Driven life cycle (e.g. a disciplined agile delivery strategy)



Agile requirements and enterprise discipline

- Enterprise disciplines, such as enterprise business modeling, portfolio management, and enterprise architecture can add complexity to project level activities
- Enterprise business modeling
  - Enterprise-level process and domain models very common
  - May need to trace stories back to these models, although perhaps a high-level trace from the project to these models is more effective
- Enterprise architecture (EA)
  - Technical stories which reflect the EA may be common
  - The EA will motivate non-functional requirements for user stories
- Enterprise portfolio management
  - Need to identify visions for potential products
  - Key user stories may help, but use cases or scenarios better options

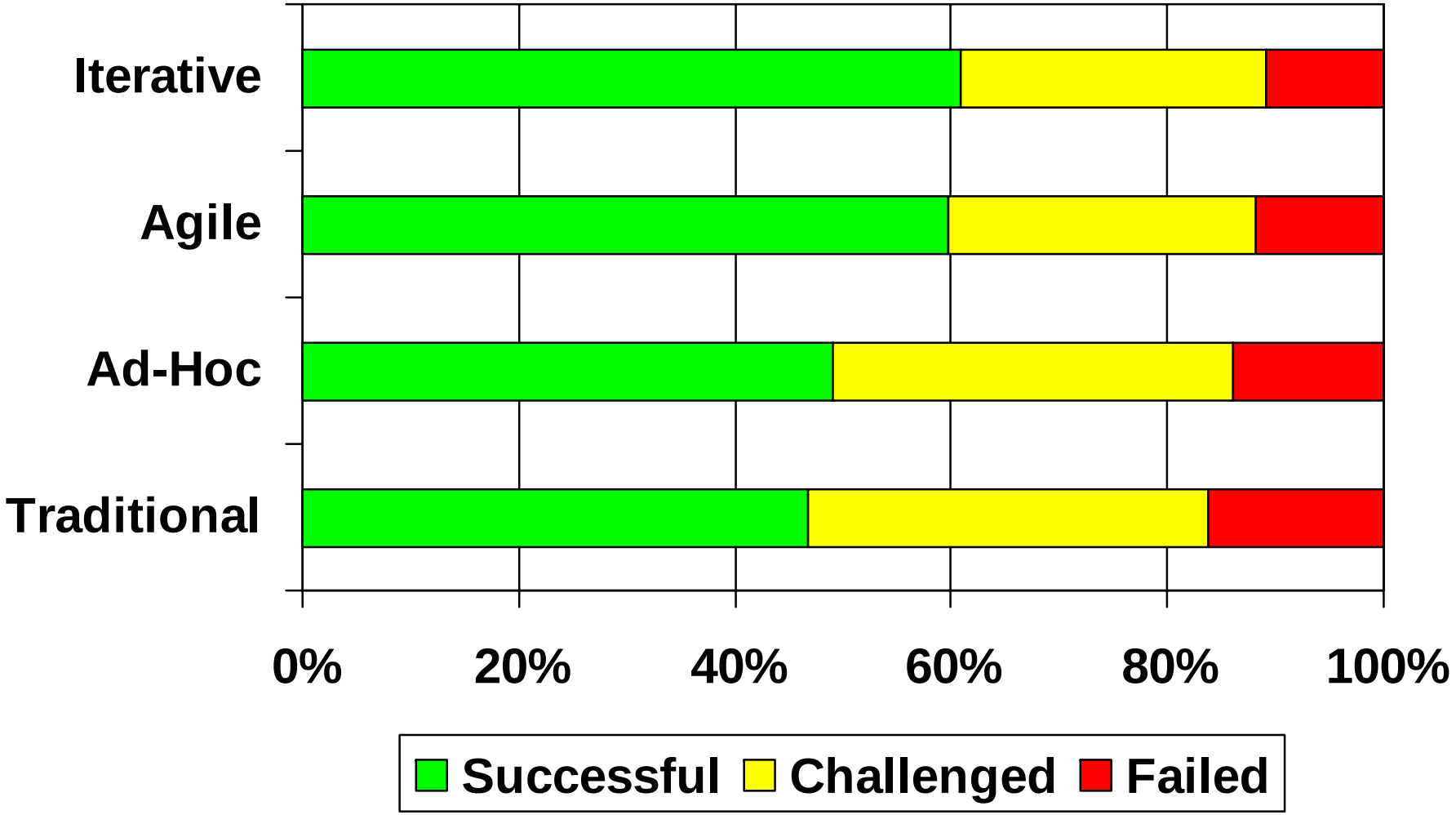
Agenda



1. Agility and requirements
2. Introduction to agility@scale
3. Agile requirements at scale
4. **Parting thoughts**

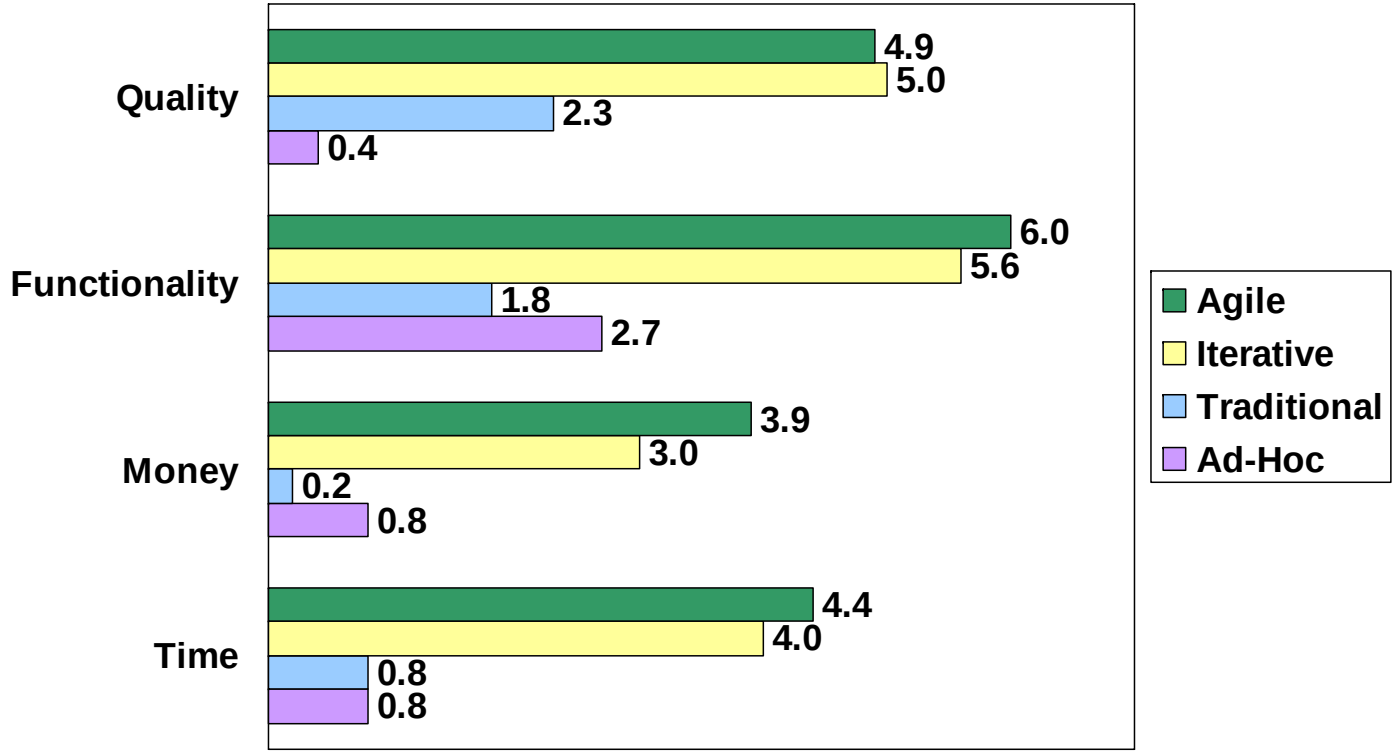
IBM agility@scale™

Why agile?



Source: 2010 Project Success Survey, www.ambysoft.com/surveys/

Why agile?



Agile teams produce higher quality work, are quicker to deliver, are more likely to deliver the right functionality, and more likely to provide greater ROI than traditional teams

Source: 2008 Project Success Survey, www.ambysoft.com/surveys/

Implications for Business Analysts

- Your goal is to build a shared understanding, it isn't to write detailed documentation
- A critical success factor is to use inclusive modeling techniques which enable active stakeholder participation
- Expand your horizons and become a generalizing specialist
- Learn how to perform acceptance TDD so that you can capture requirements as executable specifications
- Recognize that one process size does not fit all, that you will need to be flexible
- Your primary goals should be to:
  - Facilitate communication between stakeholders and developers
  - Put developers in direct contact with stakeholders wherever possible
  - Help developers learn better communication skills

Implications for Organizations

- Don’t be fooled by the agile rhetoric
  - You still need to invest in modeling
  - You still need to invest in requirements management
- Don’t be fooled by the traditional rhetoric
  - Detailed documentation adds risk to IT projects
- Individual teams find themselves in unique situations, so will have unique tailorings of your process
  - Don’t sub-optimize around a “standard” process in a naïve attempt at IT governance

www.ibm/software/rational

www.ibm.com/developerworks/mydeveloperworks/blogs/ambler/

Twitter: scottwambler