IBM Software Group

Scaling Scrum: Lessons from the Trenches

Scott W. Ambler
Chief Methodologist/Agile

Rational. software

IBM agility@scale™

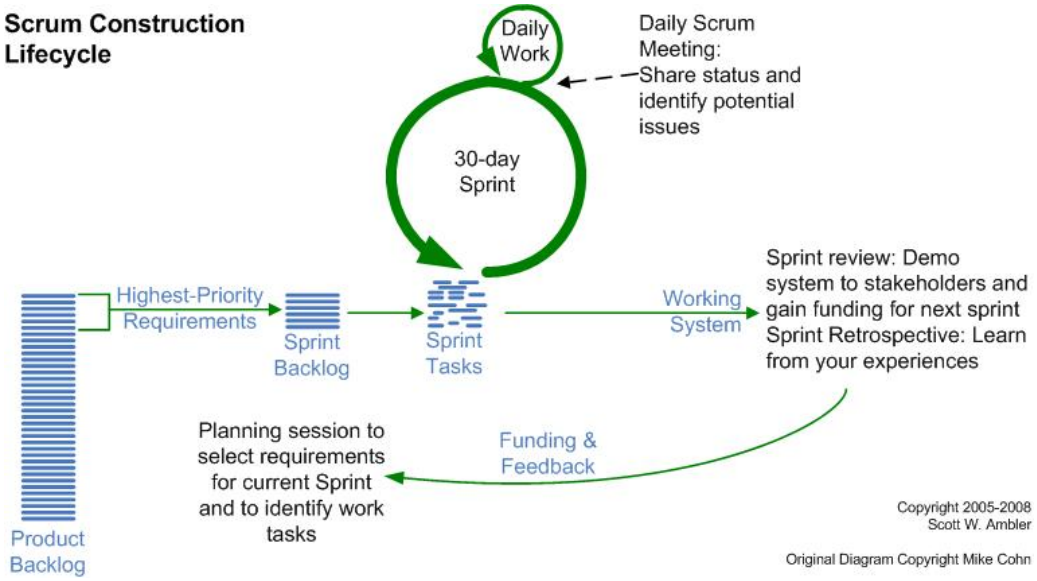
Agenda

- Introduction to Scrum
- The Bigger Picture
- Scaling the Scrum Life Cycle
- Scaling Scrum Roles
- Scaling Scrum Practices
- Parting Thoughts

Agenda

- Introduction to Scrum
  - Scrum life cycle
  - Scrum roles
  - Scrum practices
  - Examining Scrum
- The Bigger Picture
- Scaling the Scrum Life Cycle
- Scaling Scrum Roles
- Scaling Scrum Practices
- Parting Thoughts

The Scrum Life Cycle



Scrum Roles

- Scrum Master
  - Leader/coach
  - Facilitates scrum meetings such as sprint planning, sprint demo, and daily Scrum meetings
- Product Owner
  - Represents the stakeholders
  - Owns the product backlog, including the requirements
  - Provides information and makes decisions in a timely manner
  - “The one neck to wring”
- Developer/Team Member
  - Anyone else on the team

Scrum Practices

- Product Backlog – Prioritized stack of requirements
- Value-Driven Lifecycle – Deliver potentially shippable software each sprint
- Self Organization – The people who do the work are the ones that plan and estimate it
- Release Planning – Develop and then maintain a high-level project plan
- Sprint Planning – At the beginning of a sprint the team plans in detail what they will deliver and how they will do the work
- Daily Scrum Meeting – Each day hold a 15 minute coordination meeting
- Sprint Demo – At the end of the sprint demo what you've built to key stakeholders
- Reflections – At the end of the sprint the team identifies potential improvements to the way that they work

Examining Scrum

Advantages

- Solid, proven kernel for effective software development
- #1 Agile software development framework used today
- Simple framework into which good practices "plug in"
- Focuses on delivering value in 2 or 4 week cycles
- Defines key points where team engages stakeholders

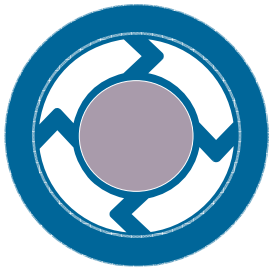
But it's missing…

- Guidance on scaling factors
- How to coordinate large-scale project to deliver value each Sprint
- How to prepare large-scale Scrum Teams to work together
- How to work together as a global Scrum Team
- How to deal with many other issues as we'll soon see

Agenda

- Introduction to Scrum
- The Bigger Picture
  - Agile Scaling Model (ASM)
  - Agility@Scale
  - The “5Ps” of IT
- Scaling Scrum Roles
- Scaling the Scrum Life Cycle
- Scaling Scrum Practices
- Parting Thoughts

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

Agile scaling factors

Team size

Under 10 developers ⟷ 1000's of developers

Compliance requirement

Low risk ⟷ Critical, audited

Geographical distribution

Co-located ⟷ Global

Domain Complexity

Straight -forward ⟷ Intricate, emerging

Disciplined Agile Delivery

Enterprise discipline

Project focus ⟷ Enterprise focus

Organization distribution (outsourcing, partnerships)

Collaborative ⟷ Contractual

Organizational complexity

Flexible ⟷ Rigid

Technical complexity

Homogenous ⟷ Heterogeneous, legacy

The “5Ps” of IT: Look at the Big Picture

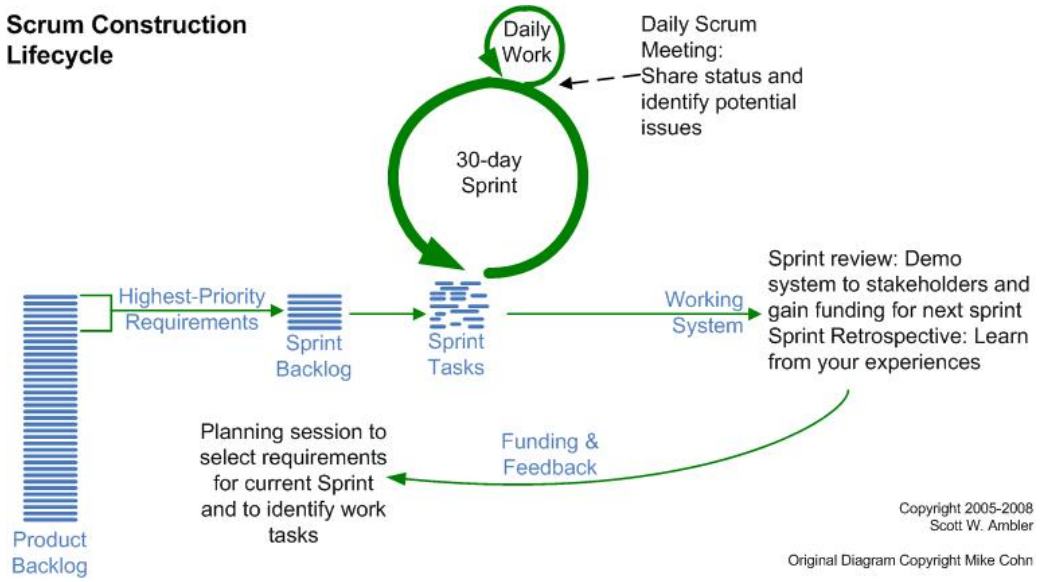
To achieve systemic improvement within an IT organization, you need to address five primary issues:

1. People. Solution delivery is a “team sport”, individuals and the way that they work together are the primary determinants of success on most projects.
2. Principles. You need a consistent and coherent philosophical foundation which reflects your values as an organization to guide your decisions.
3. Practices. Practices are contextual, describing how people work together to achieve a goal.
4. Products. The tools and technologies which people use to perform their work.
5. Process. The “glue” which pulls everything together.

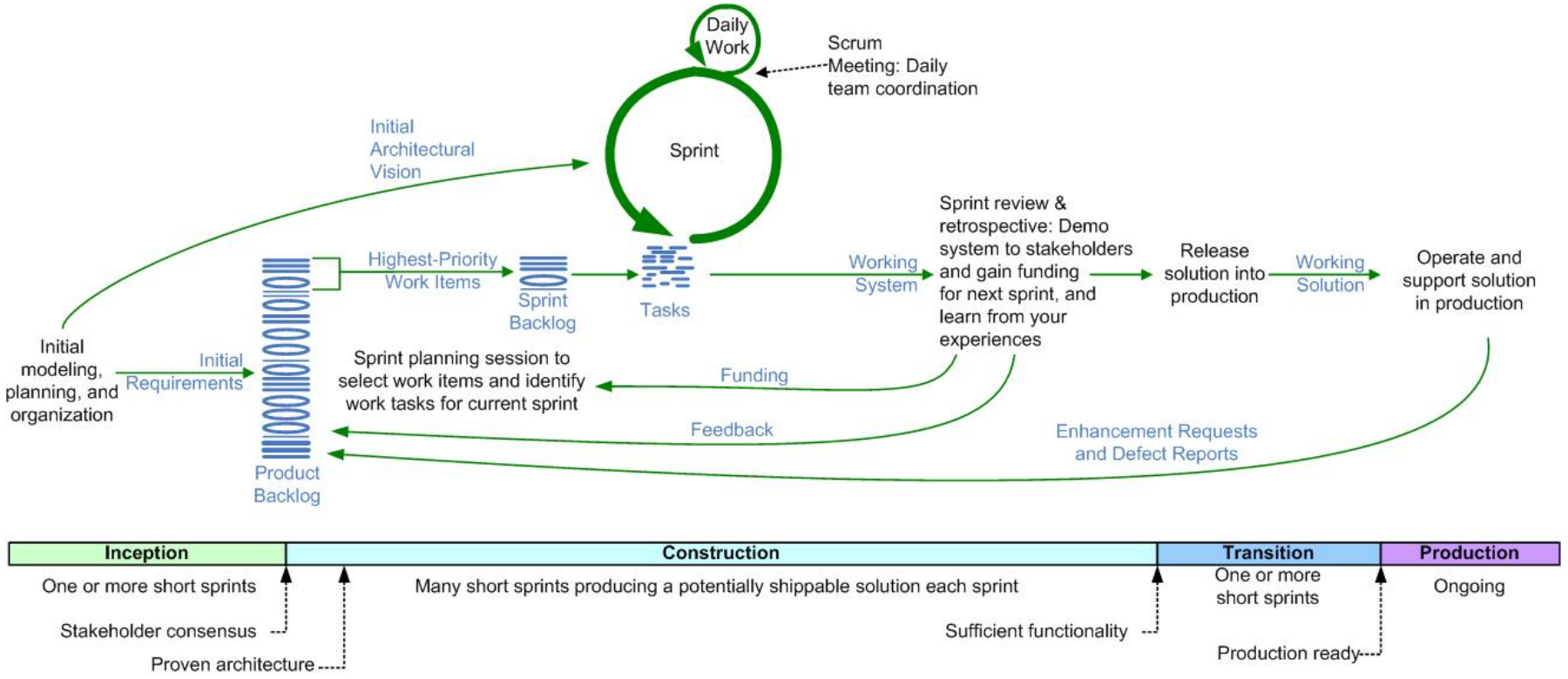
Agenda

- Introduction to Scrum
- The Bigger Picture
- Scaling the Scrum Life Cycle
- Scaling Scrum Roles
- Scaling Scrum Practices
- Parting Thoughts

The agile construction lifecycle



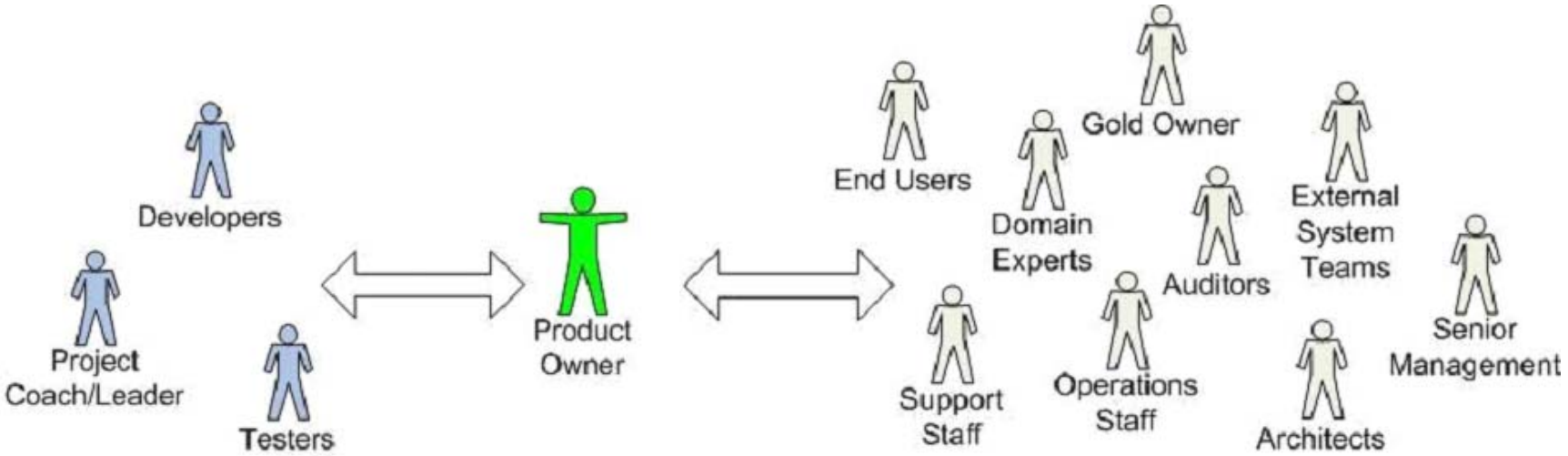
The disciplined agile delivery life cycle



Agenda

- Introduction to Scrum
- The Bigger Picture
- Scaling the Scrum Life Cycle
- Scaling Scrum Roles
  - Product Owner
  - Coordination Roles
- Scaling Scrum Practices
- Parting Thoughts

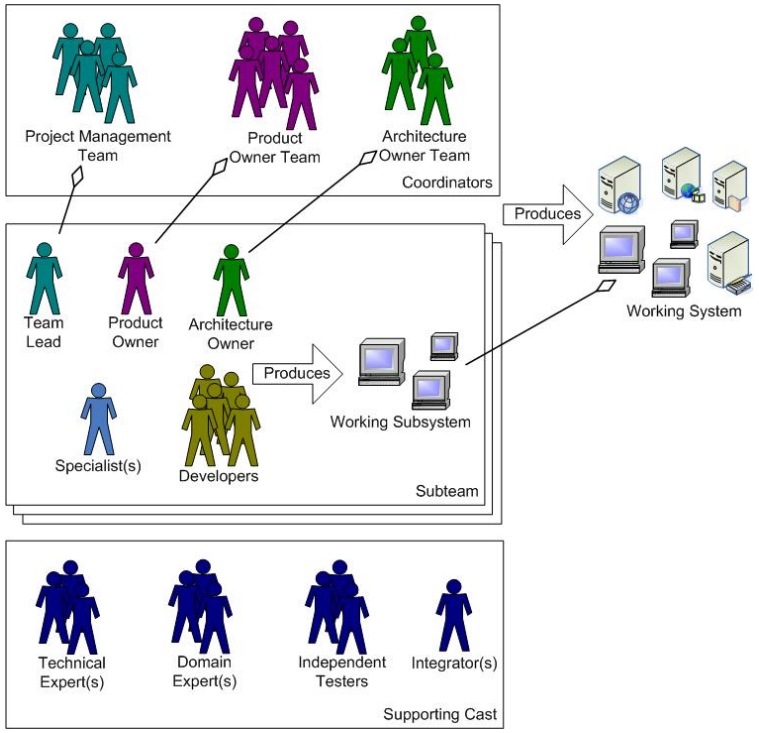
Scaling Product Owner



- Product owner is really a communication conduit between the team and stakeholders
  - Owns the work item list, including the requirements
  - Must have agile business analysis skills
  - Will often work with business analysts who elicit requirements from others
  - PO gets the team access to the relevant stakeholders just in time
  - Needs to negotiate, negotiate, negotiate

Large Teams Require Additional Roles

- Architecture Owner
  - Facilitates technical decisions
  - Coordinates technical discussions between teams
- Domain Expert
  - Has detailed knowledge about one or more aspects of the problem domain
- Technical Expert
  - Has detailed technical knowledge needed for short period
- Independent Tester
  - Focuses on complex testing efforts, working parallel but independent of the team
- Integrator
  - Responsible for building the entire system from its various subsystems
- Specialist
  - Sometimes component subteams require people focused on narrow specialties
  - E.g. Business analysts, security experts, database administrators, usability professionals



Note: Many of these additional roles may be needed on smaller teams too!

Agenda

- Introduction to Scrum
- The Bigger Picture
- Scaling the Scrum Life Cycle
- Scaling Scrum Roles
- Scaling Scrum Practices
- Parting Thoughts

Scaling Daily Stand Up Meetings





- Geographic distribution
  - Meeting occurs over phone, video, electronically…
  - Rational Team Concert (RTC) to share information
  - Change meeting times to reflect team distribution – spread the pain
- Team size
  - Kanban strategy is to ask 1 question: What new issues do you foresee?
  - Subteams need to coordinate via coordinators, perhaps in a “scrum of scrums”
- Regulatory compliance
  - Take meeting attendance and record action items (if any)
- Organizational distribution
  - Additional coordination between organizations may be required
  - Project dashboard access for external organizations may be required
  - Document decisions/action items pertaining to external organizations
- Enterprise discipline
  - Enterprise

Scaling release planning

- Geographic distribution
  - Plan needs to be captured electronically so that everyone has access
- Team size
  - Sprint lengths/rhythms of the subteams should be a divisor of the larger team
  - Subteams may need their own release plans
  - Overall plan must reflect the plans of the subteams
- Regulatory compliance
  - Plan, and updates to it, may need to be documented
- Organizational distribution
  - Planning across multiple organizations will potentially take longer and require greater detail
- Technical complexity
  - Dependencies on other teams are critical and need to be reflected in release plan, particularly non-agile teams which have lower chances of on-time delivery
- Organizational complexity
  - Dependencies on non-agile teams may require changes in the strategies of all teams involved
- Enterprise discipline
  - Release plan must reflect portfolio-level and product-level plans

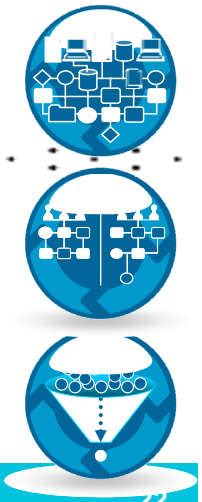
Scaling sprint planning




- Geographic distribution
  - Need to capture the plan electronically
- Team size
  - Each sub-team is responsible for its own iteration planning
  - Scrum Masters need to be aware of major dependencies with other subteams, particularly when the other subteams have different iteration lengths/schedules
  - Teams needs to be aware of dependencies between work items lists
  - More likely the teams will need to meet prior to the Sprint Planning to review user stories, disaggregate, estimate discuss dependencies, duplications across stories,…
- Regulatory compliance
  - Sprint plans may need to be documented
- Domain complexity
  - Teams need to engage in lookahead planning, not just plan for upcoming sprint
- Technical complexity
  - Teams need to be aware of technical dependencies between subsystems
  - Teams need to engage in lookahead planning, not just plan for upcoming sprint

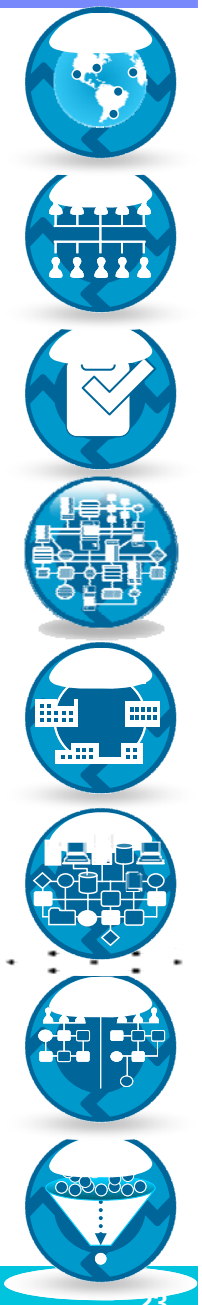
Scaling self organization

- Disciplined agile delivery
  - Disciplined agile developers are self organizing within an appropriate governance framework
- Regulatory compliance
  - Plans, estimates, and so on may need to be recorded
- Organizational complexity
  - May need to educate management team on collaborative leadership and facilitation
  - May need to provide education to help others shift from command-control to self-organizing
- Enterprise discipline
  - Application architecture decisions are constrained by enterprise infrastructure and futures directions
  - Application architecture enhanced by leveraging existing infrastructure and reusable assets
  - Release plan must reflect portfolio and project plans
  - Agile teams should follow enterprise-level guidelines (e.g. coding standards, data standards, UI standards, …)
  - Adopt a Lean Development Governance strategy (see paper by Ambler and Kroll)



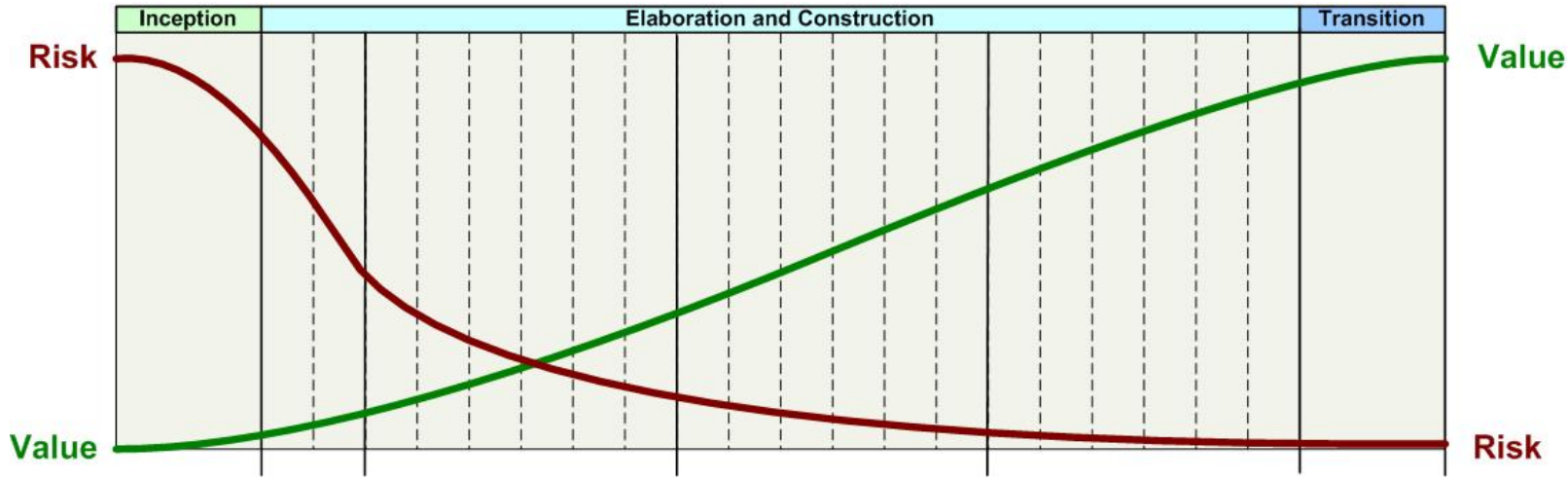
Scaling product backlogs

- Disciplined agile delivery
  - Defects treated like requirements and managed on backlog
  - Non-functionality work items, such as training, reviews, can be managed on backlog
- Geographic distribution
  - Manage the backlog electronically
- Team size
  - Subteams may have their own backlogs, but that makes rollups harder
  - Burndowns of subteams need to be rolled up into overall team burndown
- Regulatory compliance
  - May need to manage backlog electronically
- Domain complexity
  - Business analysts look ahead on the product backlog to explore upcoming complexities
- Organizational distribution
  - A given organizational unit may only be allowed to see portions of the backlog
- Technical complexity
  - Team members look a bit ahead on stack to consider upcoming complexities
- Organizational complexity
  - Your team may need to conform to existing change management processes
- Enterprise discipline
  - Electronic backlog management enables automation of burndown charts and other metrics via project dashboard (e.g. in Rational Team Concert), supporting improved governance



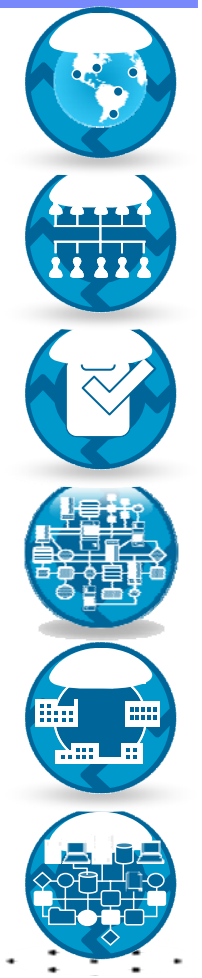
Scaling Value Lifecycle to Risk-Value Lifecycle

- Provides the extended team with explicit milestones centered on balancing risk mitigation and value creation
- Observations:
  - Key stakeholders frequently do not have time to carefully review and discuss the results of every iteration.
  - Iteration demos are still a good idea for getting feedback.
  - Fewer key milestones where go/no-go decisions are made are actually needed.



Scaling Sprint Demos

- Geographic distribution
  - Demos occurs physically where possible, but transmitted electronically
  - Change demo times to reflect team distribution – spread the pain
  - Greater need to schedule the demos ahead of time
- Team size
  - Prioritize which subteam(s) should demo to the entire team
  - Individual subteams should do their own demos to their smaller community
- Regulatory compliance
  - Take meeting attendance and record action items (if any)
- Domain complexity
  - Invite a wider range of stakeholders required, not just insiders
- Organizational distribution
  - Invite stakeholders from each organization unit
  - Separate demos may be required by some organization units specific to them
- Technical complexity
  - Some of the work may not be visible through user interface, so may need to do a technical walkthrough instead

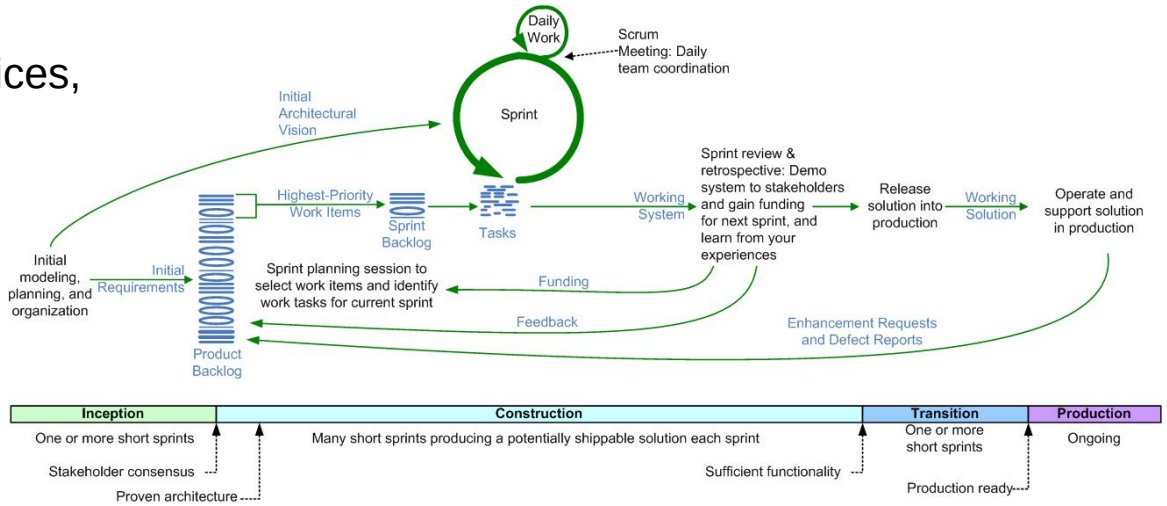


Scaling reflections/retrospectives

- Disciplined agile delivery
  - Track your progress with IBM Rational SelfCheck
- Geographic distribution
  - Hold retrospectives electronically to include everyone
- Team size
  - Subteams should hold their own retrospectives
  - Subteams should share ideas with each other
- Regulatory compliance
  - May need to record any changes to your process
- Organizational distribution
  - May not be allowed to share improvements between organizations
- Organizational complexity
  - Existing process improvement strategies may focus on reviews (post mortems) at the end of the project
  - Process improvement may be “owned” by a specific process group
- Enterprise discipline
  - Consider an agile center of competency (CoC) to share ideas

Other Practices

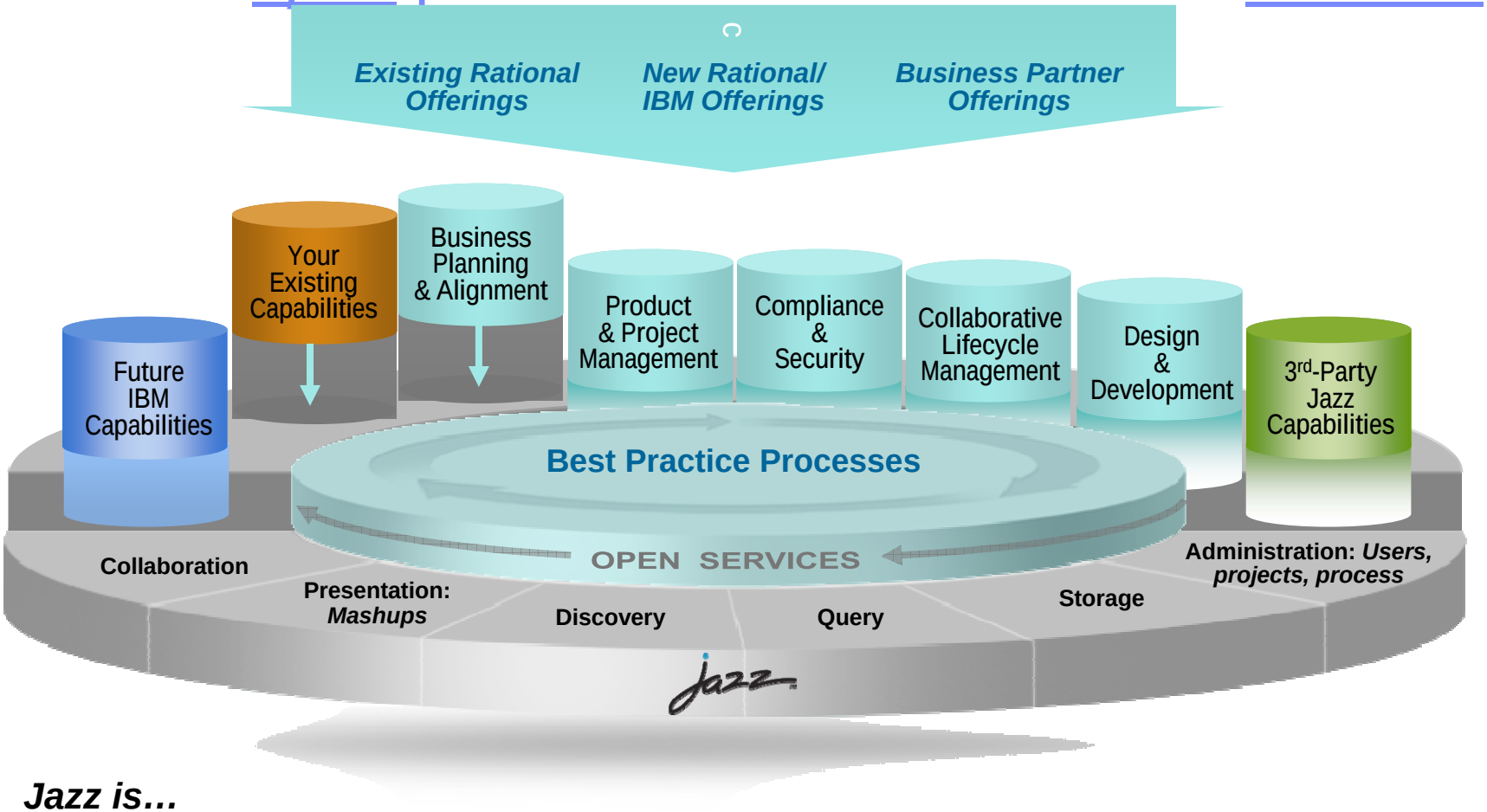
- At the delivery level:
  - Development techniques, www.eclipse.org/epf/
  - Agile modeling and documentation practices, www.agilemodeling.com
  - Agile data and database practices, www.agiledata.org
  - And many more…
- At the enterprise level:
  - Portfolio management
  - Strategic reuse
  - Enterprise architecture
  - Enterprise administration
  - Enterprise business modeling
  - Operations and support
  - www.enterpriseunifiedprocess.com



Agenda

- Introduction to Scrum
- The Bigger Picture
- Scaling the Scrum Life Cycle
- Scaling Scrum Roles
- Scaling Scrum Practices
- Parting Thoughts

Jazz is an *open* platform with a shared set of *services*



Jazz is…

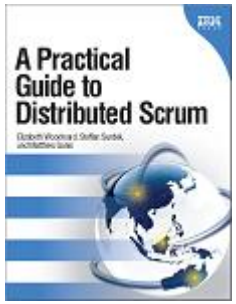
- A scalable, extensible team collaboration platform which supports Scrum development
- A community at Jazz.net, where you can see Jazz-based products being built
- An integration architecture, enabling mashups and non-Jazz based products to participate

Why IBM?

- Our integrated tooling based on the Jazz platform enables disciplined agile software development
- Our Measured Capability Improvement Framework (MCIF) service offering helps organizations to successfully improve their IT practices in a sustained manner
- We are one of the largest agile adoption programs in the world
- We understand the enterprise-level issues that you face
- We scale from pilot project consulting to full-scale agile adoption
- Our Accelerated Solutions Delivery (ASD) practice has years of experience delivering agile projects at scale

Critical resources

- www.ibm.com/rational/agile/
- www.jazz.net
- www.ibm.com/developerworks/
- www.ibm.com/developerworks/blogs/page/ambler



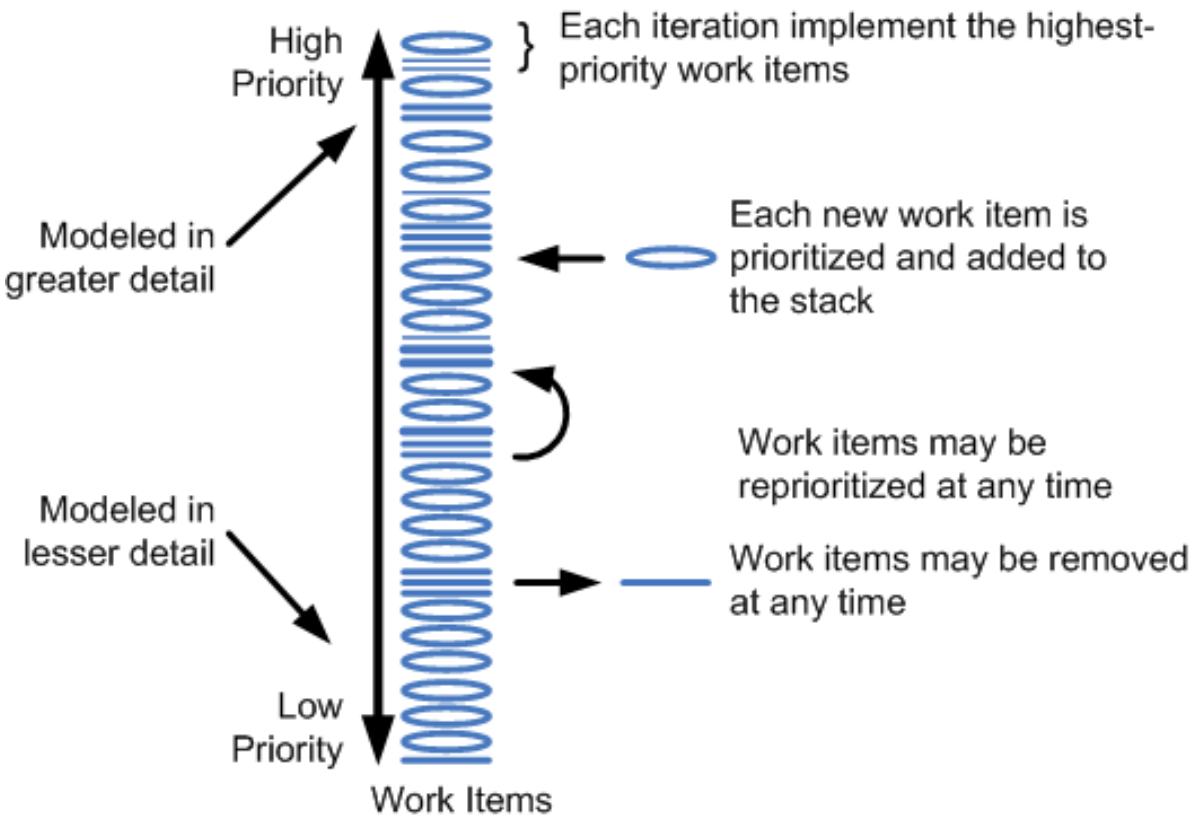
IBM Software Group

Backup slides

Rational. software

IBM agility@scale™

Scaling Product Backlogs: Work Item Lists



Development teams do more than just implement requirements, they also fix defects, go on training, review the work of other teams, and so on. All of this work needs to be taken into account when planning.

Smart teams look a few iterations down the stack for complex work items and then mitigate the risk by modeling a bit ahead.

www.jazz.net

Scaling Self-Organization

- Lean development governance
  - Teams don't work in a vacuum
  - Self-governance must be constrained by enterprise goals and environment
  - *Lean Development Governance* paper by Ambler and Kroll, 2008
- Automated measurements and project dashboards
  - It is possible to streamline much of the Scrum bureaucracy through automation
  - This is true empiricism, not just marketing rhetoric
  - Rational Team Concert (RTC) for project dashboard
  - Rational Insight for portfolio-level dashboard
- Adopt corporate development conventions
  - "Enforce" and monitor via static analysis tools