Introduction to Disciplined Agile Delivery (DAD)

Mark Lines
Calgary, Alberta, Canada
Co-founder, UPMentors
Mark@UPMentors.com
www.DisciplinedAgileDelivery.com
twitter.com/Mark_Lines

Agenda

- Defining Disciplined Agile Delivery (DAD)
- People first
- Learning oriented
- Hybrid agile framework
- A risk-value driven lifecycle
- Goals driven lifecycle
  - Inception
  - Construction
  - Transition
- Enterprise aware
  - Optimize the whole
  - Agile governance
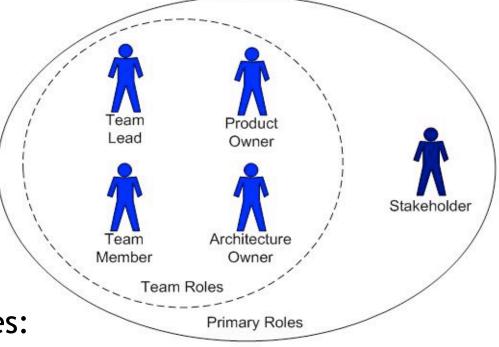- Agility@scale
- Questions

- The Disciplined Agile Delivery (DAD) process framework is a hybrid agile approach to IT solution delivery that is:
  - People-first
  - Learning-oriented
  - Risk and value driven
  - Goal-driven
  - Scalable
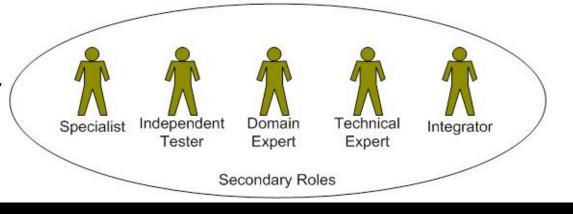 - Enterprise aware

- Primary roles:
  - Stakeholder
  - Team Lead
  - Product Owner
  - Agile Team Member
  - Architecture Owner

- Secondary/optional roles:
  - Domain Expert
  - Technical Expert
  - Independent Tester
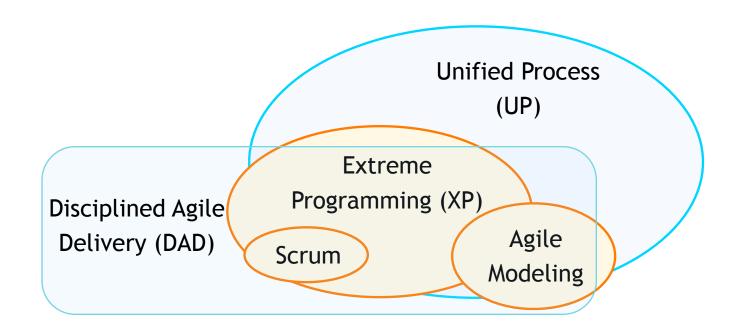 - Integrator
  - Specialist

Learning oriented

- Domain learning
  - Initial requirements envisioning
  - Incremental delivery of a potentially consumable solution
  - Active stakeholder participation throughout lifecycle
- Process improvement
  - Retrospectives at the end of an iteration
  - Tracking of improvements
  - Sharing of skills through non-solo development
- Technical learning
  - Architecture spikes
  - Proving the architecture with working code
- General strategies
  - Training
  - Education
  - Mentoring/coaching
  - Individuals are generalizing specialists, not just specialists

Unified Process (UP)

Disciplined Agile Delivery (DAD)

Extreme Programming (XP)
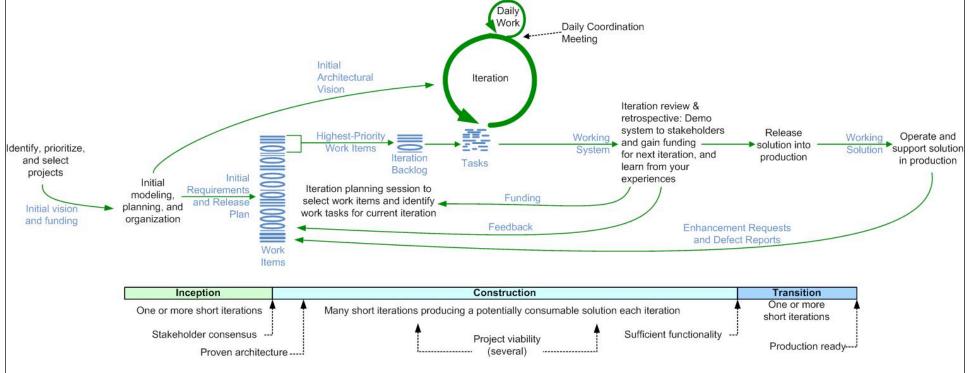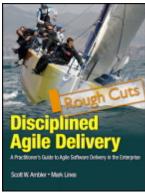
Scrum

Agile Modeling

DAD is a hybrid process framework adopt best practices and philosophies from several methodologies

The Disciplined Agile Delivery life cycle

The Disciplined Agile Delivery (DAD) process framework is a people-first, learning-oriented hybrid agile approach to IT solution delivery. It has a risk-value lifecycle, is goal-driven, scalable, and is enterprise aware.

Goals for the Inception Phase	Goals for Construction Phase Iterations	Goals for the Transition Phase
- Identify the vision for the project - Bring stakeholders to agreement around the vision - Align with enterprise direction - Identify initial technical strategy, initial requirements, and project plan - Set up the work environment - Form initial team - Secure funding - Identify risks	- Produce a potentially consumable solution - Address changing stakeholder needs - Move closer to deployable release - Maintain or improve upon existing levels of quality - Address highest risk(s)	- Ensure the solution is production ready - Ensure the stakeholders are prepared to receive the solution - Deploy the solution into production

Ongoing Goals

- Fulfill the project mission
- Grow team members skills
- Enhance existing infrastructure

- Improve team process and environment
- Leverage existing infrastructure

UPM >
UNIFIED PROCESS MENTORS
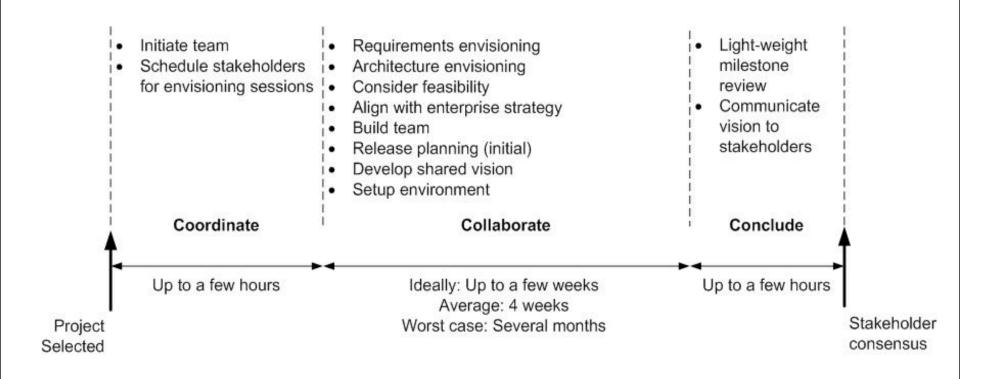
- The coordinate-collaborate-conclude rhythm occurs at several scales on a disciplined agile delivery (DAD) project:

Release rhythm	**Inception** Coordinate	**Construction** Collaborate	**Transition** Conclude

Iteration rhythm	**Iteration Planning** Coordinate	**Development** Collaborate	**Stabilize** Conclude

Daily rhythm	Coordination Meeting Coordinate	Daily work Collaborate	Stabilize Conclude

- Initiate team
- Schedule stakeholders for envisioning sessions

- Requirements envisioning
- Architecture envisioning
- Consider feasibility
- Align with enterprise strategy
- Build team
- Release planning (initial)
- Develop shared vision
- Setup environment

- Light-weight milestone review
- Communicate vision to stakeholders

Coordinate **Collaborate** **Conclude**

Project Selected

Up to a few hours

Ideally: Up to a few weeks
Average: 4 weeks
Worst case: Several months

Up to a few hours

Stakeholder consensus

- Prove the architecture works via end-to-end working slice of the solution

- Incrementally produce a consumable solution
- Share project status with stakeholders
- Align with organizational goals
- Align with other project teams
- Improve individual and team performance

- Determine sufficiency
- Harden the solution

Coordinate

Collaborate

Conclude

Stakeholder Consensus

Typical: 1 iteration
Worst case: Many iterations

Proven Architecture

Several iterations

Ideally: Several hours

Sufficient Functionality

- Iteration planning
- Iteration modeling

"Standard" practices:
- Visualize work
- Daily coordination meeting
- Refactoring
- Developer regression testing
- Model storming
- Continuous integration (CI)
- Sustainable pace
- Prioritized requirements
- Architecture spike
- Configuration management
- Burn-down chart
- Automated metrics

"Advanced" practices:
- Test-driven development (TDD)
- Acceptance TDD (ATDD)
- Continuous deployment (CD)
- Look-ahead modeling
- Parallel independent testing
- Continuous documentation
- Non-solo development
- Look-ahead planning

- Iteration demo
- Retrospective
- Release planning (update)
- Consider sufficient functionality

Coordinate

Collaborate

Conclude

Typical: One to four weeks
Average: Two weeks
Worst case: Six weeks

2 hours for each week of the iteration length

One hour per week of iteration length

Iteration start

Potentially consumable solution

- Phase planning

- Transition planning
- End-of-lifecycle testing and fixing
- Pilot/beta the solution
- Finalize documentation
- Communicate deployment
- Train/educate stakeholders

- Production readiness review
- Deploy solution

Coordinate

Collaborate

Conclude

Ideally: Nothing
Typical: One hour per week of collaborate time

Ideally: Nothing
Average: 4 weeks
Worst case: Several months

Ideally: Less than an hour
Worst case: Several months

Sufficient Functionality

Production Ready

- Follow corporate conventions:
  - Standards and guidance for the architecture
  - Coding standards
  - Data guidelines
  - User interface (UI) guidelines
  - And many more

- Enhance the organizational ecosystem:
  - Reusing and leveraging the existing infrastructure is great
  - Enhancing and building out the infrastructure is better
  - Work closely with the enterprise architecture (EA) team

- Share learnings:
  - Personal and team improvement is great
  - Organization-level improvement is better
  - Your agile center of competency should harvest and share improvements

- Interact with other (potentially non-agile) teams:
  - Enterprise architecture
  - Data management
  - Governance
  - Quality assurance
  - Project management office (PMO)

UPM >
UNIFIED PROCESS MENTORS

- Agile teams provide:
  – Significantly greater visibility to stakeholders regarding their actual status
  – Many more opportunities for stakeholders to steer the project
  – BUT… require stakeholders to be actively involved and accountable

- Practices:
  – Active stakeholder participation
  – Potentially consumable solutions every iteration
  – Risk-value lifecycle
  – Explicit, light-weight milestone reviews
  – Daily coordination meetings
  – Iteration demos
  – All-hands demos
  – Follow enterprise development guidance
  – Work closely with enterprise architects
  – Automated metrics gathering

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

UPM >
UNIFIED PROCESS MENTORS

- The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments
  - ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/raw14204usen/RAW14204USEN.PDF

- Scaling Agile: An Executive Guide
  - ftp://public.dhe.ibm.com/common/ssi/sa/wh/n/raw14211usen/RAW14211USEN.PDF

- Improving Software Economics: Top 10 Principles of Achieving Agility at Scale
  - ftp://public.dhe.ibm.com/common/ssi/ecm/en/raw14148usen/RAW14148USEN.PDF

- Enable the Agile Enterprise Through Incremental Adoption of Practices
  - http://public.dhe.ibm.com/common/ssi/ecm/en/raw14077usen/RAW14077USEN.PDF

- ## DAD Training
  - IBM's Advanced Disciplined Agile delivery workshop: 3 days (21 PDUs)
    - google "RP252 DAD"
    - (PMI approved, registered under provider number 1107)
  - The Disciplined Agile Delivery Experience: 3 days
    - www.UPMentors.com/courses

- ## DAD Services
  - DAD Quick Start with IBM Rational Team Concert (RTC)
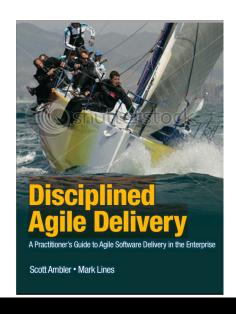 - DAD w/ Atlassian Coaching, Consulting, Workshops
    - www.AppFusion.com/Agile

- ## DAD Products
  - The DAD process template for IBM Rational Team Concert (RTC)

- ## Or even better....
  - www.DisciplinedAgileDelivery.com

- IBM's approach to Agile development
  - written by Mark Lines & Scott Ambler for IBM Press
  - Adds enterprise discipline to Scrum, Extreme Programming/XP and other agile techniques
  - available Q1 2012 on Amazon etc.
  - some chapters now available of Safari Roughcuts for review
- www.DisciplinedAgileDelivery.com
  - the DAD community website
  - *join the discussion today!!*

Disciplined Agile Delivery
A Practitioner's Guide to Agile Software Delivery in the Enterprise

Scott Ambler • Mark Lines

Mark@UPMentors.com

Julian Holmes, Co-founder UPMentors, Julian@UPMentors.com (UK)

twitter.com/scottwambler, twitter.com/Mark_Lines

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

www.ibm.com/rational/agile