

Introduction to Agile

Marty Acks

MKS Inc.

June 2, 2011

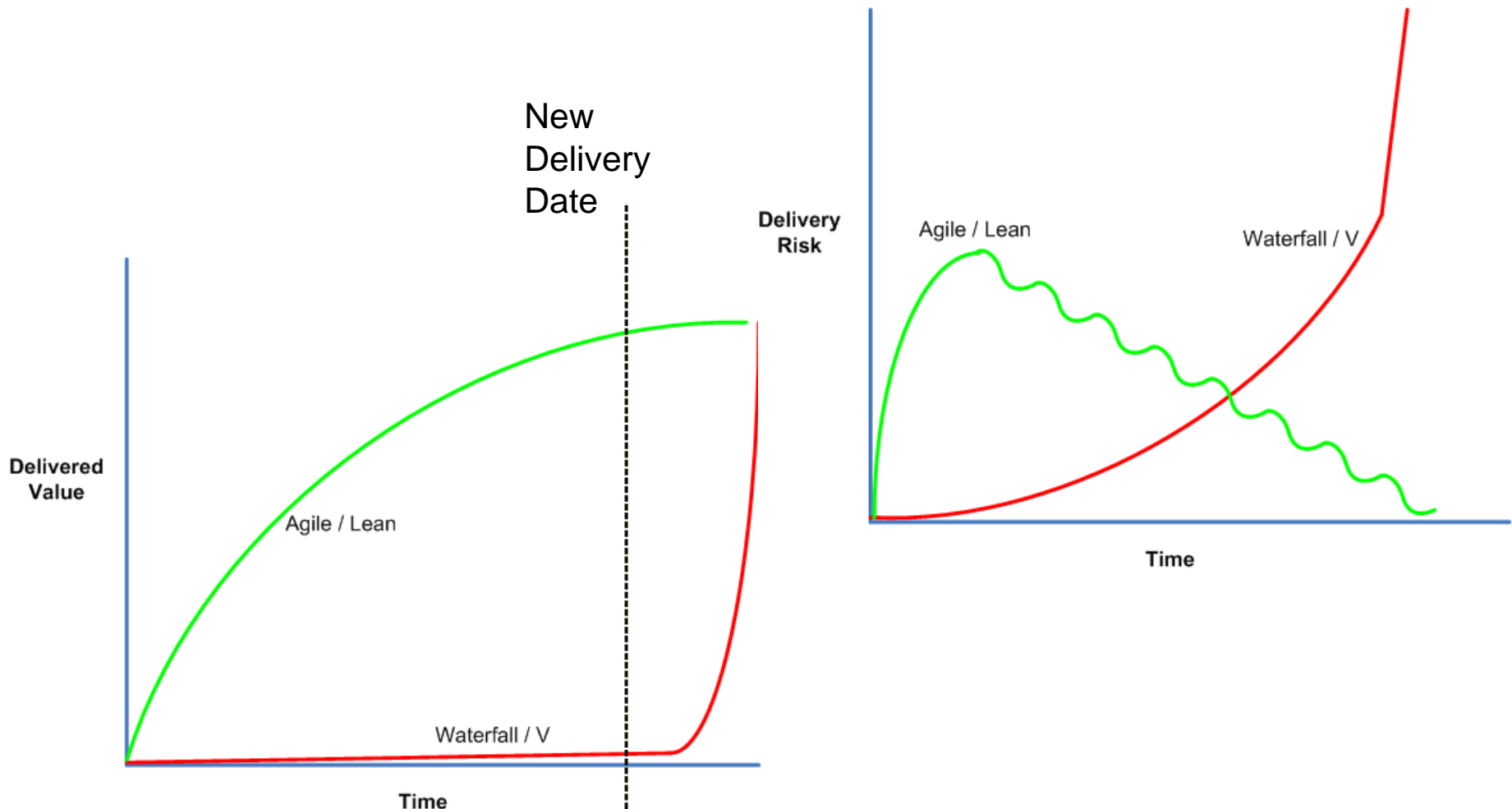


MKS

Agenda

- What makes Agile compelling?
- Agile in the enterprise
 - Challenges in the enterprise
 - Highly regulated enterprise environments
- Strategy for Agile adoption in the enterprise
- Agile at MKS
- Summary
- Q & A

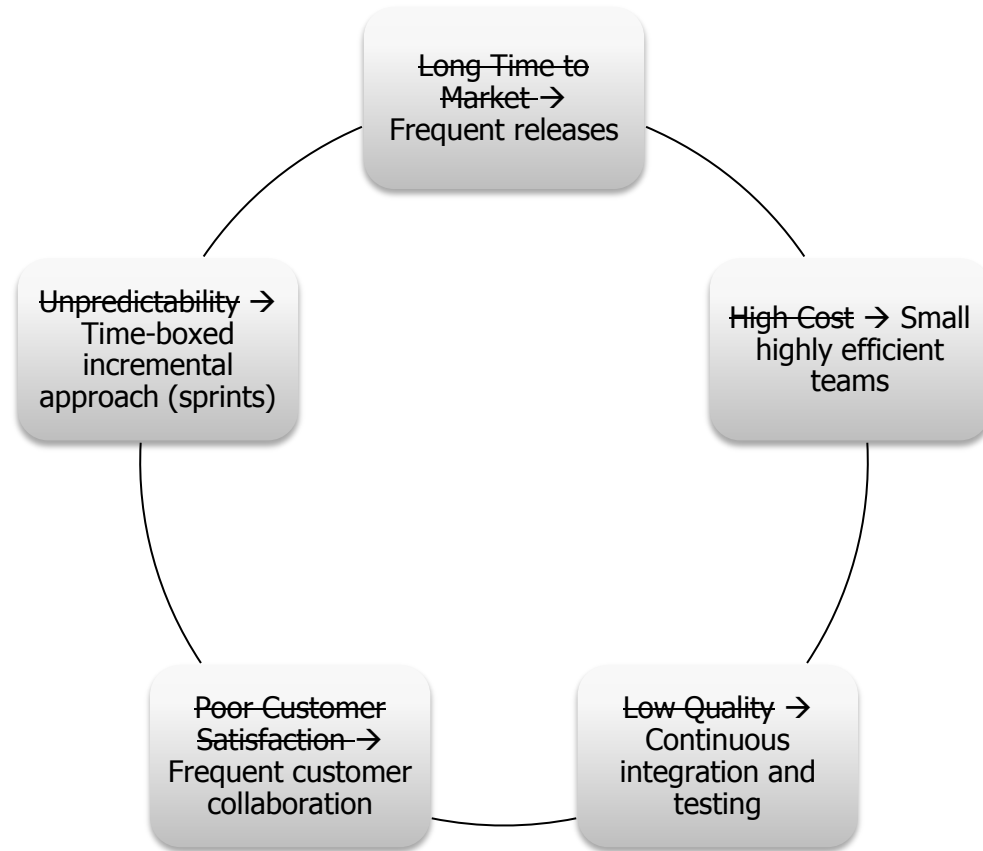
What Makes Agile Compelling: Agile vs Plan-Driven Project Management



Specific Customer Challenges



What Makes Agile Compelling



Agile in the Enterprise



MKS

Enterprise Challenges

Large distributed teams

Application inter-dependencies

Compliance and governance

Phased transformation

PMO – monitoring and WBS

Multiple product lines and variants



Highly Regulated Enterprise Environments

- Tend to follow a Waterfall, V-Model or Spiral Model development life cycle
 - Some standards (e.g. EN 61508) explicitly require the V-Model as the basis for the Systems or Safety life cycle
 - While many standards imply the use of such models for the software development life cycle, the majority explicitly avoid mandating a particular model
- Tend to be plan-driven, with a formal Project Management Office
 - Change management is a formal, high-ceremony process
- Easy to fall into the trap of being overly artifact driven and reliant on gated reviews to catch issues & update “the big picture”
 - Lends itself to relatively isolated silos by discipline

Highly Regulated Enterprise Environments

- Significant systems engineering and analysis done up-front
 - More of the software requirements are well defined at the start of development than is the case for typical software projects
- Tend to be a significant number of non-functional constraints and more dependencies between requirements
 - Constraints apply across many or all functional requirements
- Mandates traceability from high level requirements to all other development artifacts
 - Strongly recommends decomposition & transformation (allocation) approach to design & implementation
- Mandates requirements based verification & validation
 - Must show that software meets all requirements and only implements the requirements

Strategies for Adopting Agile in the Enterprise



MKS

Considerations for Making Enterprises More Agile

- Cultural barriers: “But that’s how we have always done it...”
- Misperceptions that “Agile” means a lack of discipline, rigor, or defined practices
- Agile is relatively untried and unproven in highly regulated software development environments
 - Groups like the SEI, USAF STSC are starting to change that
- Senior management requires proof of success before implementing Agile beyond a single project

Approach #1 – Creative Re-Interpretation

- Rather than trying to implement Agile as the overall development lifecycle, incorporate Agile practices and values within the existing framework (ex.V-Model)
- Use the constraints of the environment (emphasis on risk management & quality) to sell such practices and approaches
- Adopting an Agile or Lean approach involves taking a more holistic view of application development

Approach #2 – Adapt Agile to the Environment

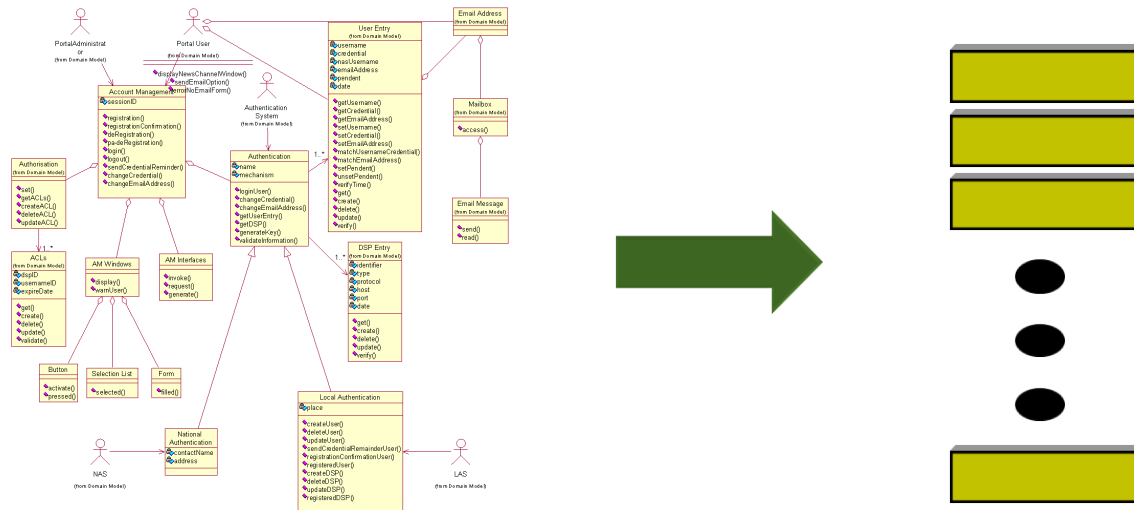
- Most Software Development Life Cycle (SDLC) standards do not prescribe a particular lifecycle model; they identify the processes and activities such a model must address
 - Make use of the CMMI-Dev process model to identify where additional rigor or a hybrid mix of traditional and Agile practices can fully address development standards [SEI-CMMI]
- Use tooling to address traceability and review requirements
 - Application Lifecycle Management (ALM) systems are particularly well suited for this, since they can ensure consistent processes and practices as well as automate traceability and documentation generation

Strategy for Managing Requirements

- Requirements cannot be subsumed into Test artifacts & Source Code as is often done in “pure” (small scale) Agile implementations
- Use tooling to reduce the effort associated with iterative & incremental decomposition/allocation of requirements
 - Automation of trace management & documentation generation
- Even if “all” or most high level requirements are defined in advance, there is still benefit to prioritizing the analysis & allocation of requirements (less potential waste when changes arrive)
 - Employ Agile values, even if using traditional tools & practices
- Key is to have requirements manager(s) closely involved with rest of development team, rather than comprehensive requirement docs
 - Allows for feedback in both directions; reduces chances of misinterpretation

Strategy for Managing Requirements

- Traditional requirements management techniques can compliment Agile backlog management practices
 - Formal requirements management has better support for Impact analysis & dependency mgmt (e.g. Use Cases better at defining “size and shape” of the product)
 - Want to transform from sets of requirements to Stories [Cockburn]
 - Stories provide a queue of relatively independent functionality that can be prioritized and addressed using pull-based workflow [Cohn-Stories]



The Business Case for Application Lifecycle Management (ALM)



MKS

The Case for an ALM Solution That Supports Agile

- Cross-functional teams exist in Agile thus, enterprise adoption of Agile practices requires organizations to follow an Application Lifecycle Management (ALM) approach

According to Gartner:

“Application life cycle management (ALM) tools provide critical governance, collaboration, change management and workflow functions to ensure that teams work efficiently and that corporate standards are maintained”.¹

1: Gartner Research “Don’t Let Short-Term Agile Create Long-Term Pain”, Thomas E. Murphy, Jim Duggan, David Norton, 7 April 2009

The Case for an ALM Solution That Supports Agile

According to Forrester:

“Planning within Agile projects happens on at least three levels:

- 1) product- or release-level planning;*
- 2) sprint or iteration planning; and*
- 3) individual planning.*

Planning also happens more frequently in Agile projects than in traditional ones, but differently...Frequent multi-level planning either pushes planning entirely out of traditional project management tools and into ALM tooling or else requires tight integration between project management and ALM tools.”²

2. Forrester Research “The Forrester Wave™: Agile Development Management Tools, Q2 2010,” Dave West and Jeffrey S. Hammond, 5 May 2010

ALM Helps Overcome Enterprise Challenges

Large distributed teams

- Agile and traditional support, remote access

Application inter-dependencies

- Related assets and support for dependencies

Compliance and governance

- Automated change tracking and fully automated audit trails
- Real-time dashboards and reports

Phased transformation

- Full Agile, full traditional, rapid tailoring, comprehensive dashboards

PMO – monitoring and WBS

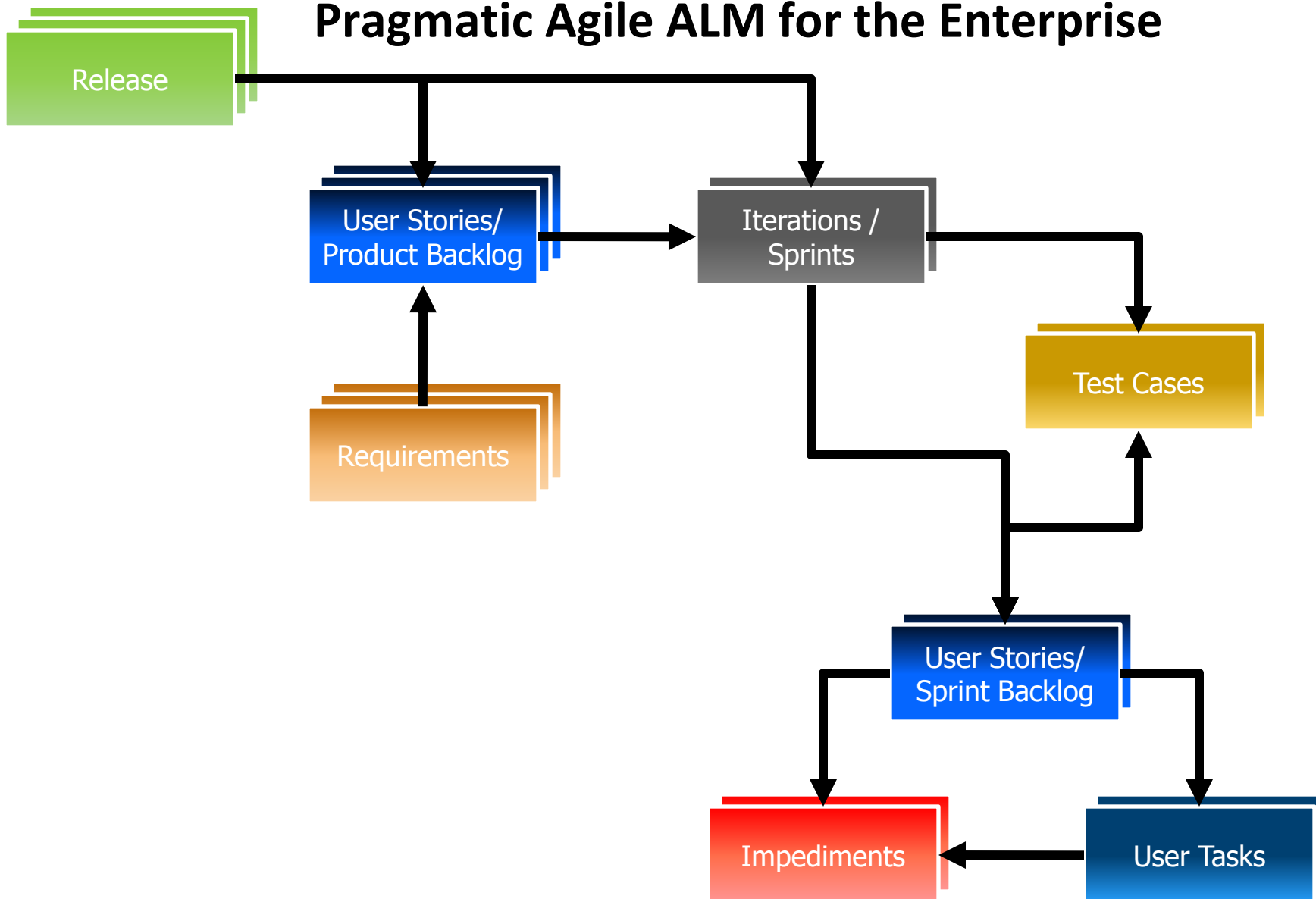
- Single source of truth with real-time data

Multiple product lines and variants

- Advanced reuse of all assets and relationships



Pragmatic Agile ALM for the Enterprise



Agile at MKS



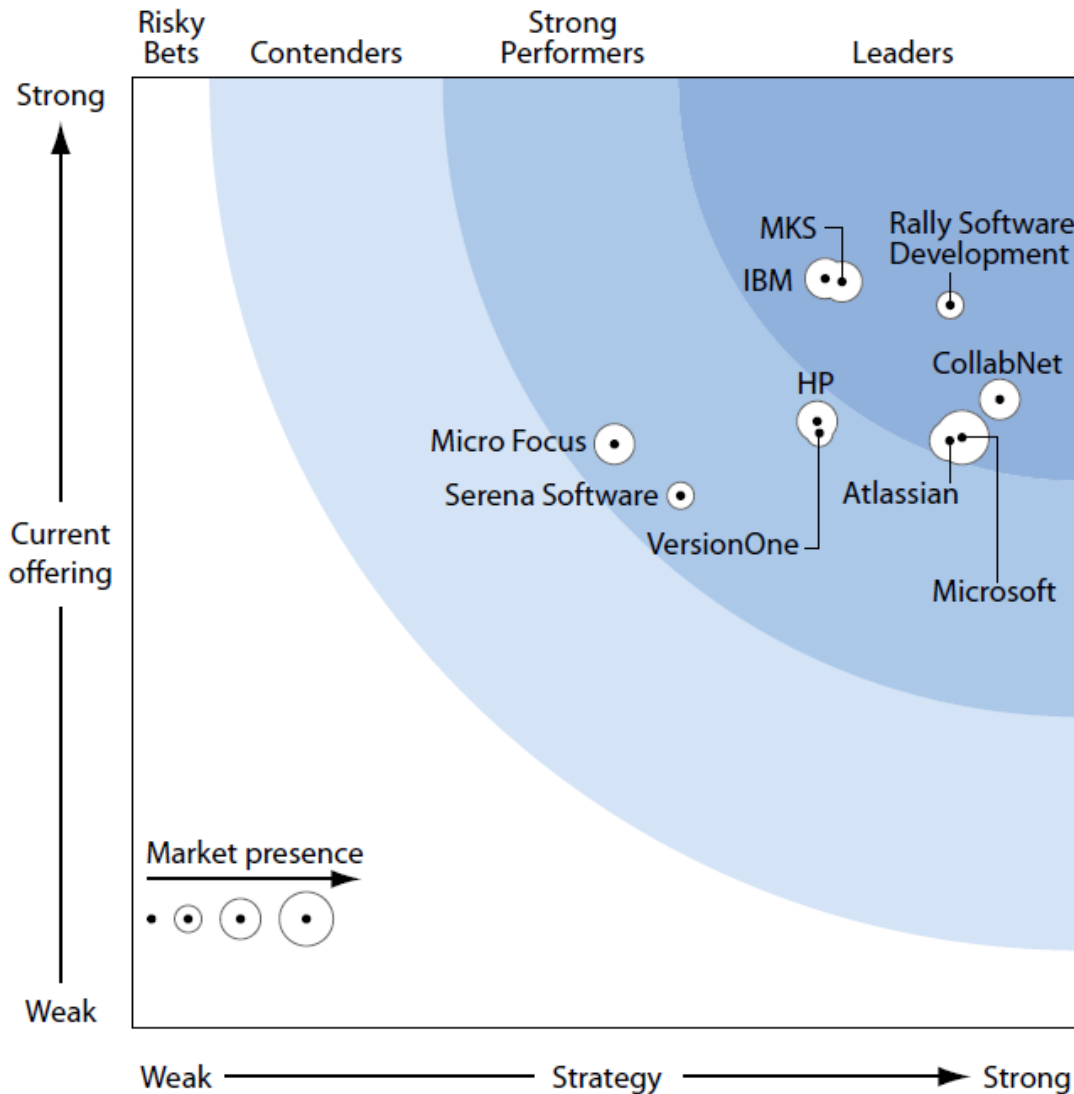
MKS

Agile at MKS

- Adopted in Fall 2010
- Goal was to deliver value to customers more quickly
- MKS Integrity 9.6 first release
 - Notable quality improvements
- MKS Integrity 9.7 in process right now
 - Change internal process to be much more Scrum baseds
- Conclusion and War stories discussion



Forrester Wave™: Agile Development Management Tools, Q2 2010



MKS

- Best overall current features
- Highest Scores in:
 - 'Security'
 - 'Running a Project' and
 - 'Process Configuration'
- *The MKS offering is "...very attractive for industries where compliance and audit are high priorities for developers."*

The Forrester Wave is copyrighted by Forrester Research, Inc. Forrester and Forrester Wave are trademarks of Forrester Research, Inc. The Forrester Wave is a graphical representation of Forrester's call on a market and is plotted using a detailed spreadsheet with exposed scores, weightings, and comments. Forrester does not endorse any vendor, product, or service depicted in the Forrester Wave. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change.

In Summary

- Adopting Agile in the enterprise is challenging – distributed teams, application inter-dependencies, compliance/governance, product lines & variants and more contribute to the complexity
- Enterprise Agile development needs to involve management of all activities & artifacts in a holistic manner – therefore, ALM is essential to scaling Agile
- MKS is a proven leader in ALM
 - Requirements management that supports adapting to change
 - Test management that facilitates test-driven development and enables reuse
 - SCCM that enables collaboration across the team and between Scrum teams
 - Unified data model enables collaboration, traceability and transparency across the entire lifecycle
 - Flexible process support simplifies transition from traditional to Agile processes
 - Visibility through metrics and reporting (burn down charts, burn up charts)



More Information

MKS: <http://www.mks.com/challenges/agile-transformation>

- U.S. & Canada: 1 800 613 7535, sales@mks.com
- U.K. & Northern Europe: 44 (0) 1483 733900, uksales@mks.com
- Central & Southern Europe: 49 (0) 711 351775 0
- Japan: +03 5789 5544
- Rest of World: (1) 519 884 2251

Get Your Copy of the Independent Report:

“Forrester Wave™: Agile Development Management Tools, Q2 2010”

http://www.mks.com/forrester_wave_agile_development_tools_assessment



Q & A



MKS

References

- [Boehm & Turner] “Balancing Agility and Discipline, A Guide for the Perplexed”, Barry Boehm & Richard Turner, 2004, Addison-Wesley, ISBN 0-321-18612-5
- [Poppendieck-Safety] “XP in a Safety Critical Environment”, essay originally published in the Sept. 2002 edition of the Cutter IT Journal, by Mary Poppendieck & Ron Morsicato; <http://www.poppendieck.com/safety.htm>
- [Cockburn] “Elephant Carpaccio”, Alastair Cockburn, 2008, on his blog; <http://alistair.cockburn.us/Elephant+carpaccio>
- [Cohn-Stories] “User Stories Applied: For Agile Software Development”, Mike Cohn, 2004, Addison-Wesley, ISBN 0-321-205685
- [SEI-CMMI] “CMMI or Agile: Why Not Embrace Both!”, Hillel Glazer, J. Dalton, D. Anderson, M. Konrad & S. Shrum, SEI Technical Note CMU/SEI-2008-TN-003; <http://www.sei.cmu.edu/reports/08tn003.pdf>
- [Karlesky] “Mocking the Embedded World: Test-Driven Development, Continuous Integration, and Design Patterns”, Michael Karlesky, G. Williams, W. Bereza & M. Fletcher, Embedded Systems Conference, San Jose, California, April 2007, ESC 413; http://www.atomicobject.com/files/ESC-413Paper_KarleskyWilliams.pdf

References

- [Forrester] “Agile Development: Mainstream Adoption Has Changed Agility”, Dave West & Tom Grant, Ph.D., Forrester Research note published January 20, 2010
- [Douglass] “The basics of being Agile in a real-time embedded systems environment: Part 2 – Benefits of Agile Methods”, Bruce Douglass, excerpt from “Real Time Agility”, Addison-Wesley, published on Embedded.com at: <http://www.embedded.com/design/opensource/224200540>
- [Wildeman] “How to Tackle Today’s Quality and Compliance Challenges in Pursuit of Developing Flawless Embedded Software”, Roy C. Wildeman, Forrester Research presentation, March 18, 2010
- [Doyle] “Agile in the Enterprise – Making Enterprise Agile Work with Intelligent ALM” <http://www.gartner.com/technology/media-products/newsletters/mks/issue1/article1.html>
- [Acks] “Agile Programming; Building Application Your Users are Sure to Like”, MC Press Online, January 24, 2011
<http://www.mcpressonline.com/programming/general/agile-programming-building-applications-your-users-are-sure-to-like.html>

Appendix

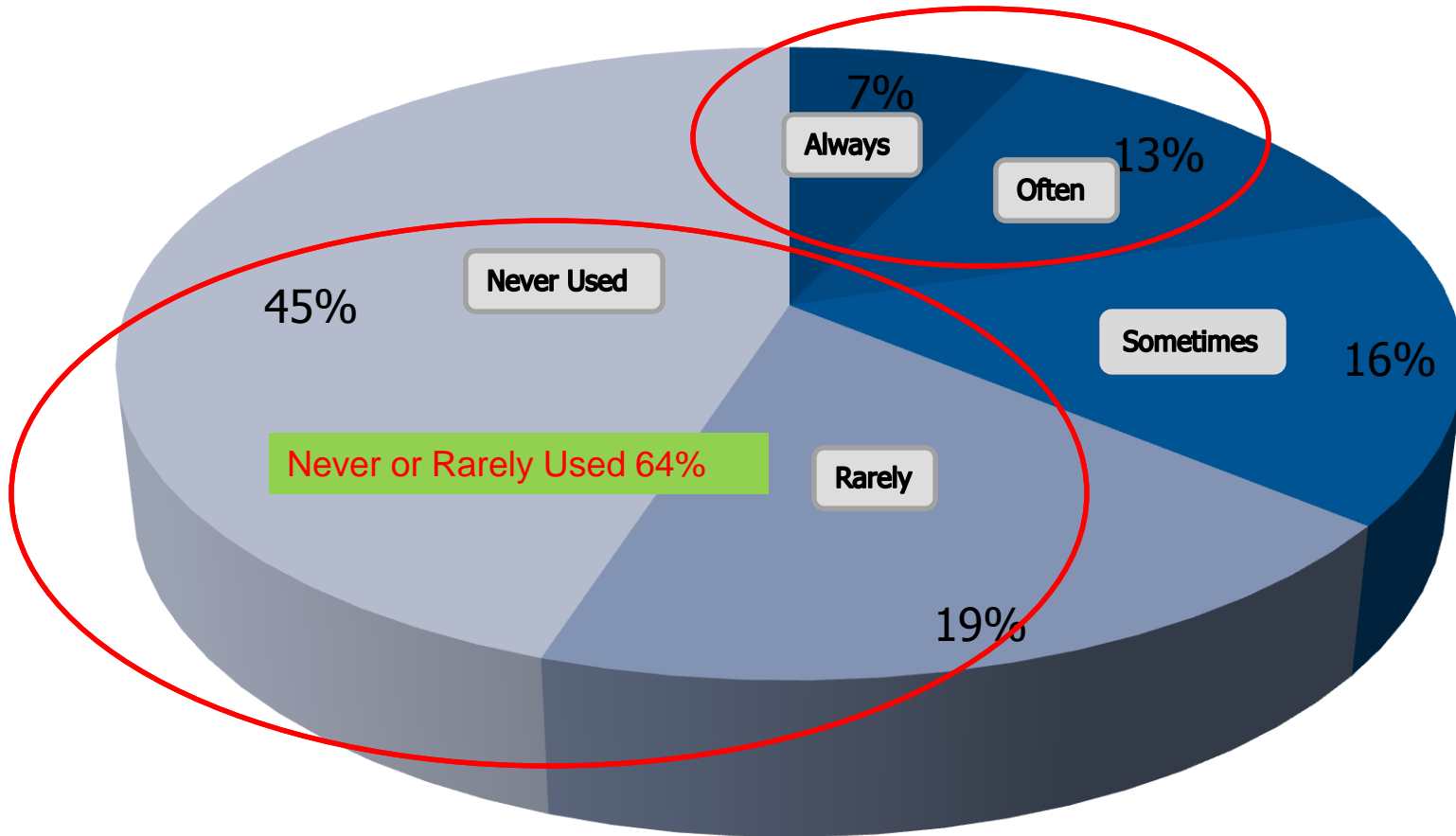


MKS

Features and Functions used in a typical system

■ Always ■ Often ■ Sometimes ■ Rarely ■ Never

Always or Often Used 20%

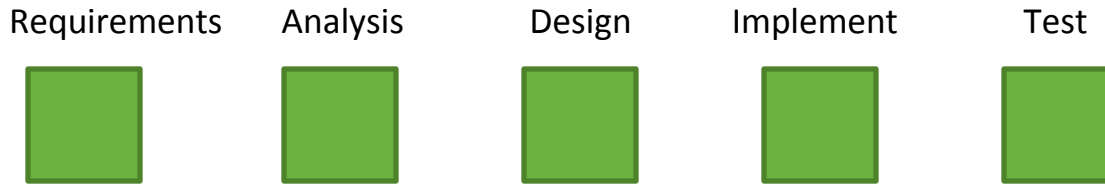


Source: Standish Group Study Reported by Jim Johnson, Chairman

Project Management Styles

- Waterfall - Isolated cycles of work

- IT Success Rate 1994 - 15%



- Iterative - Overlapping work

- IT Success Rate 2004 - 34% (<http://www.softwaremag.com/L.cfm?Doc=newsletter/2004-01-15/Standish>)



- SCRUM - All at once

- Agile Success Rate 2008 - 75% (http://www.versionone.com/pdf/3rdAnnualStateofAgile_FullDataReport.pdf)



- This reminded Takeuchi and Nonaka of a rugby scrum

MKS support for Agile/Lean ALM

Scrum Master ViewSet - MKS Integrity 2010

File View Tools Go Actions Help

Open ViewSet New View Relationships View Burndown Create Release Create Sprint Agile Dashboard Expand: - Select - select query to run

Query: Product Backlog Query: Active Releases Query: All Sprints Query: All Teams Sprint: 956 Release: 863


Show items containing where

Structure	State	Summary
863 Release	In Progress	Weather Version 1
864 Sprint	Completed	Make sample data go from server to device
870 Sprint	Completed	Minimal working version
877 Sprint	Completed	Plug in the real weather data
860 Team	Active	Red Team
878 User Story	Accepted	Fetch one day temperature data from the weather provide system
883 User Story	Accepted	Fetch rain, snow, etc details from the provider
888 User Story	Accepted	Fetch several days from the provider
891 User Story	Accepted	Auto-refresh feature
922 Impediment	Closed	Server went down for 24 hours
892 Task	Implemented	Make the client ping server once per 4 hours
893 Task	Implemented	Make the server update the client
905 User Story	Accepted	The background should be light blue
923 User Story	Accepted	New User Story in the backlog
925 User Story	Accepted	New user story in back
994 User Story	Accepted	Sample sub story under an epic
943 Test Suite	Open	Real-time Update for Weather App
1045 Requirement	Active	
997 Task	Implemented	Task related to User Story for demo
998 Task	Draft	Another task under user story
956 Sprint	In Progress	Maintainence Sprint
996 Sprint	Scheduled	Final Sprint
876 User Story	Defined	Provide ability to change location / city
875 User Story	Defined	Implement support for rain, snow etc. icons
957 User Story	Defined	Weather application must be in both English and French
1089 User Story	Defined	New User Story in the Release backlog
1093 User Story	Draft	User Story related to a requirement

User Story: 905

Created by [admin](#) on 13-May-2010 2:11:35 PM

Modified by [admin](#) on 5-Jul-2010 4:31:29 PM

User Story 

Overview Notes Tasks / Sub Stories Relationships Attachments Workflow History

Summary The background should be light blue

State: Accepted

Story Size: Almost Free [0.5]

Product: Weather - (859)

Release Project: /Agile/Weather Version 2

Story Owner: admin

Relative Priority: Nice to Have

Story Category: User Story

Rank:

Total # Related Tasks 2

Time spent on Story 8.0 hrs

Remaining Effort for Story 0.0 Ideal Hours

CPs in User Story 1

Files Modified for Story 2

Details

Description

The background should be light blue and not blue

(c) 2010

29 Rows, selected 1

Structure is a system field and is not editable on User Story: 905

admin@Sabikhi-7-64:7001

How Agile Improves Quality

- Continuous Integration rather than “big bang” integration
 - Early & ongoing mitigation of integration risks; regular regression testing
 - Allows for experimentation, due to short feedback cycles
- Test Driven Development ensures all functionality is tested, and that development is done in the context of customer requirements
 - Designing for change is as important as planning for change
 - Defer irreversible decisions until the last responsible moment
 - TDD, if done properly, results in executable specifications [Karlesky]
- Repeated, extensive peer review through cohesive, collaborative behavior of team (everyone working to a common big picture view)
 - Pair Programming can be thought of as continuous peer review

How Agile Improves Visibility

- Agile does not equate precision (a fully detailed plan) with accuracy; iterative & incremental updates to plan reduce chances of it becoming out of date or irrelevant
 - Responding to change rather than trying to minimize plan updates
- Status reporting focuses on completed functionality, with a clear definition of “done”, and frequent, regular updates
- Risk is addressed up front (continuous integration, design for change) rather than deferred until the end of the project
 - Greater chance that “done” actually means “done”
 - Defects resolved right away or during the next Sprint

Agile Project Management Benefits

- Just In Time Planning (Product & Sprint Backlogs) avoids waste by only elaborating as needed, for shorter planning windows
 - Less rework of planning artifacts, more confidence in detailed plans
 - Incrementally developing the highest priority requirements
- Time-boxed iterations provide plenty of opportunities for course correction and responding to change
 - Regular feedback on development progress and potential risk
- Better visibility (less deferred risk, better measures of status)
- Some organizations are incorporating Agile practices into their PMOs
 - The PMI has sponsored an Agile Community of Practice

An Agile Lifecycle Approach

