INCOSE Agile Health Care Systems Conference

May 23, 2016

Synchronizing Sub-teams When Mixing Agile With Other Methodologies

Rainy Mumper — Sr. Software Engineer
Clark Fortney — Sr. Software Engineer
Blake Ryan Alberts — Systems Engineer

Battelle
The Business of Innovation

Meet Your Presenters

Rainy Mumper
Senior Software Engineer
Battelle Consumer, Industrial and Medical

Rainy Mumper has almost 20 years of software development experience in embedded and application software for medical products. Ms. Mumper's expertise spans the development life-cycle from leading agile and waterfall based development teams to conducting requirements analysis, leading safety risk assessments, developing software architectures, managing requirements, developing verification strategies and overseeing verification. As a software task lead she has experience in software development planning and establishing Agile development processes that are appropriate for regulated devices. Ms. Mumper also has a working knowledge of software regulatory requirements for domestic and international medical devices.

Meet Your Presenters

Clark Fortney
Senior Software Engineer
Battelle Consumer, Industrial and Medical

For over 20 years, Mr. Fortney has focused his career on software development for medical devices. He has a broad background in Electrical Engineering, with an emphasis on embedded system/software design. He has served in software leadership roles for a wide variety of medical devices, including several drug delivery devices, enteral feeding pumps, and a heater/humidifier. Mr. Fortney's primary skills are in Software Engineering (especially hardware/software integration), System Engineering, and Task Management.

Meet Your Presenters

Blake Ryan Alberts
System Engineer
Battelle Consumer, Industrial and Medical

Mr. Alberts has a technical background in systems engineering in highly controlled industry environments such as military application ready subsystems and medical device development. Having worked at several companies which perform all aspects of engineering design and development, Mr. Alberts is dedicated to serving every stage of the life cycle of complex interdisciplinary engineering projects from concept generation to validation and manufacturing. His specialization is in electrical and software design, development, and integration.

Presentation Overview

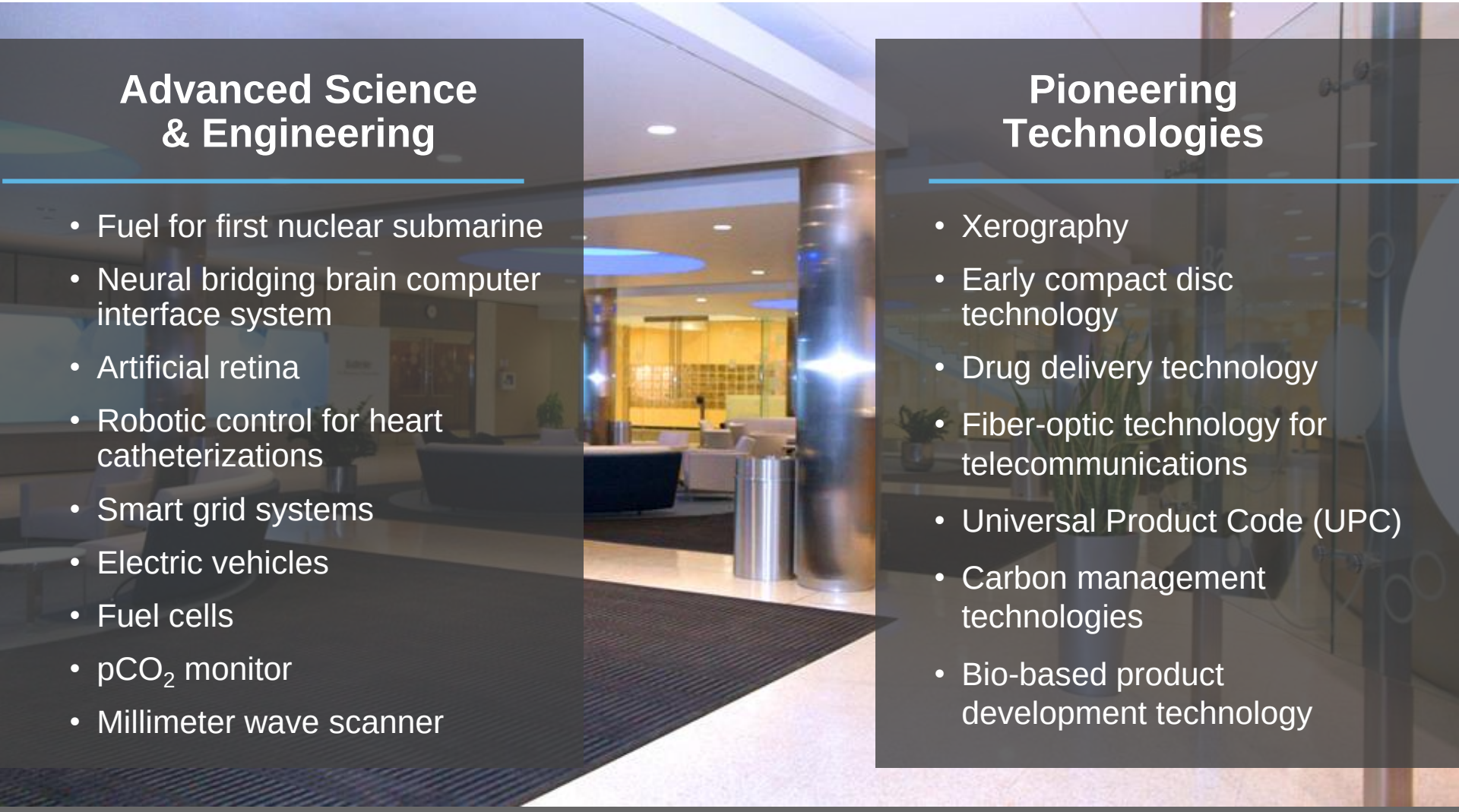
- Introduction to Battelle Medical Devices
- Mixing Agile With Other Methodologies
- Mixed Model Scenarios
- Strategies for Mixed Model Interactions
- Case Studies
- Wrap Up

BATTELLE MEDICAL DEVICES

Battelle

- $4 billion/year global R&D enterprise located in Columbus, OH
- Non-profit, charitable trust formed in 1929 by the Will of Gordon Battelle to provide contract R&D for the betterment of the world
- More than 20,000 employees in over 60 world-wide locations
- Serving customers across Consumer, Industrial & Medical; Energy & Environment; National Security and Laboratory Management
- Net income invested in science & technology, education and charity

Over 80 Years of Innovation



Advanced Science & Engineering

- Fuel for first nuclear submarine
- Neural bridging brain computer interface system
- Artificial retina
- Robotic control for heart catheterizations
- Smart grid systems
- Electric vehicles
- Fuel cells
- pCO_2 monitor
- Millimeter wave scanner

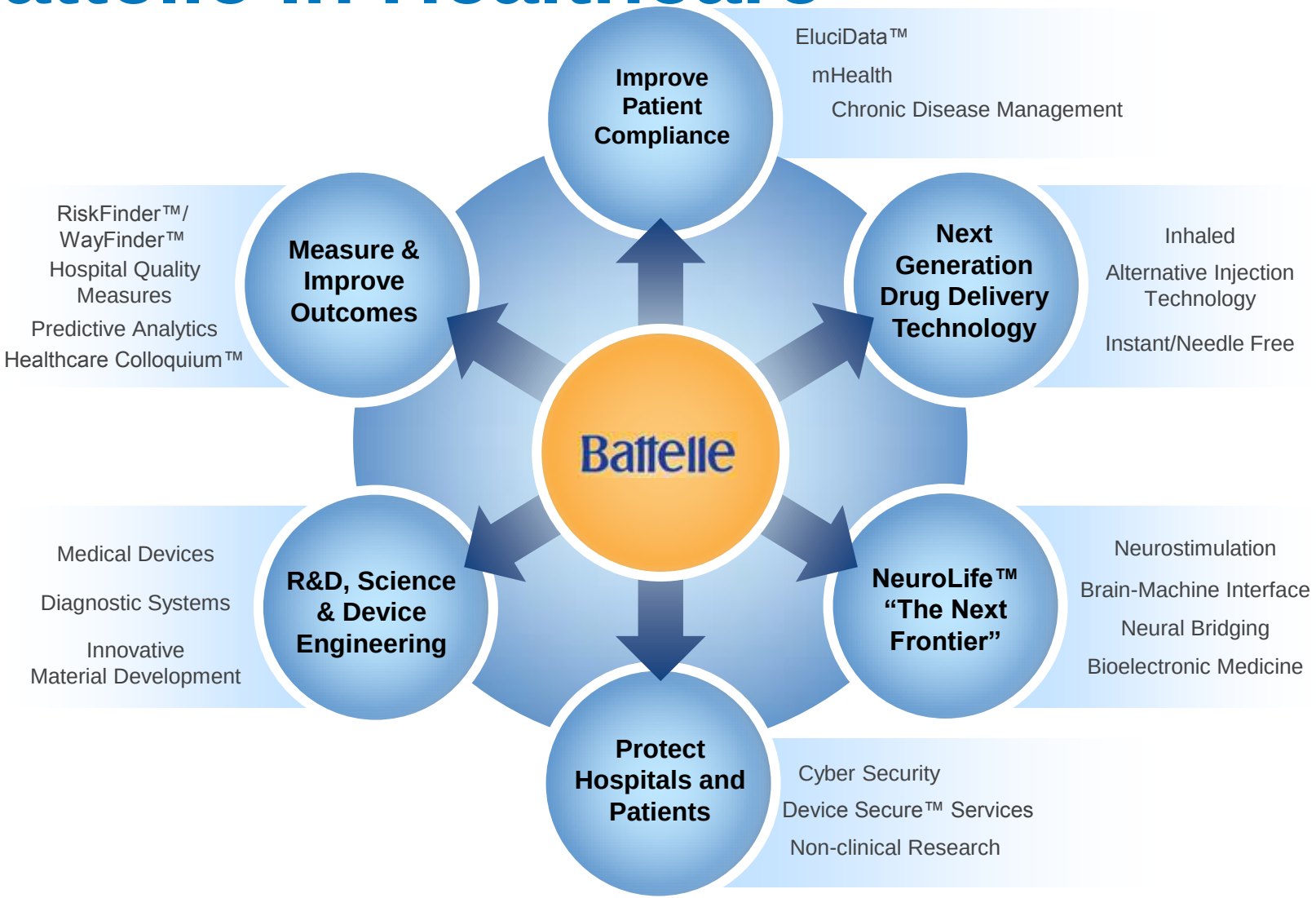
Pioneering Technologies

- Xerography
- Early compact disc technology
- Drug delivery technology
- Fiber-optic technology for telecommunications
- Universal Product Code (UPC)
- Carbon management technologies
- Bio-based product development technology

Multidisciplinary Ecosystem

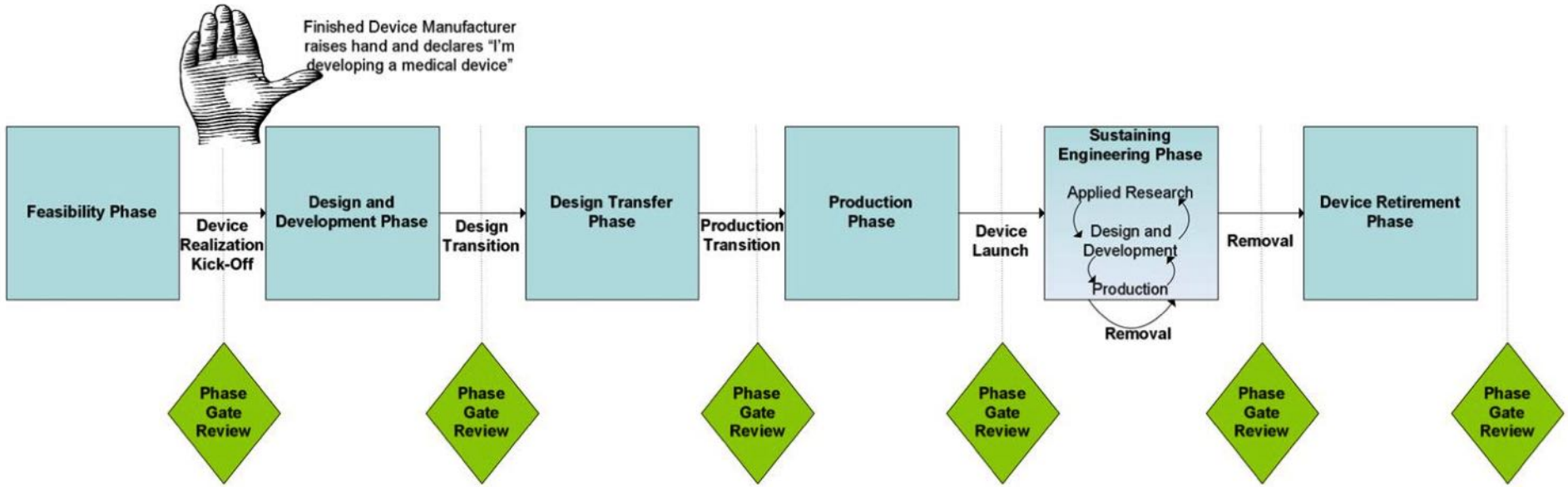
9

Battelle In Healthcare



Medical Device Development Process

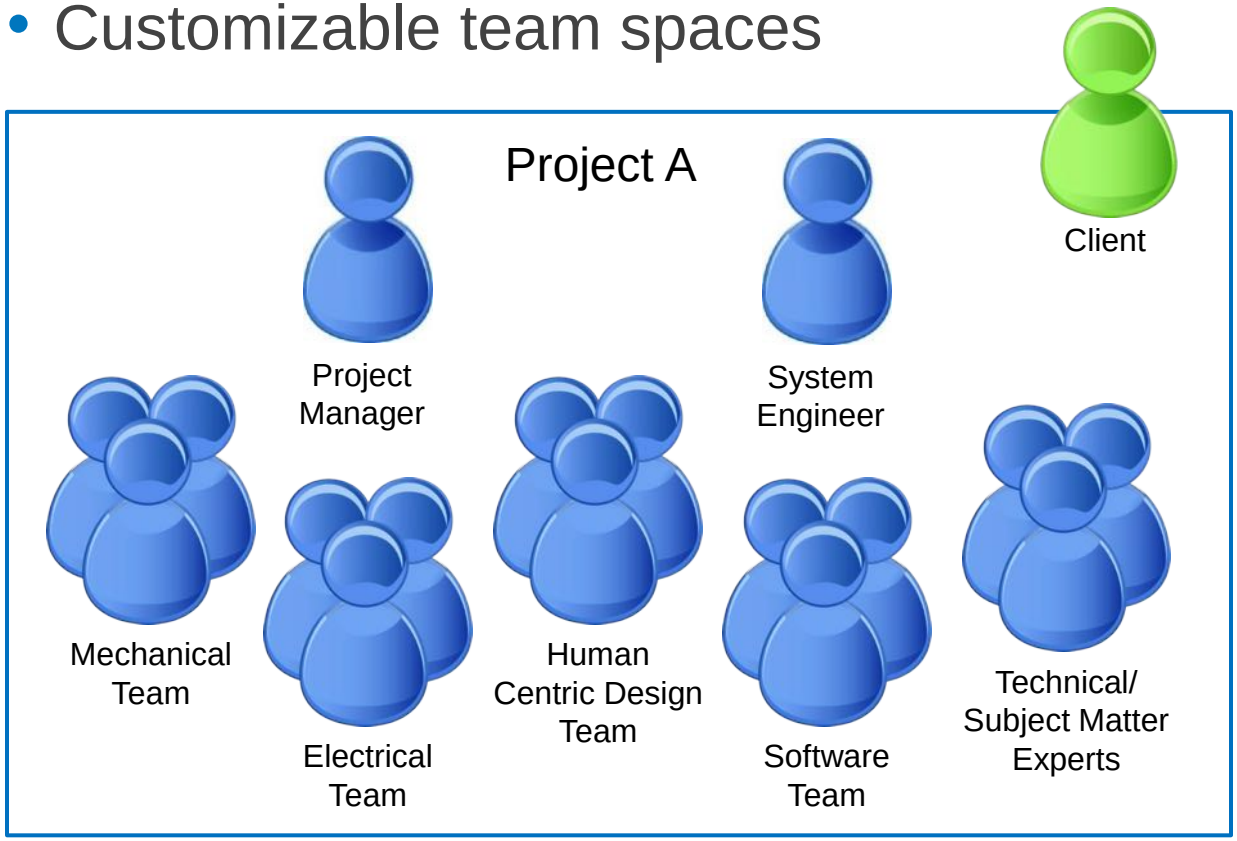
- Customizable Process to support a diverse client base
- Projects can enter and exit the process at any point

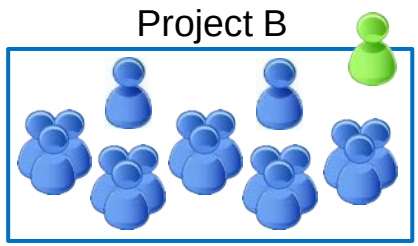


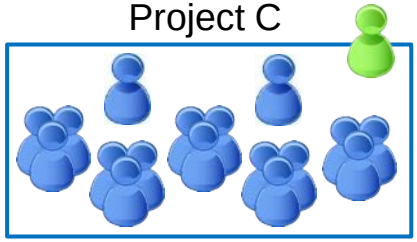
Reference - "Medical Device Development Process," Jones D., Masters M.; INCOSE International Symposium, June 15–19, 2008, Volume 18, Issue 1

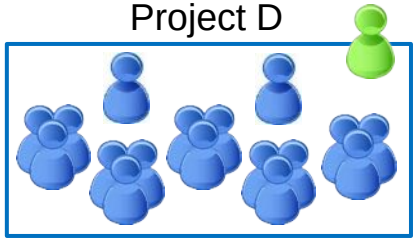
Medical Device Development Teams

- Co-located multidisciplinary, cross-industry project teams
- Customizable team spaces









Evolution of Agile Use in Battelle Medical Devices

- Agile lifecycle first used at Battelle for a Medical Device starting around 2000
- Agile use started in Software Teams but gaining wider adoption with Electrical and Systems
- Project teams using mixed and/or hybrid methodologies inside an overall Stage Gate Process

Graphic Source - “Value-Based Agile Adoption," Sumant Raja, August 1, 2013
https://www.scrumalliance.org/community/articles/2013/august/value-based-agile-adoption

MIXING AGILE WITH OTHER METHODOLOGIES

Why Mix Agile with Other Methodologies?

- Some groups within a company, especially those outside of software, may be unfamiliar with Agile practices.
- Mixing allows for gradual adoption of Agile practices across the various disciplines of an organization.
- Provides flexibility to match best methodology for project/problem subset with available sub-team skillset.

Challenges with Mixed Methodologies

- Differences in development schedules
- Effective communications between sub-teams requires understanding each other's terminology
- Inadequate communication and coordination between sub-teams can lead to project inefficiencies.

Survey

(1) Do you use Stage Gates as part of an overall process?

- ☐ Yes
- ☐ No

(2) Are you using a mix of lifecycle methods for teams that interface?

- ☐ Yes
- ☐ No

(3) Are team interactions driven by a structured process or ad-hoc?

- ☐ Structured
- ☐ Ad-hoc

MIXED MODEL SCENARIOS

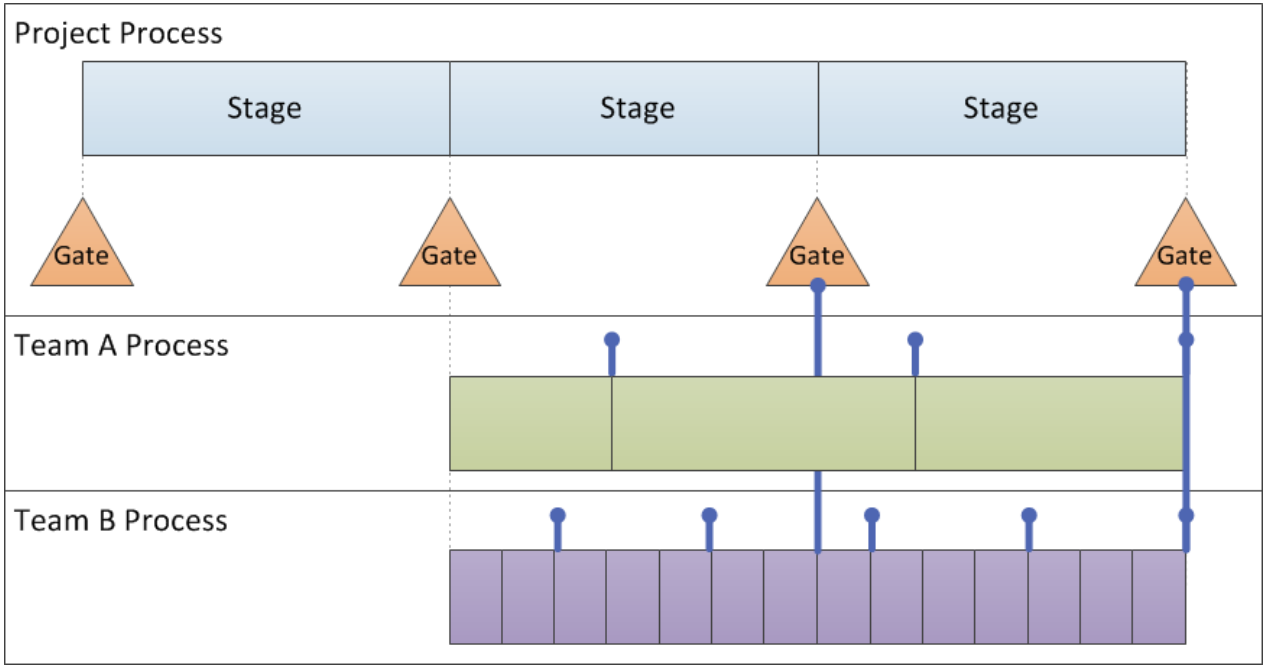
Mixed Model Scenarios

- Possible mixed environment scenarios
  - Independent Teams: Completely independent of each other and just need to be deployed together (functionality does not require integration)
  - Low Dependency: Some dependencies exist but success doesn't require constant synchronization
  - High Dependency: Substantial dependencies exist throughout development, therefore a high level of interaction and synchronization is required in order for each team to succeed
- Each scenario requires a different level of synchronization
- Establish a common governance structure for mixed method teams on the project

Reference - "Mixing Agile and Waterfall Development in the Scaled Agile Framework," Ken France, December 29, 2012
http://www.scaledagileframework.com/mixing-agile-and-waterfall-development-in-the-scaled-agile-framework/

Synchronizing Independent Teams

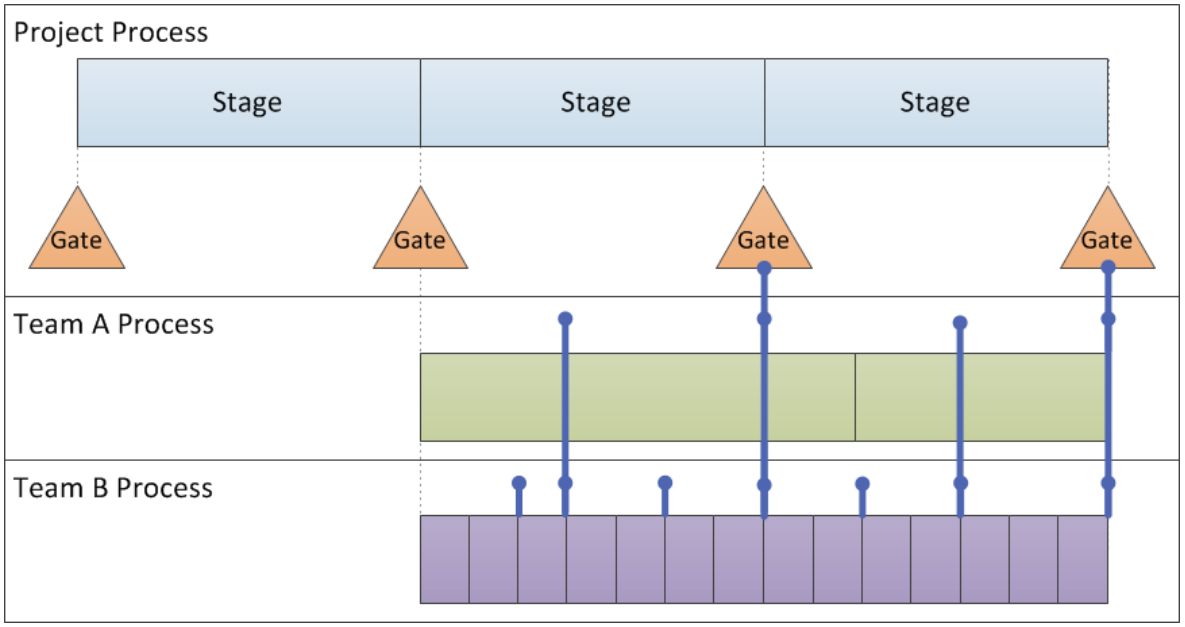
- Synchronize at Project Milestones
- Teams agree to which Stage Gates apply to them and what must be provided



Reference - “Mixing Agile and Waterfall Development in the Scaled Agile Framework," Ken France, December 29, 2012
http://www.scaledagileframework.com/mixing-agile-and-waterfall-development-in-the-scaled-agile-framework/

Synchronizing Low Dependency Teams

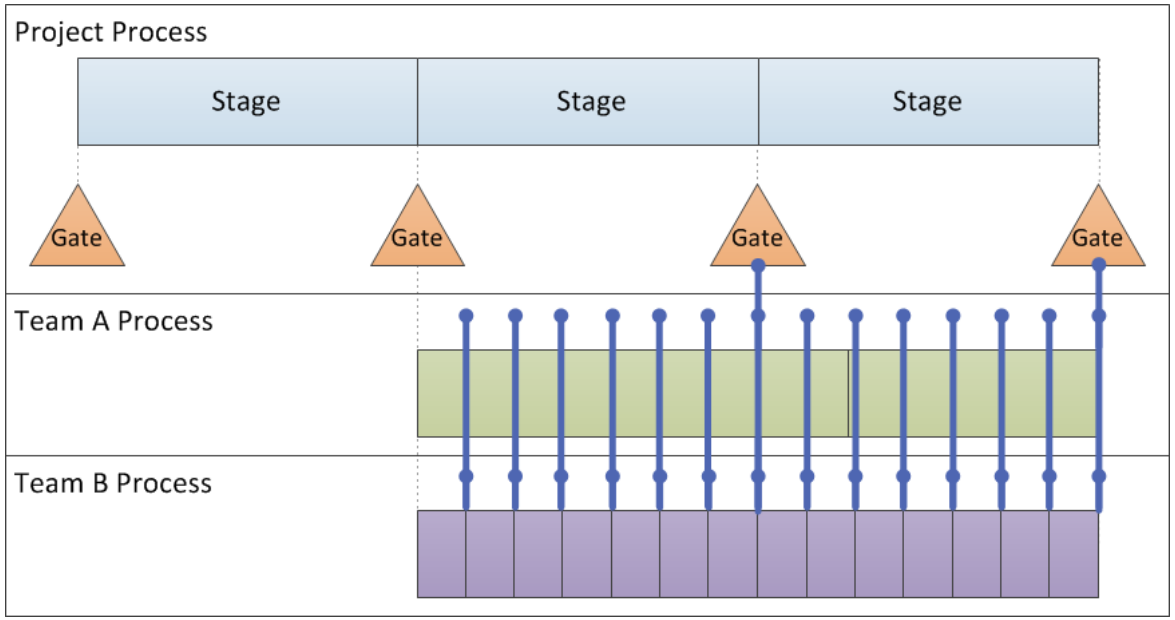
- Synchronize at (1) project milestones and (2) interfacing team milestones
- Waterfall teams might need to be more incremental
- Iterative teams need to identify which iteration boundaries align to interfacing team milestones.
- Boundaries may need to be moved/adjusted from normal cadence at integration points.



Reference - “Mixing Agile and Waterfall Development in the Scaled Agile Framework," Ken France, December 29, 2012
http://www.scaledagileframework.com/mixing-agile-and-waterfall-development-in-the-scaled-agile-framework/

Synchronizing High Dependency Teams

- Synchronize at (1) project milestones and (2) whichever frequency the dependency structure between the teams dictates
- Synchronization at the shortest cadence boundaries recommended. Every cycle on the team with the shortest cadence is a potential integration point.
- Frequent interaction at a technical level – attend each other's meetings and ceremonies (weekly team meetings, daily standups, planning and demos/retrospectives)



Survey

(1) For teams using mixed models, which scenarios apply to your team?
Check all that apply

- ☐ A. Independent Teams
- ☐ B. Low Dependency Teams
- ☐ C. High Dependency Teams

STRATEGIES FOR MIXED MODEL INTERACTIONS

Mixed Model Interaction Strategies

- Joint Project Planning
- Joint Requirements and Design Workshops
- Mockups and Designing to Interfaces
- Encapsulation and Refactoring
- Joint Iteration Planning
- Iteration Coordination
- Frequent Integration
- Joint Demos/Retrospectives
- Tips for Managing Releases across Sub-Teams
- Embedding Special Roles

Joint Project Planning

- Define the overall project milestones together
- Define key team milestones and dependencies between teams
- This allows the sub-teams to define their various dependency scenarios and structure their team interactions accordingly

Joint Requirements and Design Workshops

- Approach initial requirements and design as a collaboration
- Strive for “enough” upfront requirements and design
- Derive concrete system behaviors. Imagine demonstrating the feature and describe what you see. Eliminate ambiguities whenever possible.
- Use various types of diagrams to describe the system (context diagram, block diagrams, activity diagrams)
- Use whiteboards to refine as you go (and take pictures)
- Identify dependencies and plan for integration points and mechanisms

Reference - “Technical Strategies for Agile and Waterfall Interoperability at Scale," Alex Yakyma
http://www.scaledagileframework.com/technical-strategies-for-agile-and-waterfall-interoperability-at-scale/

Mockups and Designing to Interfaces

- Iterations should be laid out with expected or anticipated needs for longer timeframe elements
- Considering mocking (stub, model, special test fixture, etc.) an interface until the real thing is available. Plan mocks as part of the requirements and design workshops.
  - Have interfacing teams define the simulated interfaces
  - Decide which team is responsible for the mock
- Design to interfaces when using mocks to allow teams to replace each mockup with the real thing as soon as it is available.
- Capture inconsistencies that are found immediately

Reference - “Technical Strategies for Agile and Waterfall Interoperability at Scale," Alex Yakyma
http://www.scaledagileframework.com/technical-strategies-for-agile-and-waterfall-interoperability-at-scale/

Encapsulation and Refactoring

- Identify areas of variability and encapsulate those to the greatest extent possible so they can be changed in relative isolation
- Example – for Software, design patterns provide approaches more immune to change and lend themselves to better refactoring

Reference - “Technical Strategies for Agile and Waterfall Interoperability at Scale," Alex Yakyma
http://www.scaledagileframework.com/technical-strategies-for-agile-and-waterfall-interoperability-at-scale/

Joint Iteration Planning

- Include dependent team representatives
- Include individuals that serve as shared resources across teams due to specialized skill sets
- Identify an iteration theme and goals
- Discuss any cross-team dependencies (existing and emerging)
- Break down the work selected for the iteration
- When tasking out a cross-team dependent piece of work, collaboratively plan how both teams will address the work
- Determine which teams/individuals have the greatest need for interaction during that iteration. Plan interactions for the iteration. Will they need to attend each other's meetings/ceremonies, what is the frequency of the interaction?

Reference - "Integrating Agile into a Waterfall World," Joseph Flahiff, June 6, 2011
http://www.infoq.com/articles/agile-in-waterfall-world

Iteration Coordination

- Teams can encounter any number of issues that require cross-team coordination
- Team Leads should meet frequently and regularly to discuss the identified team dependencies and any new items coming out of key team meetings.
- A Project Manager or System Engineer should attend this meeting to listen to the cross-team dependencies and determine if they are needed in any way to facilitate the resolution of issues.

Reference - “Integrating Agile into a Waterfall World," Joseph Flahiff, June 6, 2011
http://www.infoq.com/articles/agile-in-waterfall-world

Frequent Integration

- Continuous Integration is ideal but frequent integration is essential when mixing Agile and non-Agile teams
- Early dependency planning should dictate integration sync points
- Integrations are joint activities (no over the fence integrations allowed)
- Dedicate time in each team's schedule for the integration (often in parallel with normal team tasks)
- Fix integration failures as they occur whenever possible. Prioritize over normal team tasks.

Reference - "Technical Strategies for Agile and Waterfall Interoperability at Scale," Alex Yakyma
http://www.scaledagileframework.com/technical-strategies-for-agile-and-waterfall-interoperability-at-scale/

Joint Demos/Retrospectives

- Involve all teams in iteration demos especially teams with dependencies
- Demonstrate the workflow that the customer would follow if they were using the systems in production
- Let the logical progression of the workflow drive each team's part in the demo

Reference - "Integrating Agile into a Waterfall World," Joseph Flahiff, June 6, 2011
http://www.infoq.com/articles/agile-in-waterfall-world

Tips for Managing Releases Across Sub-teams

- The Project Manager and Team Leads should work together to plan for the release
- Define cross-team dependencies for an upcoming release and identify impacts to each team's normal work flow
- Team Leads manage the release process to minimize impact to working teams
- Start release planning/tasking on an iteration boundary to keep the team focused on current iteration execution
- Assign the bulk of release related responsibility to a individual team member to resolve defects quickly.

Reference - "Integrating Agile into a Waterfall World," Joseph Flahiff, June 6, 2011
http://www.infoq.com/articles/agile-in-waterfall-world

Embedding Special Roles

- Embed specialists short-term or part-time
- Example specialist roles
  - Lead System Engineer/Architect
  - Subject Matter/Technical Expert
  - Human Factors/User Experience teams
  - Manufacturing
  - Testers and QA

Reference - Ambler, Scott; Lines, Mark (2012).
Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise.

Discussion

For teams using mixed models, what are some techniques you've had success with for synchronizing sub-teams?

CASE STUDY:
DRUG DELIVERY DEVICE

Team Methodologies

- Systems – Waterfall Model with feedback and planned increments
  - Upfront capture of User and Design Inputs. Includes planning the project as a whole, risk analysis, requirements, and high level architecture.
  - Prototype/Benchtop Increment – A cycle of design, implementation, integration of select components and select testing based on technical risks
  - Alpha Increment – A cycle of design, implementation, integration for most system capabilities. Developing and characterizing verification protocols.
  - Verification Increment – A cycle of refinements with some redesign of components/subsystems as needed. Final execution of verification protocols.
  - *Each increment includes updates to risk analysis, requirements, and high level architecture*
- EE/ME/HCD – Iterative and Incremental
- Software – Agile in a waterfall

Team Synchronization

- Sub-Teams synchronized at the system level and based on dependencies of other teams
- Synchronization Methods
  - Joint Project Planning
  - Joint Architecture Workshops
  - Mockups and Designing to Interfaces
  - Encapsulation and Refactoring
  - Joint Demos/Retrospectives
  - Iteration Coordination

Synchronization Highlights – Alpha Unit Build Coordination

- Alpha Unit Builds coordinated between SE, ME, SW, EE, HCD, PM and Manufacturing
- Weekly touch base to identify dependencies prior to build
- Build Coordination
  - Dedicated build room
  - Daily standups with all sub-team representatives
  - Whiteboard status review of build units
  - Coordination of design redlines
- Team Demonstration of the first functional integrated unit
- Team Retrospective to finalize changes, lessons learned

Synchronization Highlights – GUI Coordination

- GUI Coordination
  - Coordination between HCD and Software
  - Iteration Planning with HCD for Delivery of Screens/Behavior Summaries
  - Screens provided weekly with reviews as needed during Sprint
  - Alternative Path coordination
    - Weekly meeting to discuss 2-3 alternative paths
    - HCD team develops workflows and screens
    - Review with Software and HCD prior to implementation, select next set to discuss

CASE STUDY: CLOUD BASED MOBILE HEALTH APPLICATION

Team Methodologies

- Hybrid Waterfall-Agile approach used for Systems, Software and UX Teams
- User Needs, Marketing Requirements, and Hazards Analysis captured up front
- Interdisciplinary Agile Sprints

Team Synchronization

- Synchronization through Agile ceremonies and temporal staging of lifecycle activities
- Synchronization Methods
  - Joint Project Planning
  - Joint Iteration Planning
  - Iteration Coordination
  - Joint Demos/Retrospectives
  - Embedding Special Roles

Synchronization Highlights – Temporal Staging

	Sprint 5	Sprint 6	Sprint 7	Sprint 8
Systems	**Sprint 6 Requirements**	Sprint 7 Requirements	Sprint 8 Requirements	Sprint 9 Requirements
	Sprint 5 Ver. Procedures	**Sprint 6 Ver. Procedures**	Sprint 7 Ver. Procedures	Sprint 8 Ver. Procedures
	Sprint 4 Verification	Sprint 5 Verification	**Sprint 6 Verification**	Sprint 7 Verification
UX	**Sprint 6 Wireframes**	Sprint 7 Wireframes	Sprint 8 Wireframes	Sprint 9 Wireframes
	Sprint 4 User Testing	Sprint 5 User Testing	**Sprint 6 User Testing**	Sprint 7 User Testing
Software	Sprint 5 Code and Test	**Sprint 6 Code and Test**	Sprint 7 Code and Test	Sprint 8 Code and Test
	Issue Fixes from User/ Verification Testing	Issue Fixes from User/ Verification Testing	Issue Fixes from User/ Verification Testing	**Issue Fixes from User/ Verification Testing**

WRAP UP

Summary

- Subteams can use different development methodologies to allow organizations to phase in Agile or provide greater flexibility across disciplines or functions.
- Although subteams may follow different processes, sub-team interfaces need to be well defined
- Identify team dependencies early
- Structure sub-team interactions based on dependency levels
- Use strategies for mixed model interactions that best suit your teams

Questions

Thank You

For more information, please contact

Rainy Mumper
Senior Software Engineer
MumperR@battelle.org

Battelle
The Business of Innovation