Agile Software Development
Application in the Medical Device Industry

Kelly Weyrauch
Medtronic, Inc.
(29 April 2008)
Introduction

■ Purpose
  ■ Provide an introduction to Agile Software Development as it applies to the Medical Device Software world

■ Desired Result of the Meeting
  ■ Understand the basics of Agile
    ■ What it is
    ■ Why you should use it
    ■ How you can use it
  ■ Determine whether you want to learn more
Introduction

Agenda

What is Agile? Why is it a good fit for Medical Device Software development?

- Principles / Values
- Benefits
- Practices

Worries / Myths / Pitfalls / Strengths
What Is Agile?

Agile Methods provide a framework for collaborative teams to reliably deliver software using a highly iterative approach that provides frequent feedback for continuous improvement of the process and product.

Key concepts include:

- Constant inspection and adaptation
- Focus on customer value
- Data driven; schedule data, product value
- Increased focus & synchronization
- Emphasis on collaboration, teamwork and learning
- Analogous to Lean Manufacturing
- Change tolerant; more business opportunities
- Open, highly visible, inclusive
Agile History

- Many variants (Extreme Programming (XP), Scrum, Crystal, DSDM, Lean SW, …)
- Values summarized by the Agile Manifesto
- Roots in the IT and/or Consultant world
- Dramatic rise in popularity
  - Publications, conference attendance, maturity of applications, breadth of businesses
- “Crossed the Chasm”
What Is Agile?

- Agile Framework
  - Values/Principles
    - Why is it the way it is, why do we do what we do
    - Things we believe to be true about our world
  - Benefits
    - Why is this a good thing
    - What am I trying to be good at / better at
  - Practices
    - How to do it
- Techniques for cooking, not just a bunch of recipes
Agile Values

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.


29 April 2008
Agile Values

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

*Individuals and interactions* over processes and tools
*Working software* over comprehensive documentation
*Customer collaboration* over contract negotiation
*Responding to change* over following a plan

That is, while there is value in the items on the right, we value the items on the left more.


29 April 2008
Agile Values

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

**Individuals and interactions** over processes and tools

**Working software** over comprehensive documentation

**Customer collaboration** over contract negotiation

**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.


29 April 2008
Agile Values

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.


29 April 2008
Agile Values

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.


29 April 2008
Agile Values/Principles:

General

- Simplicity
- Deliver Customer Value
- Embrace Change
- Inspect and Adapt
- Eliminate Waste
- Maximize Transparency
- Iterate Incrementally
- Cultivate Collaboration
Agile Values/Principles: Specific to our world

- **Validation**
  - Tight connection to the customer
  - Regular validation points

- **Safety / Risk Management**
  - Embedded into every iteration, not separate

- **Verification**
  - Test Driven Design (aka “Test First”)

- **Process**
  - Process ownership - Engineers allowed to think
  - Value-added process – Time spent in the right place
Agile Benefits: Our List

- **Quality**
  - Find bugs sooner
  - Improve feedback
  - Faster response to change
  - Better & faster testing
  - Encourage small-team atmosphere
  - Increase visibility of project levers
  - Improve learning
  - Engage team
  - Empower team
  - Increase collaboration

- **Productivity**

- **Predictability**

What’s on YOUR list?
Agile Practices

- Team Structure
- Definition (Requirements, Testing)
- Planning
  - Rapid Iteration
- Verification
- Validation
- Simple Design
Agile Practices:
Team Structure

- Team Ownership (of the Product and the Process)
  - Shared ownership, guided by roles
  - Multiple verification points to address “independence”
  - Collaboration (Pair Programming, Co-Location, Collective Ownership (of code))
  - Team Reflections

- Roles
  - Project Manager Role
  - Coach Role
  - Customer Role
  - Tracker Role
  - Tester Role
  - Developer Role
Agile Practices: Definition (Requirements, Testing)

- Product Backlog
  - Features
  - Stories
  - Sub-Stories
  - Customer-facing

- “Tests as Requirement”
  - Acceptance Tests
  - Functional Tests
  - Design-level testing
Agile Practices: Planning

- “If you can’t plan well, then plan often”
- Layers of Planning
- As driven by:
  - Delivering customer value
  - Reducing risk
Agile Practices:
Planning - Rapid Iteration

- Releasable software
  - Able to ship at any time
  - Minimize/eliminate “Design Debt”

- Sprint Cycle
  - Short-sprints of delivering customer value

- Burndowns
  - Mechanism to plan, track, re-plan
Agile Practices: Verification

- Test-Driven Development: “We’re done when the tests say we’re done”
  - Determine how the SW will be verified
    - Unit tests, Module tests, Integration Tests, System Tests, Analysis
  - Write tests first, automate them, write code
- The Tests are the Requirements
- Collaboration (Pair Programming, Switching Pairs, Third Party Review)
- Continuous Integration
  - Build at least once a day
  - Run all tests constantly
  - Don’t break the build
Agile Framework: Practices – Validation

- Customer Role
  - On-Site Customer, tightly coupled to the team

- Acceptance Tests
  - “We’re done when the customer says we’re done”

- Demo
Agile Framework: Practices – Simple Design

- YAGNI
- Refactoring
- Patterns
- Shared Code Ownership
  - Coding Standards
Quality System

Regulations, Guidance Documents, External Standards

Design Control Policy
- Establish the Quality System
Scope: Entire Company

Global Process
- Process Architecture
- Development Phases
- Major Process Requirements
- Major Deliverables
Scope: Organization

Procedures (Protocols)
- Next Level detail of Process Requirements
- Project Deliverables
Scope: Product Line

Procedures (Work Instructions)
- Detailed Process Requirements
- Detailed Deliverables
Scope: Discipline (ex Software)

Plans
- Project-specific processes/procedures, or project-specific details of common Procedures
- Work plan: tasks that instantiate the procedures
Scope: Project

Training - Details of day-to-day execution
Scope: Project, Team, or Discipline

AGILE METHODS

Satisfies
Satisfies
Influenced By
Satisfies
Satisfies
Follows
Explain, clarifies, amplifies
Worries, Fears, Bad Rumors
Worries, Fears, Bad Rumors

“Agile says you don’t need documentation”

- Produce what you need
- Be sure you need it
- Make it relevant to those producing it, not just to those consuming it
Worries, Fears, Bad Rumors

- “Agile says you don’t need a plan”
- “Agile planning is: You’re done when you’re done”
  - Utter nonsense
  - If you can’t plan well, then plan often
    - Plans change – embrace that as a good thing
  - Increases predictability
  - Defer decision-making to “The Last Responsible Moment”
Worries, Fears, Bad Rumors

“Agile is undisciplined”

- Emphasizes the discipline of the team that owns its processes/practices
- De-emphasizes the discipline of an imposed process

“Simple, clear purpose and principles give rise to complex, intelligent behavior. Complex rules and regulations give rise to simple, stupid behavior.” - Dee Hock, VISA Founder
Worries, Fears, Bad Rumors

- How much will FDA get into the practices?
  - Depends on how deep they want to go.
  - Depends on Auditor Expectations.
    - “Suitability of XP/Agile may depend more on Culture and Expectations than Regulation.”
    - If Auditors expect to see artifacts that look like Waterfall, will require training to map those expectations into the Agile world.
Is Agile suitable to the safety-critical world?

- If you do it well, then yes. If you do it badly, then no.
- Some elements of Agile bring high value to the safety-critical world
- Works very well in the context of a robust Quality System
What’s Next

- Questions, comments, worries?
- What else do you want to know?

Desired Result of the Meeting

- Understand the basics of Agile
  - What it is
  - Why you should use it
  - How you can use it
- Determine whether you want to learn more
References

- www.agilealliance.com Home of the Agile Alliance
- www.controlchaos.com Home of Scrum
- www.scrumalliance.org Another Scrum site
- www.xprogramming.com Ron Jeffries Extreme Programming website
- www.extremeprogramming.org Another Extreme Programming website
- Agile Conference www.agile2008.org