Agile & Compliant Medical Product Development

Dr. Scott S. Elliott – TechZecs LLC
Aaron W. Joseph – Consensia, Inc.

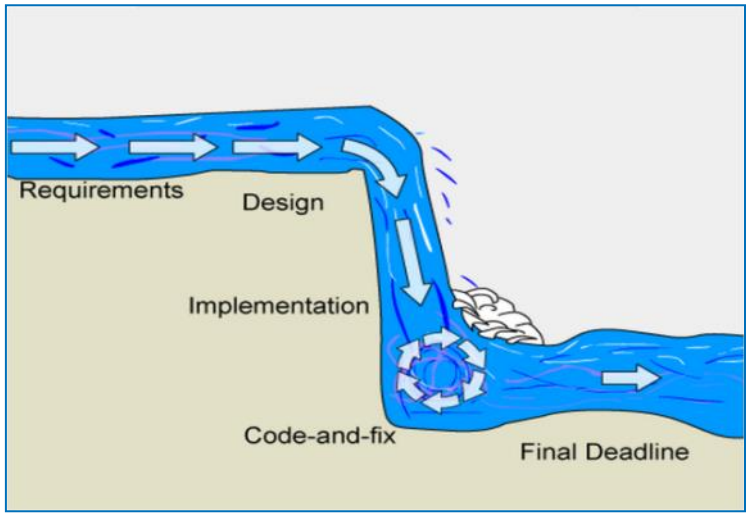
May 2016

Main Themes

- Lean & Agile Product Development works better than Waterfall for medical products
- These methods can enhance regulatory compliance
- How to think, plan and execute in Sprints
- Choosing the right mix of Agile/Lean processes for your products and culture

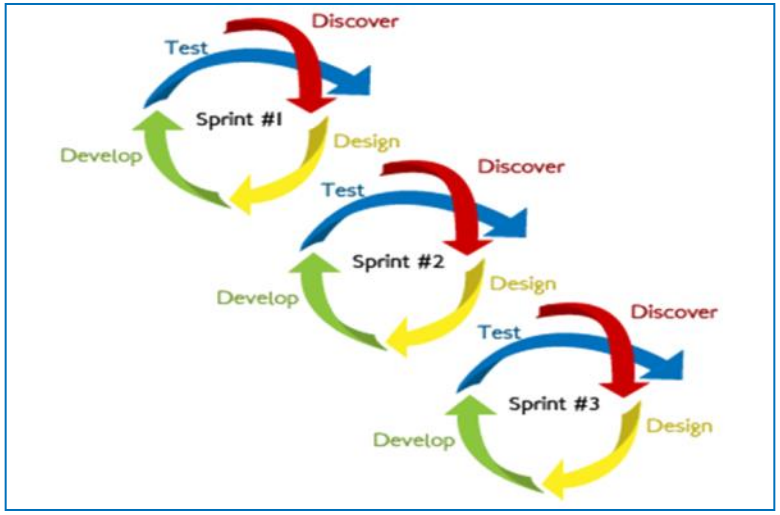
How is Agile Different from Waterfall?

The Waterfall Model



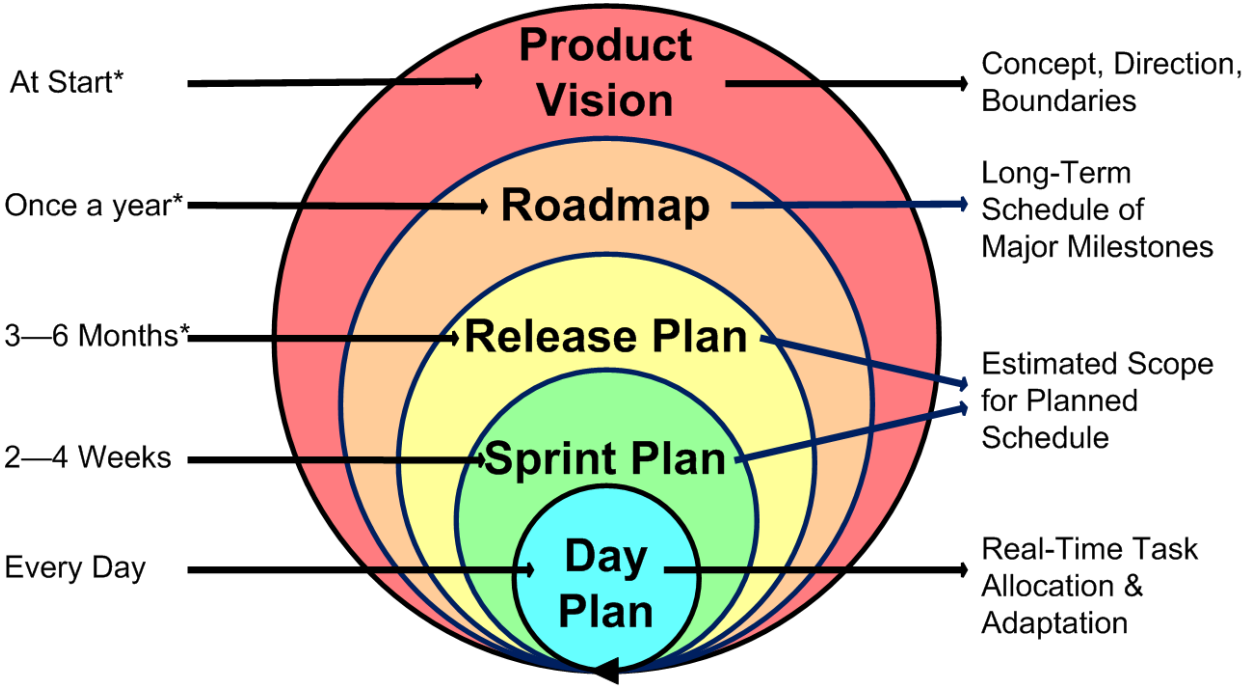
Freeze the Specs and Don't Look Back

Agile Method



Develop Fast, Learn, Improve

Agile Development Time Horizons



* Revise as often as needed (per Release, per Sprint, …)

Advantages of Agile

- Fast prototypes to customer/user
- Hone the product definition during development
- Visibility of work to be done much higher
- Frequent, short bursts of communication
- Fast decision making
- Much wasted time eliminated

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.

Medical Device Regulations ➔ processes & documentation
Solution: find a balance to maintain benefits of Agile while still fulfilling regulatory requirements

Feb. 2001 - http://agilemanifesto.org/

AAMI TIR45:2012 – Guidance on Use of Agile for Medical Device Software

Agile Principles for Software Development

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstances

http://en.wikipedia.org/wiki/Agile_software_development

Agile Development Terminology (Scrum)

- Acceptance Test
- Backlog
- Backlog Grooming
- Burn Down Chart
- Customer Owner or Product Owner
- Information Radiators
- Planning Poker
- Release
- Retrospective
- Scrum
- Sprint
- Stand-Up Meeting
- Timeboxing
- Task Board
- Task Points
- User Stories & Epics

Scrum Definitions - 1

Sprint

- A short, uninterrupted development interval (usually 1 to 4 weeks)
- Software is written and tested
- Resulting product is potentially shippable
- Can be
  - New feature
  - New feature set
  - Improvement or expansion
- Ends with a retrospective

Scrum Definitions - 2

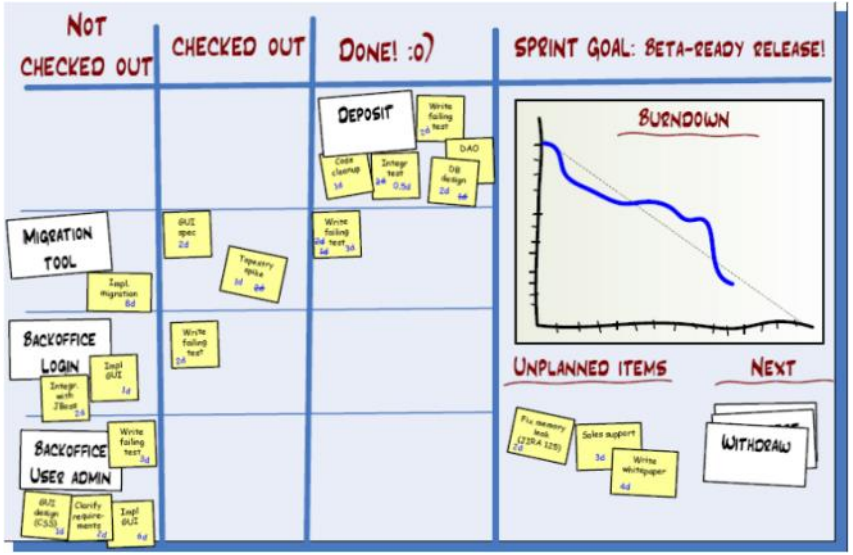
Backlog

- List of product feature requirements
- Backlog may be for one Sprint or for a Release
- Prioritized by the Customer Owner
- Feature requirements are expressed in terms of User Stories
- Backlog Grooming is the process of adding, deleting or re-prioritizing the Backlog

Scrum Definitions - 3

Information Radiators

- Big, Visible Charts (BVCs) for the Whole Team
- Charter
- Sprint tasks
- Backlog
- Burn-Down Charts*
- Task Board with Owners



*Will cover later

Requirements vs. User Stories

Requirements

- Must-have features
- May not be prioritized
- Mostly quantitative
- Needed every release

Example: System boots up to a ready state in less than 60 second

User Stories

- May be desired rather than required
- Prioritized
- Contain qualitative as well as quantitative data
- Not necessarily all needed for any one Sprint

Example: As a user pushing the “on” button, I want the cursor to be ready for my input well within 1 minute, faster if possible so I don’t lose productivity.

Typical Two-Week Sprint Agenda

Before Sprint

- Overall Project Goals & Status
- Results of previous Sprint(s)
- Collect & Prioritize User Stories

Sprint Planning (1/2 day)

- Desired Sprint Outcome (Scoping)
- Estimate Task Durations (Task Points) & Risk
- Plot dependencies / Critical Path
- Prioritize the Backlog & Plot Burn Down Target

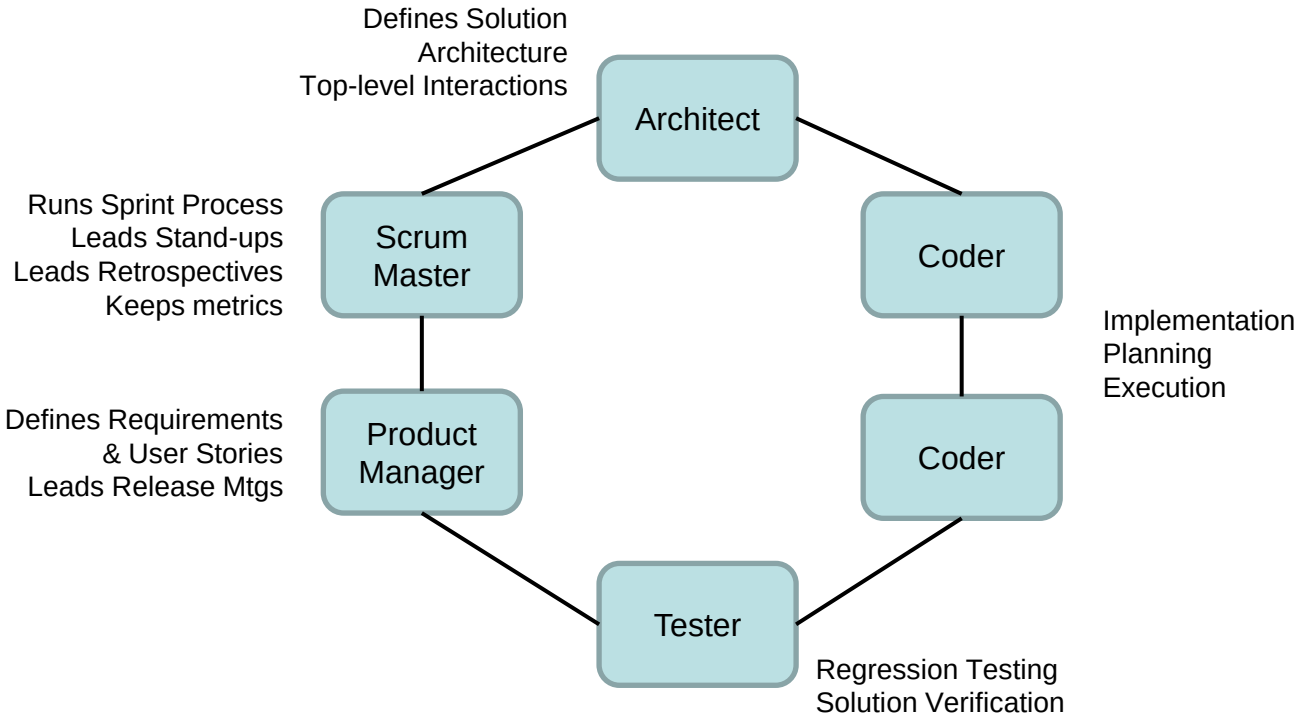
During Sprint

- Daily Stand-ups
- Monitor the progress (Burn Down)
- Do the work individually or in teams

End of Sprint

- Review objectives and accomplishments
- Review overall Project
- Conduct Sprint Retrospective

Scrum Team (Typical)



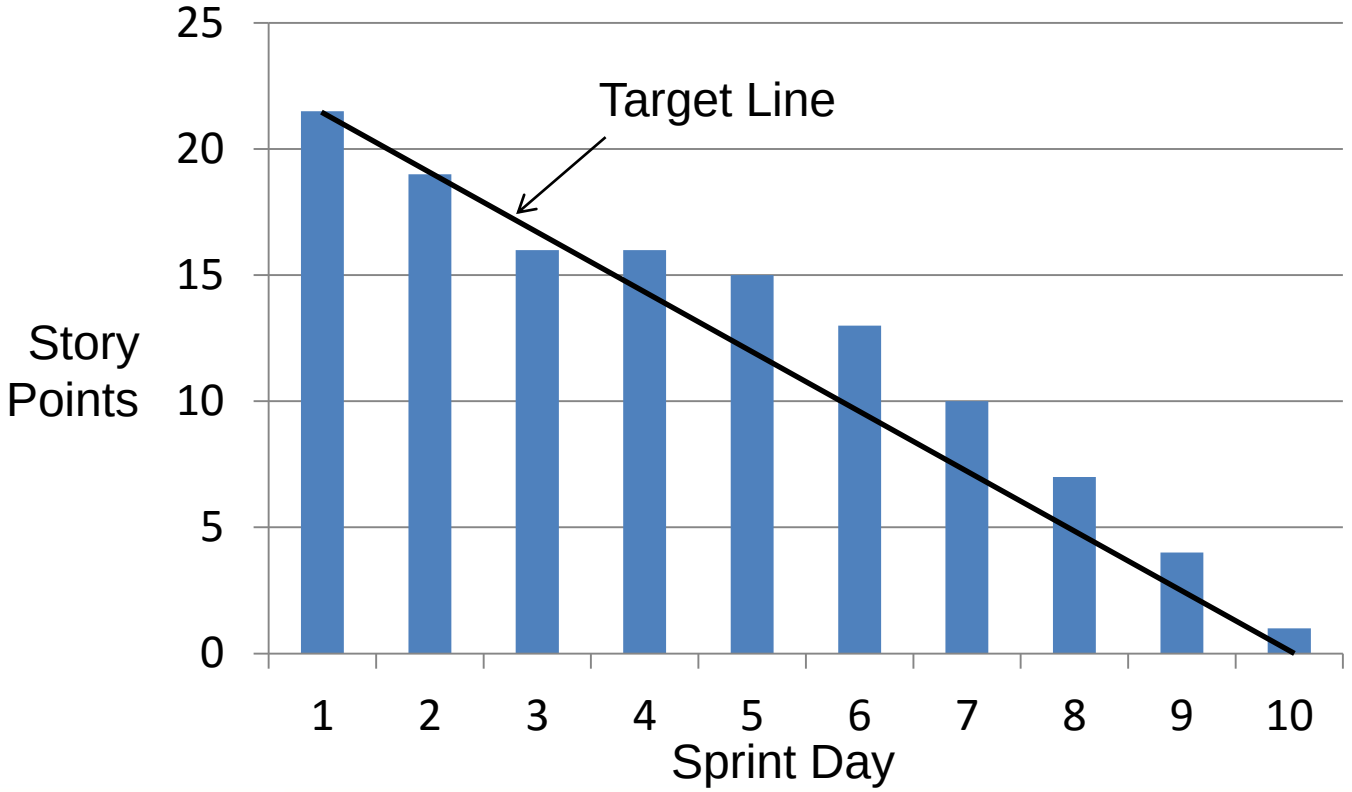
Story Points

- Tasks satisfying User Stories usually require different amounts of time and resources
- To measure progress, we need to normalize tasks
- For a two-week sprint with five programmers, it is typical to assume
  1 programmer-day = 1 Story Point

How many story points might we expect to complete on the sprint?

Can we assume that an equal number of Story Points should be accomplished per day?

Burn-Down Chart for 3 App Example



Can Agile Product Development work for Medical Devices?

- Medical devices are a regulate products
- Medical device regulations govern:
  - Product testing
  - Product manufacturing
  - Product development (“design controls”)
- Each company decides how to implement design controls as part of their quality system
- The quality system procedures can either support lean/agile methods or obstruct them

What's special about medical devices?

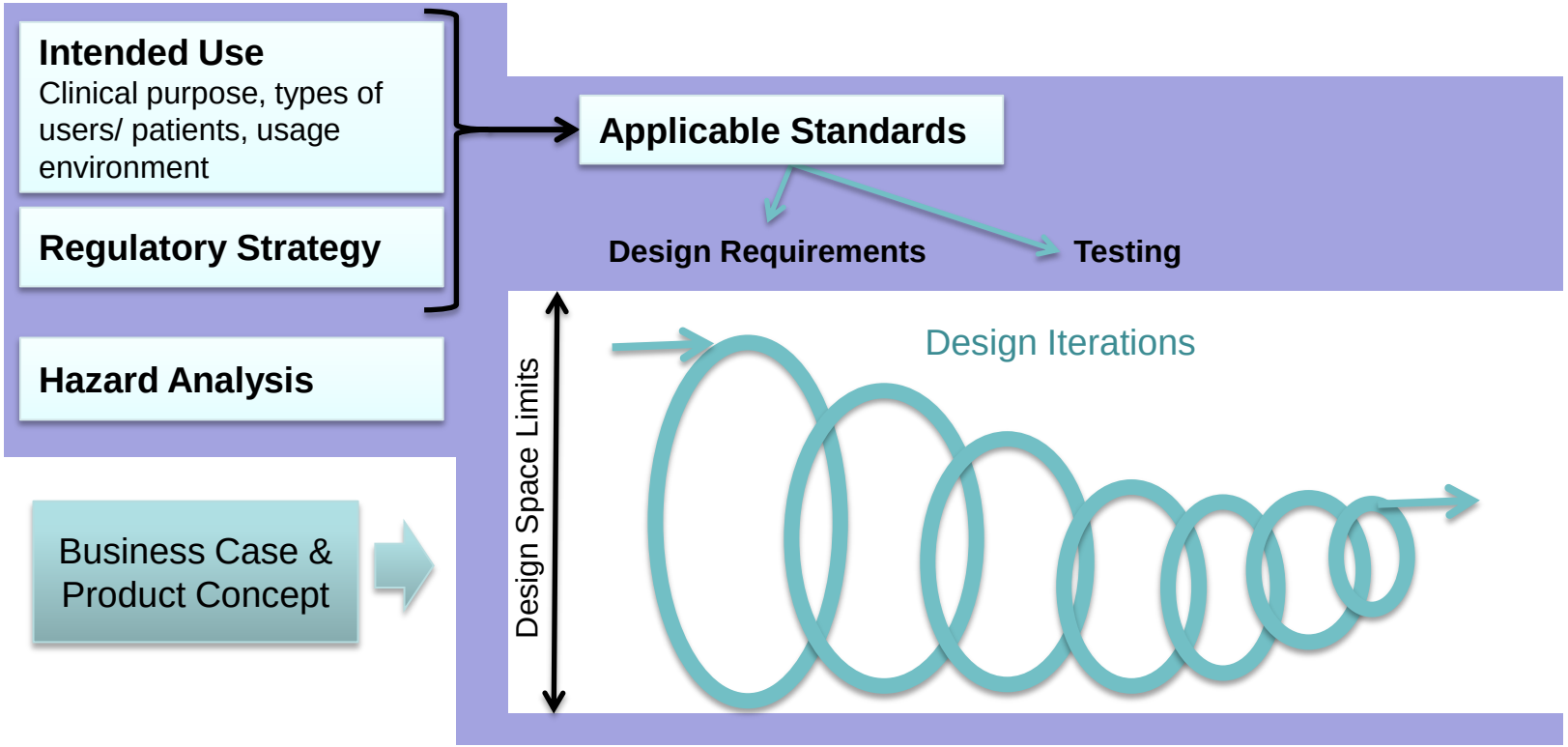
- Products are developed for 2 customers:
  1. Clinical users
  2. FDA (or other regulatory body)
- Double challenge for product development team to fully understand the needs of either group (different mindset from engineering…)
- Product development involves clinical evaluation of finished product and regulatory submissions
- Late breaking problems with a new product trigger a cascade of issues for a medical device company—can lead to regulatory problems on top of technical problems

Exponential increase in compliance/documentation effort near product launch

The Facts:

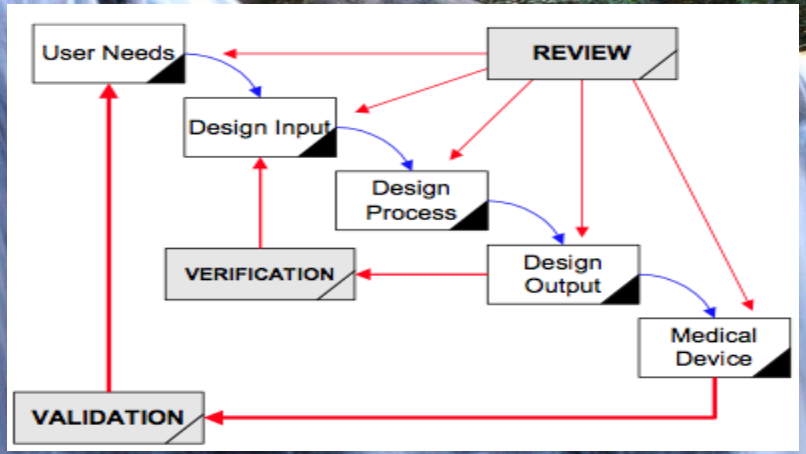
- Lean/agile methods do not conflict with the requirements of FDA regulations, ISO 13485, IEC 62304, or other medical device standards
- Lean/agile methods do conflict with the quality systems of many medical device companies
- Quality system procedures must be adapted for lean/agile methods and the methods must be adapted (with additional documentation steps) for compliance

4 Key Areas Establish Regulatory Framework for Medical Device Product Development



Waterfalls

FDA Design Controls (21CFR820.30)

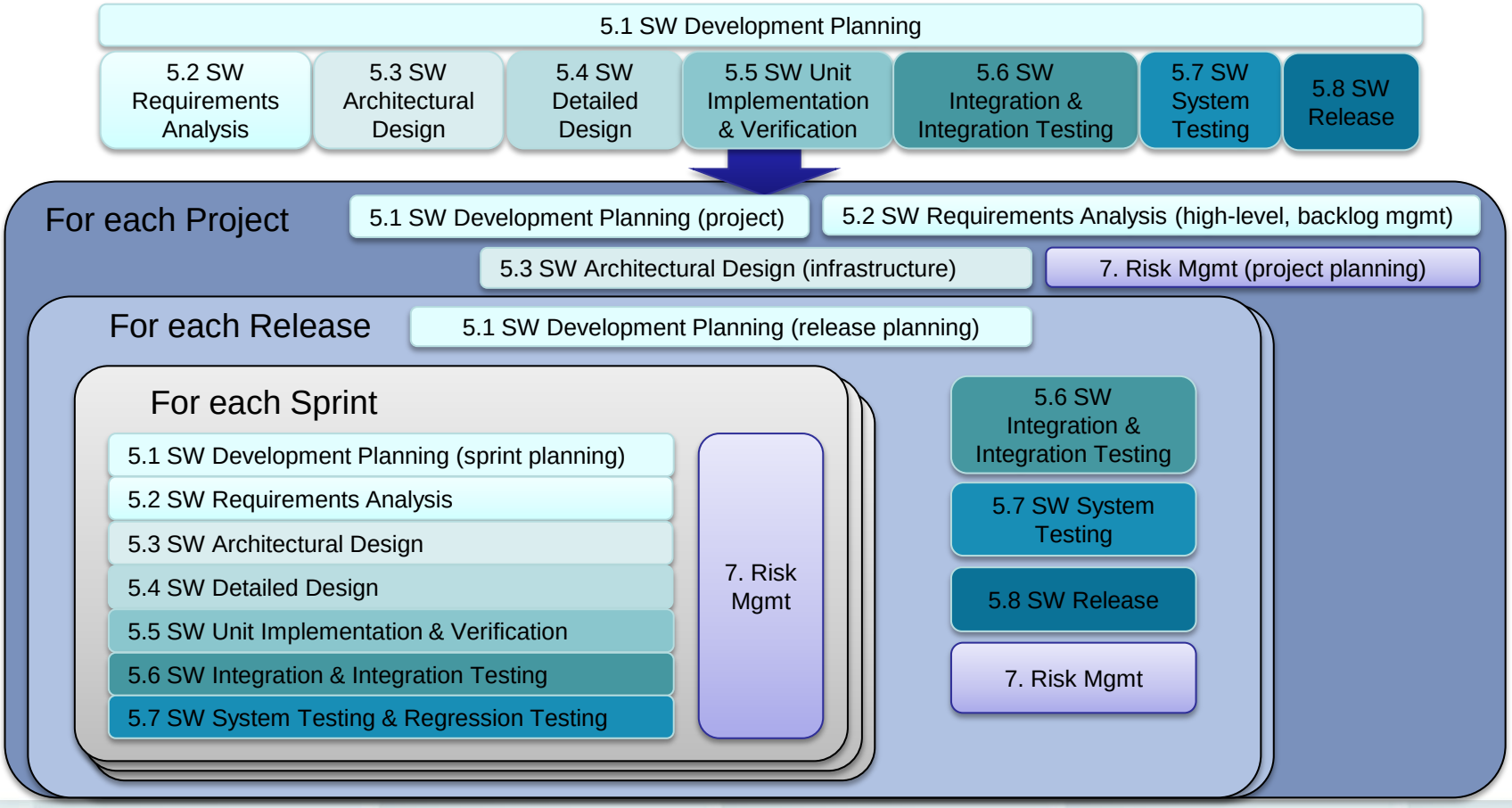


IEC 62304 Medical Device Software Standard

5.1 SW Development Planning

5.2 SW Requirements Analysis	5.3 SW Architectural Design	5.4 SW Detailed Design	5.5 SW Unit Implementation & Verification	5.6 SW Integration & Integration Testing	5.7 SW System Testing	5.8 SW Release

Transposing Waterfall to Agile (IEC 62304 Medical SW Standard)



AAMI TIR45:2012 – Guidance on Use of Agile for Medical Device Software

What is Different About Hardware Development?

- In Software, the Product itself is being developed. For hardware, a set of knowledge on how to manufacture a product is being developed*
- Much longer time to useful product
  - Lead time of materials, parts and shaping
  - Many more people to coordinate
  - Time to develop Manufacturing Processes, Tooling and the Supply Chain
- Much more expense
  - Cost of materials, parts and shaping
  - Cost of prototype runs
  - Inventory holding/scrapping costs
  - Manufacturing Tooling is very costly to obtain and to change
- Electronic circuitry can be more complex and less predictable than software (spurs, nonlinear, etc.)

* For very low volume or “one-off” products, this distinction disappears.

Nonetheless – Agile works for Hardware Development too

The Principles of the Agile Manifesto apply if modified

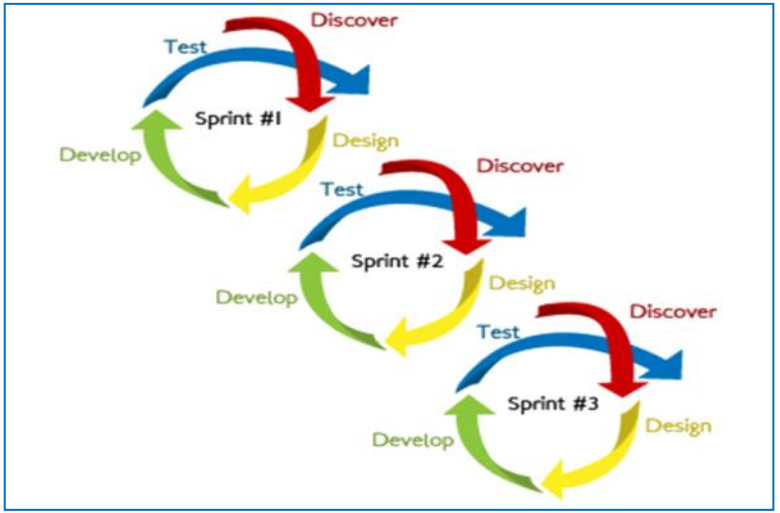
Agile Principles for Hardware Development

1. Customer* satisfaction by rapid delivery of useful Prototypes
2. Welcome changing requirements, even late in development
3. Working Prototypes are delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location or co-wired)
7. Working Prototypes are the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstances

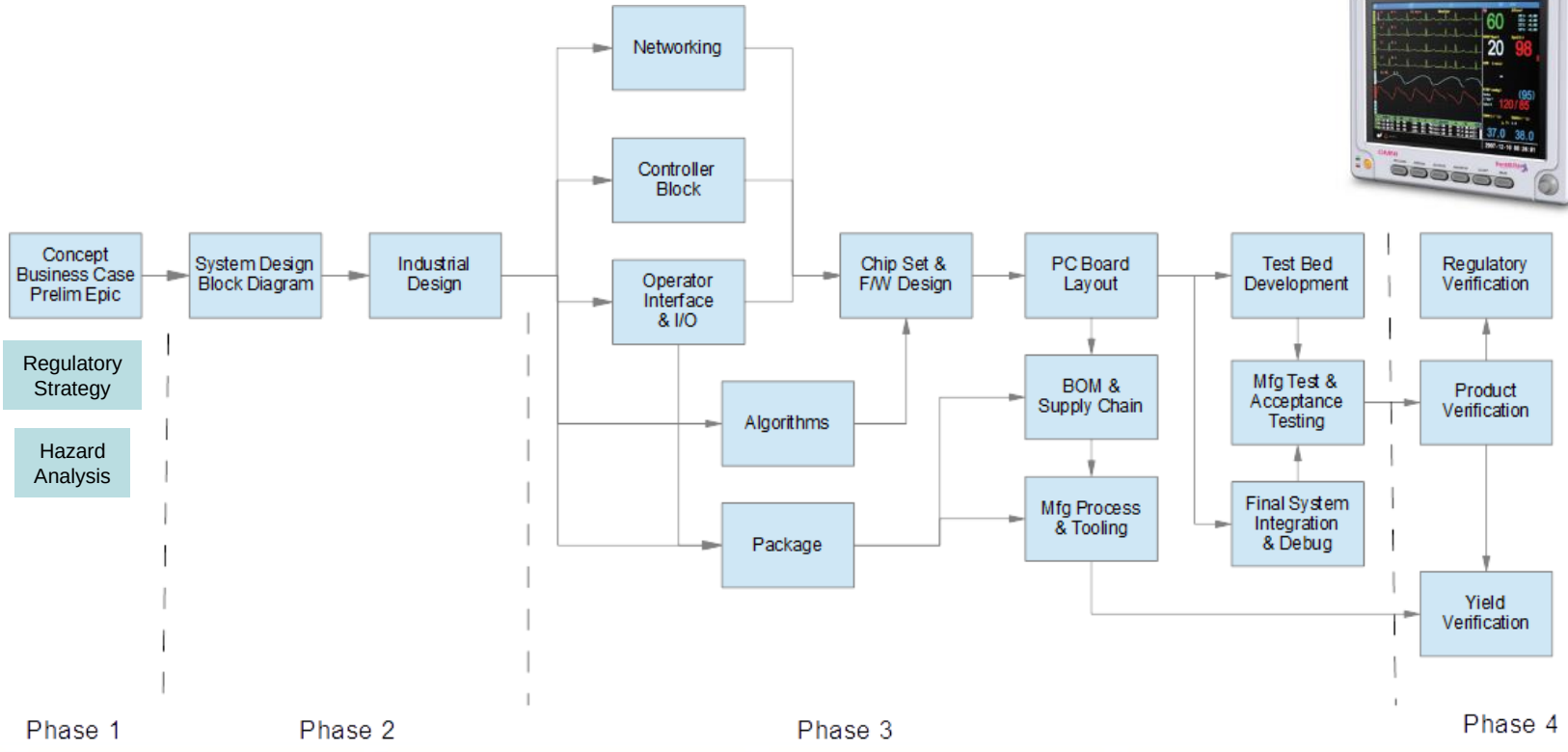
* Here “Customers” may include downstream developers or Manufacturing

Example: Cardiac Monitor Development

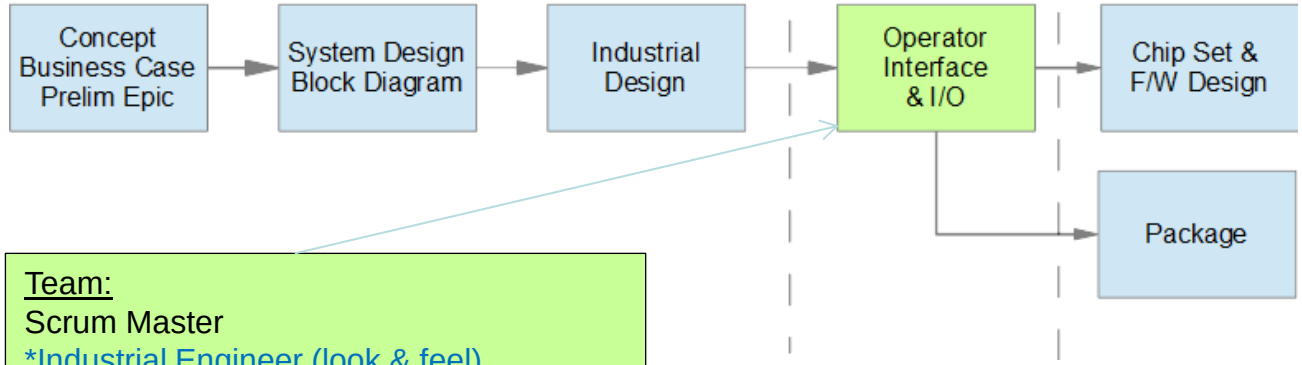
Can we develop this system in Sprints?



Cardiac Monitor Development Flow



Example: Operator Interface & I/O Sprints



Team:
Scrum Master
*Industrial Engineer (look & feel)
*ME (front panel and case)
*EE (Logic & Firmware)
*EE (Interface circuitry)
*Component Engineer (Display & Buttons)
*Medical SME (Medical data needs)
Customer Owner (User/Operator Stories)

*Could be employee or partner
Co-located or remote

Inputs:
Block Diagram
Basic Industrial Design
Prioritized User Stories

Outputs:
Prototype Case & Front Panel
Operator Use Model
Selected Parts & Components
I/O Logic Schematics

(Probably multiple Sprints)

Example: Operator Interface & I/O Sprints

- Customer Owner Visiting & Market Research
- Controller Refined
- Mechanical & Power Needs Refined

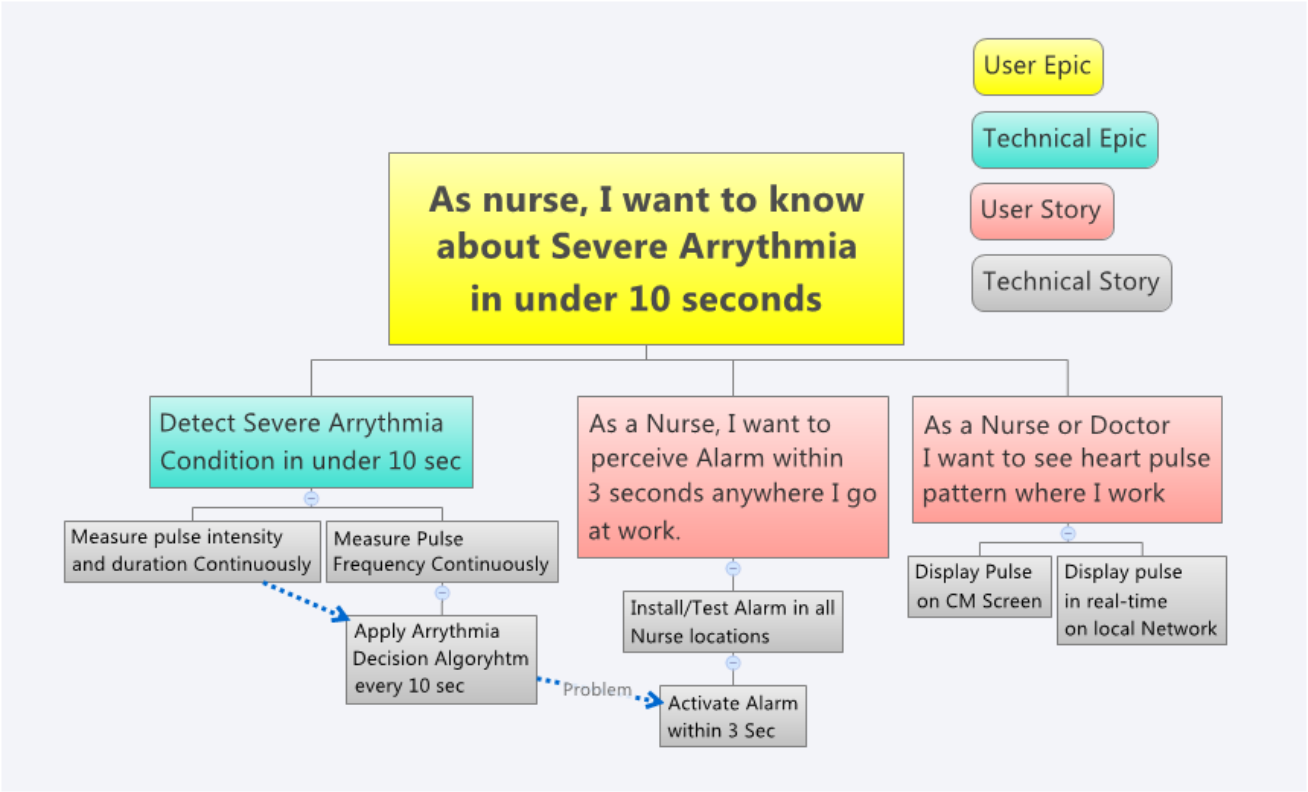
- Customer Owner Visiting & Market Research
- Controller Refined
- Mechanical & Power Needs Refined

Sprint 1: (2 weeks)
- Sprint Planning
- Operator Stories Prioritized
- Story Points estimated
- Layout and 3D CAD of Front Panel
- Touch Screen Supplier Selected
- Block Diagram of I/O Logic
- Program Manager Review
- Sprint 1 Retrospective

Sprint 2: (2 weeks)
- Sprint Planning
- Added Operator Stories & Story Points
- 3D Printed Front Panel
- Touch Screen Breadboard & Test
- I/O Logic Schematics done & Simulated
- Program Manager Review
- Sprint 2 Retrospective

Sprint 3: (2 weeks)
- Sprint Planning
- Added Operator Stories & Story Points
- Prototype Front Panel with working Screen and Buttons
- I/O Logic Emulator Attached
- Sensor Connectors prototype demonstrated
- Program Manager Review
- Sprint 3 Retrospective

Epics & Stories



Typical User “Epic”

Title	Detection of Ventricular Fibrillation				Rank	
ID	37		Estimate		Total Task Est.	

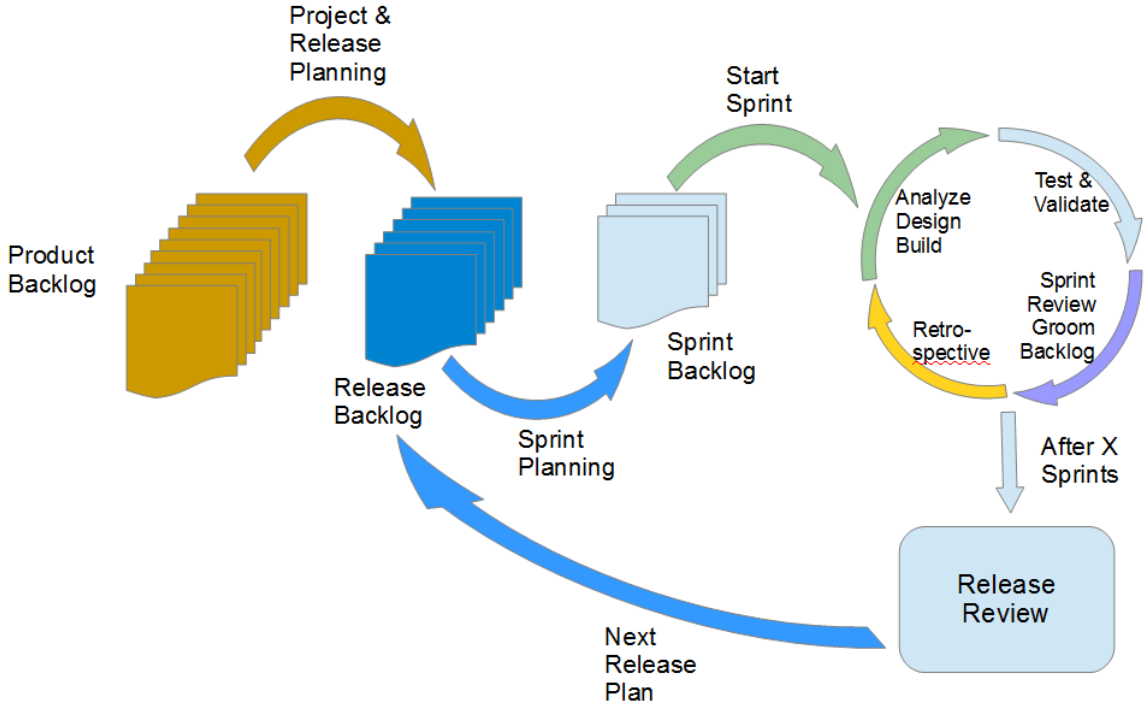
Narrative

As a Doctor, I want to know if a patient is experiencing Ventricular Fibrillation, so that we can initiate immediate treatment to preserve the patient’s life. When fibrillation is detected, the cardiac monitor should notify associated Monitor stations, which will then execute the standard protocols for emergency notification.

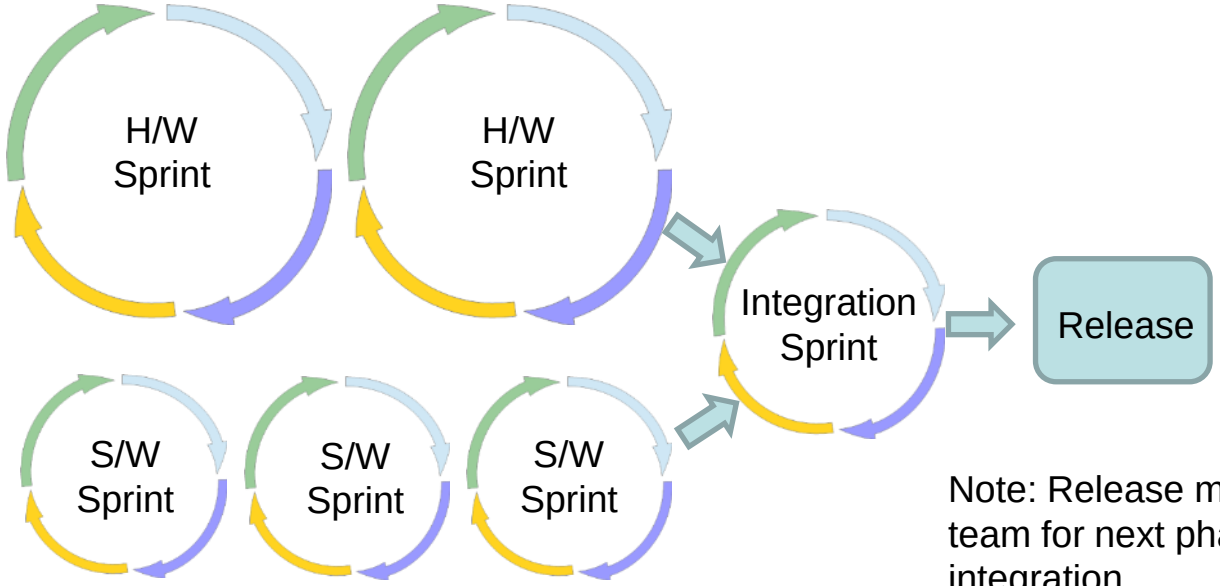
Acceptance Criteria

1. End-to-end notification should occur within five seconds following onset of event.
2. The notification displayed on Monitor stations should include basic diagnostic information.
3. False positives should be made as infrequent as possible, but it is more important that false negatives never occur.

Sprints and Releases



Hardware, Software & Integration Sprints



Note: Release may be to team for next phase of integration.

Scott Elliott
scott.elliott@techzecs.com
+1.415.830.5520

Aaron Joseph
aaronanswers@gmail.com
+1.408.529.2490

For your kind attention

Connect with us on LinkedIn!
https://www.linkedin.com/pub/dr-scott-elliott/0/15a/a09
https://www.linkedin.com/in/ajosephprofile