



Copyright

© Copyright 2010 Crisis Prevention and Recovery, LLC. (CPRLLC), all rights reserved. SoftwareCPR® is a DBA of Crisis Prevention and Recovery, LLC and the SoftwareCPR® logo is a registered trademark.

SoftwareCPR® authorizes its clients and SoftwareCPR®.com subscribers use of this document for internal review and training. Any other use or dissemination of this document is expressly prohibited without the written authorization of SoftwareCPR®. Individual original FDA documents in their original form without SoftwareCPR® annotations are public information and may be shared without restriction.

Legal Disclaimer

The training document that follows should only be applied in the appropriate context with oversight by regulatory and software professionals with direct knowledge and experience with the topics presented. The document should not be used as a cookbook since it is not adequate for some situations and may be excessive for others.

While SoftwareCPR® attempts to ensure the accuracy of information presented, no guarantees are made since regulatory interpretations and enforcement practices are constantly changing, and are not entirely uniform in their application.

Disclaimer of Warranties: The information is provided AS IS, without warranties of any kind. CPRLLC does not represent or warrant that any information or data provided herein is suitable for a particular purpose. CPRLLC hereby disclaims and negates any and all warranties, whether express or implied, relating to such information and data, including the warranties of merchantability and fitness for a particular purpose.

Agile methods for medical device software ...

Can it be compliant?
Can it be safe?



SoftwareCPR® is a software regulatory, safety, compliance, quality, and management consulting firm. We **specialize** in project and regulatory crisis recovery in addition to training, preventive action, and continuous improvement.

Our **expert** staff and **partners** are industry practitioners – most have 25+ years experience in the medical device industry at technical levels through executive management. We provide strategic and hands-on assistance to assure the success of your software investments.

Our approach focuses on your business objectives, project and regulatory risk management, and **pragmatic approaches** that are tailored to your internal culture. We work closely with your internal staff, external vendors, your FDA counsel and FDA itself.

Can we use Agile methods?

★ YES

- Very common question among medical device companies.
- Resources:
 - AAMI developing a Technical Information Report (TIR) on the use of agile methods for medical device software
 - ASQ will be releasing a position statement on agile

Can we use Agile methods?

★ YES

How should we do it?

Make sure the elements of IEC 62304 are satisfied using the flexibility inherent in it.

“You may find that some agile methods, when implemented properly, are not only compliant with regulatory law & guidance, but are better than traditional methods for ensuring safety and effectiveness.”

Why use a software process standard?

- Provides some assurance that what you are doing is consistent with established state of the practice.
- Creates a common “checklist” of expectations between manufacturer and regulator.
- Provides a common language to communicate to regulators
- Regulatory expectation or requirement is increasing

IEC 62304 background

- Specifically created for medical device software
- IEC 60601-1-4 and general software engineering standards were not considered adequate
- Significant FDA involvement from start
- Scope includes “stand-alone software” and “embedded software”
- Based on ANSI/AAMI/SW68 with a few significant differences
- Omits requirements duplicated elsewhere (QMS)
- Adds requirements considered essential for medical devices (safety aspects)

Status of IEC 62304

- Approved by both IEC and ISO as an international standard (joint development effort)
- Adopted by CENELEC as EN and harmonized 11/08 under the MDD, AIMDD and IVDD
- Adopted by ANSI as US national standard (replacing ANSI/AAMI/SW 68)
- Recognized by FDA for use in premarket submissions
- China – SFDA adopted 62304
- Translations exist in French, German, Spanish, Chinese and Japanese
- In final phase of adoption as a Japan Industry Standard
- ANSI/AAMI/IEC 62304 is identical to IEC 62304:2006 as is EN 62304:2007 which will be the harmonized one

IEC 62304 - What is it?

- A framework – processes, activities and tasks
 - Process is the top level; a process has activities and an activity has tasks. Specific requirements in IEC 62304 are generally at the task level.
- Identifies requirements for what needs to be done and what needs to be documented
- Specifies a software safety classification scheme
 - Additional requirements apply as safety becomes more important
 - Much more significant than minor/moderate distinction in SW68

IEC 62304 – What's not in it?

- Does not prescribe how to accomplish requirements
 - Not a “how to” with defined methods or practices
- Does not require a specific software life cycle
- Does not specify documents
 - What to document, not where it must go.
- All of these decisions are left to the manufacturer – within reason?
- Does not address “validation” – SW68 did – Why?

Key to FDA compliance

- FDA is flexible on lifecycle, methods, and documentation organization and format
- Ad hoc, informal development and validation is not acceptable
- Internal Plans and SOPs define your approach
- Conformance to your plans and SOPs is required
- THIS IS CENTRAL TO ALL ASPECTS OF A FORMAL QUALITY SYSTEM

IEC62304

- Section 5.2.1 Process Definition, Note 2:

“These activities and tasks may overlap or interact and may be performed iteratively or recursively. It is not the intent to imply that a waterfall model should be used.”

FDA

GPSV - 5.1. SOFTWARE LIFE CYCLE ACTIVITIES

“This guidance does not recommend (specify) the use of any specific software life cycle model. Software developers should establish a software life cycle model that is appropriate for their product and organization.”

Three Common Pitfalls (and possibly misconceptions)

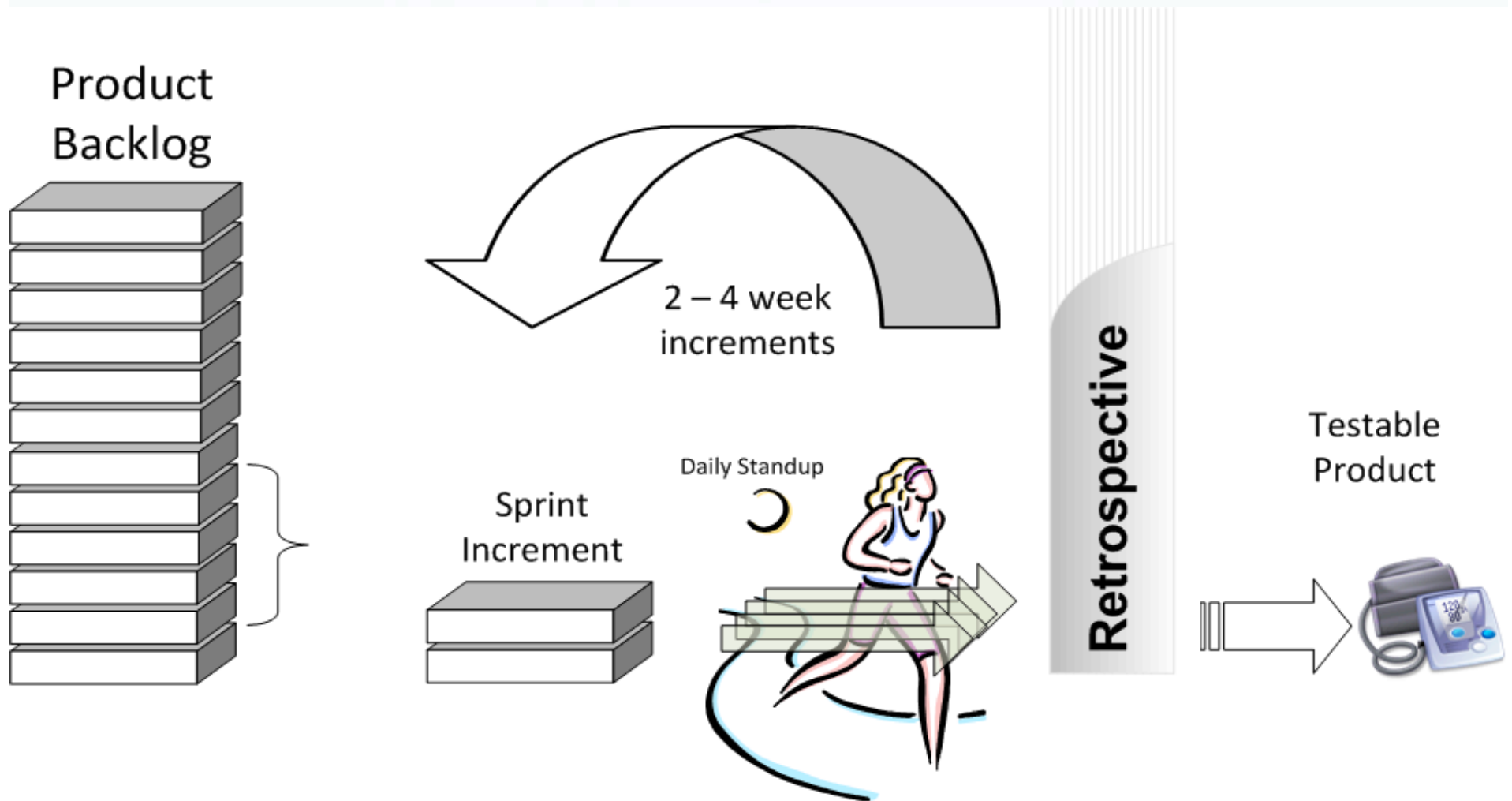
“Agile methods are not suited for medical device software development because ...”

- the lack of formal requirements
- no formal verification and validation
- no formal process for ensuring all hazards are properly mitigated

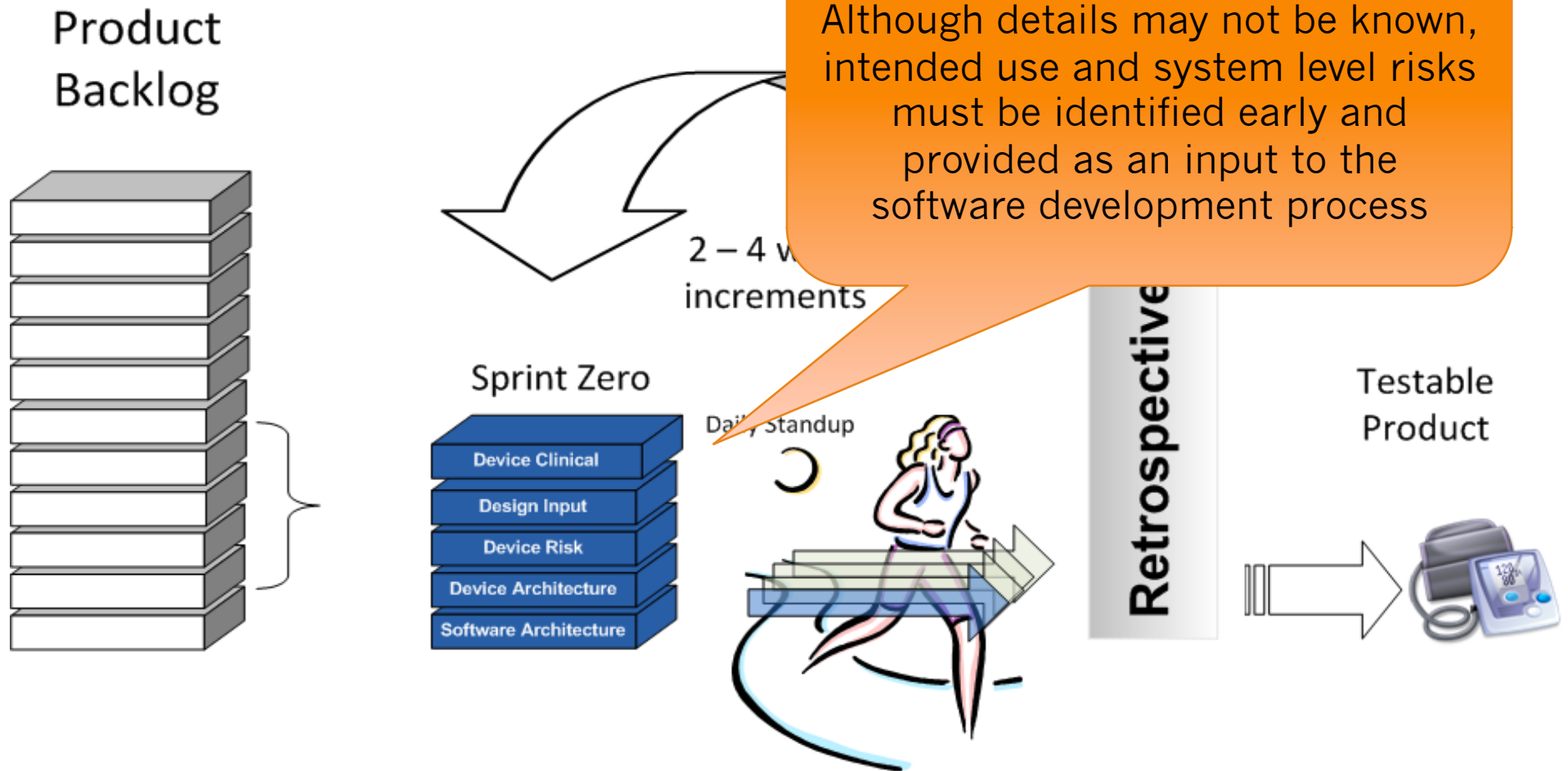
Pitfall #1: “Agile methods could be non-compliant because of the lack of formal requirements.”

- Planning process and documentation may neglect design inputs
- Plans and procedures may not address increment planning and process for determining increments
- May get overly focused on software (implementation) and not capture requirements (essence)
- May improperly focus on non-safety, non-efficacy related requirements
- Subsequent sprints may re-factor implementation but fail to update requirements
- no CM for requirements documentation

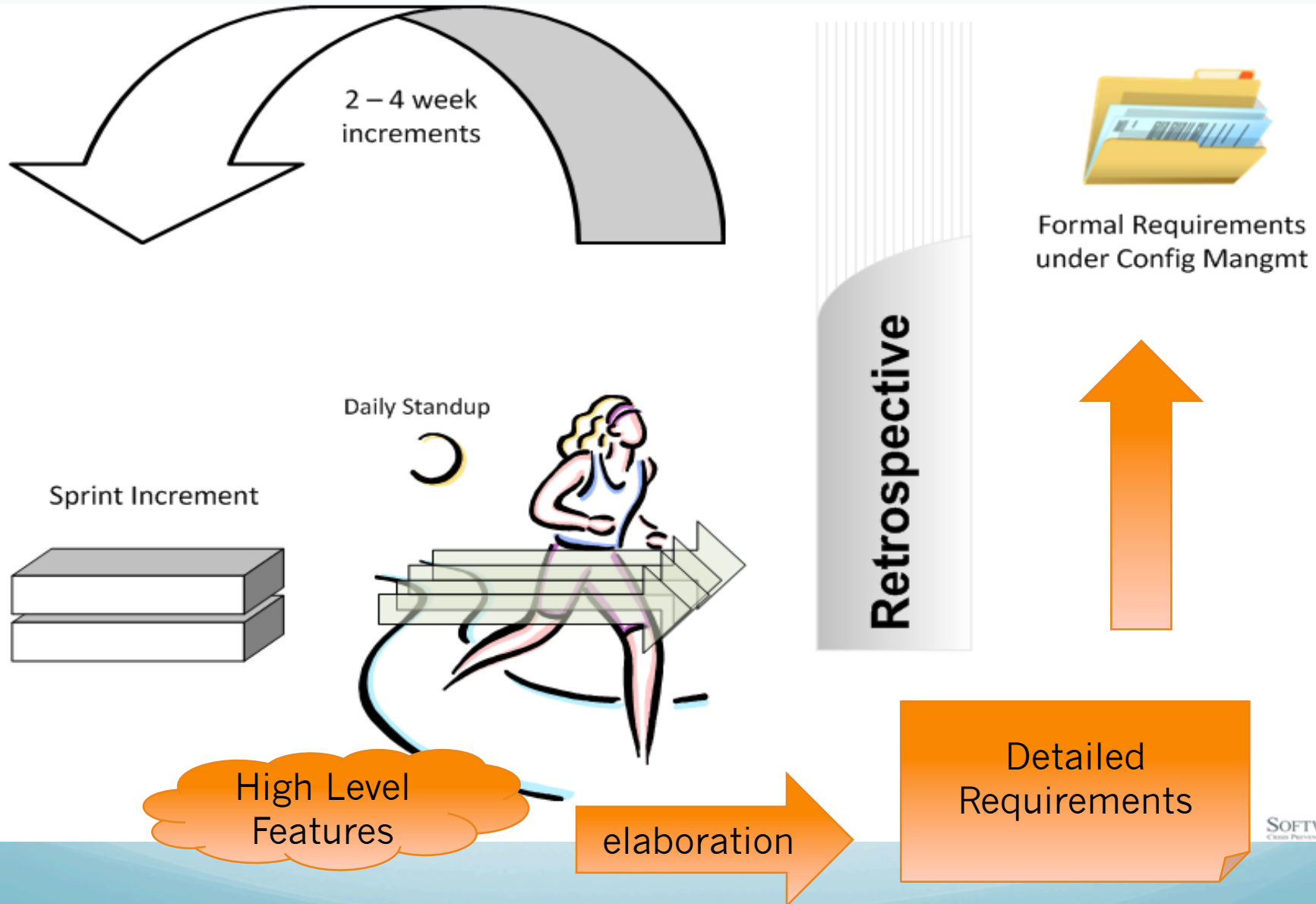
Using a scrum approach as an example ...



Critical Discipline Point: **Special Sprint Zero**



Discipline Point: Capture Requirements During Sprint



Discipline Point: “Requirements” are central to everything, BUT don’t need to be at same levels (waste)



Pitfall #2: “Agile methods are not suited for medical device software development because there is no formal verification and validation.”

- May get overly focused/weighted toward unit testing
- May omit formal technical reviews with evidence
- May not perform proper regression testing or know when to perform regression testing
- May fail to capture integration and system level test cases during sprints
- Self-managed teams may neglect formal testing altogether!

Discipline Point:

Capture test cases during the iteration

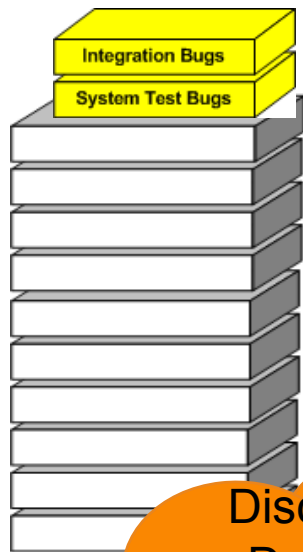
- Each sprint should capture unit testing for the development of that sprint
- As the system grows, each sprint should also capture integration and system level test cases as appropriate for the system interactions added during that sprint
- Integration and system test failures should be captured in a defect tracking system
- Test and spec approvals prior to running formal tests during certain identified sprints
- All test documentation is under CM

Testing and V&V

Discipline Point:
Track bugs from integration
and system testing and
feedback to Product Backlog.

Test Procedures
under Config Mangmt

Product
Backlog

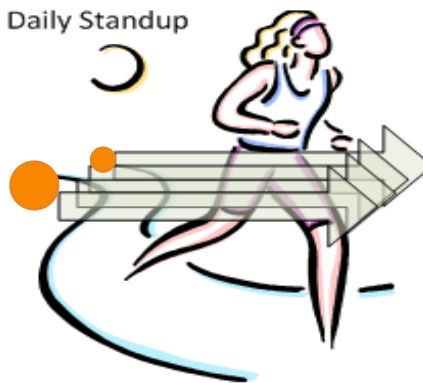


Discipline Point:
Don't forget to
capture integration
and system test
cases during each
sprint!

Sprint
Increment

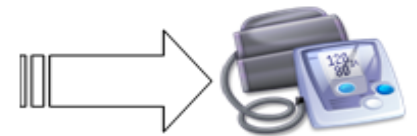
2 – 4 week
increments

Daily Standup



Retrospective

Testable
Product



Pitfall #3: “Agile methods are not suited for medical device software development because there is no formal process for ensuring all hazards are properly mitigated.”

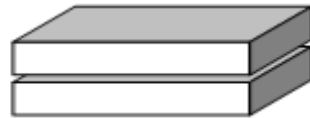
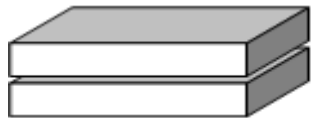
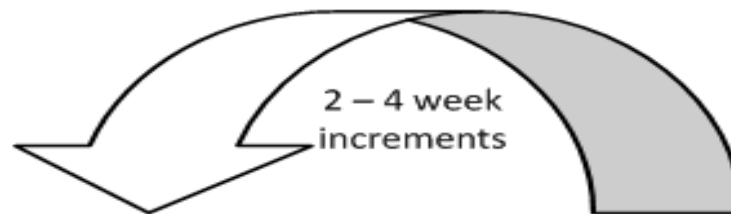
- While system level hazards may be known, sprint team may be not identify software-cause hazards
- Team may equate zero-defects with risk-free
- Even if hazards are mitigated, mitigation may not be formally documented for test
- Subsequent sprints might alter mitigation software and inadvertently dilute the mitigation effect

Discipline Point: **Add RM activities to sprints**

- Each sprint should capture any new hazards or causes identified and document in risk management file
- Each sprint should capture any mitigations developed in that sprint and capture in requirements documentation
- Phase planning should account for reviews of the RM documentation
- Consider a designated risk team member to participate in selected sprints or reviews

Critical Discipline Point:
Sprint Zulu – Final system and device testing, risk
review, open issues reviews

Product
Backlog



Daily Standup

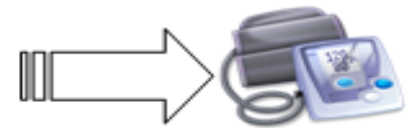


Retrospective



Device Integration and
System Testing

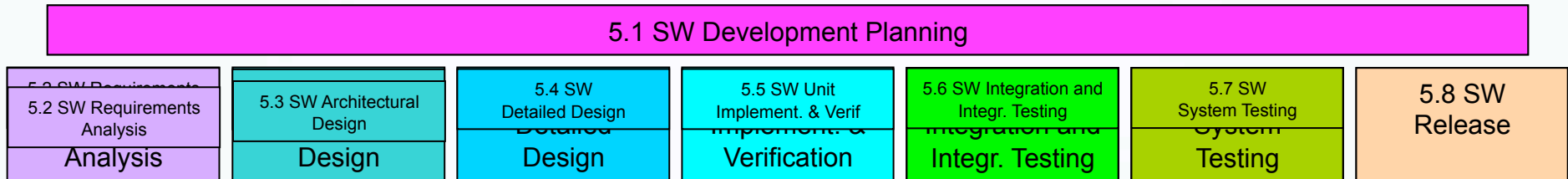
Testable
Product



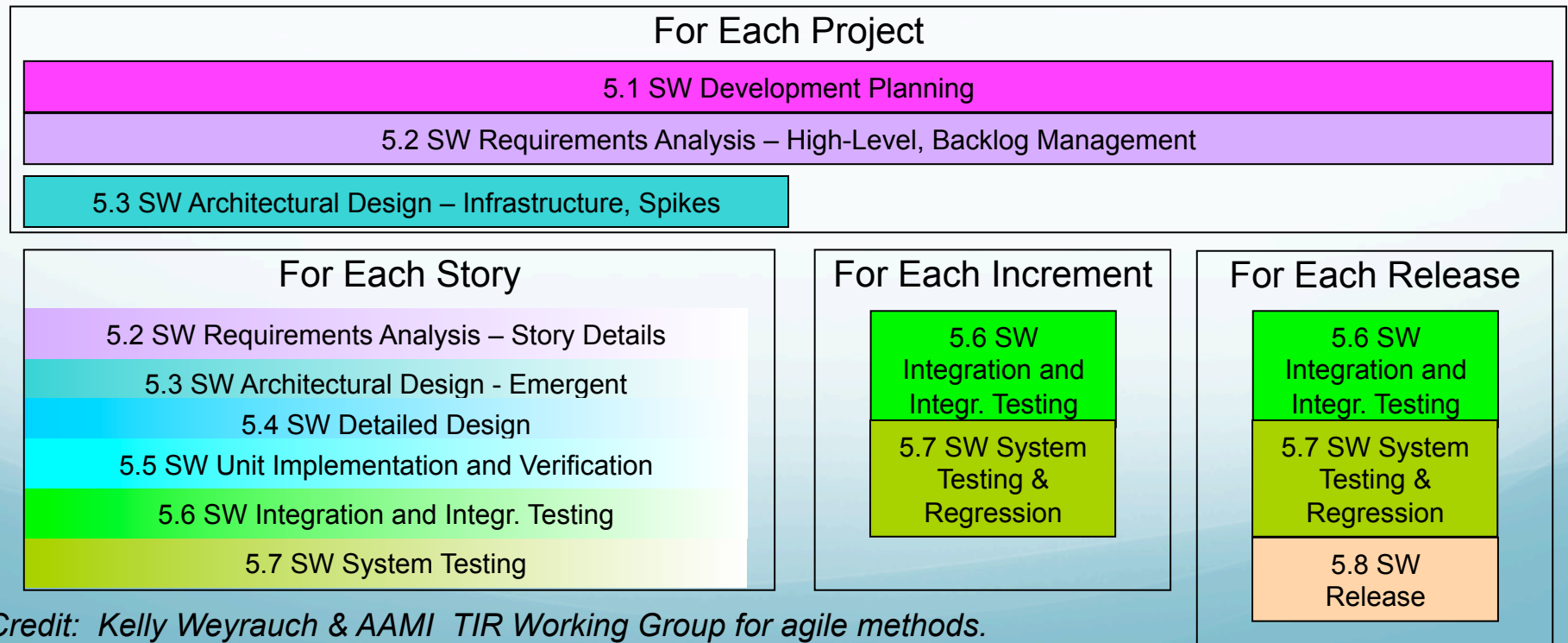
Medical device software “gotta haves”

- Software development plan
 - for both pre-production and post-production (maintenance)
- Software Architectural Design
- Detailed Software Requirements for:
 - safety-related software - software that
 - if it fails could lead to hazard
 - detects hardware failures that could lead to hazard
 - core clinical performance - software that performs primary function of medical device, e.g. heart rate algorithm
- Traceability analysis showing
 - requirements to design/code
 - requirements to test cases
 - linkage of mitigations (from risk analysis) to requirements and verification
- Written test procedures for testing detailed software requirements.
- Verification and validation plan
 - providing overall strategy for establishing safety and efficacy of software, particularly regression test strategy for changes
- Problem resolution plan or procedure
 - identifying process for addressing design changes both pre-production and post-production
- Risk management plan and analysis
 - showing system level and software level hazards, pre-mitigation risk score, mitigations, post-mitigation risk score, and residual risk analysis.
- Configuration management plan or procedure
- Software media release/control plan
- Usability testing plan or procedure
- Documented test results
 - identifying test case, software/hardware configuration tested, clear indication of pass/fail, and for any failures, clear linkage to resolution/disposition

Mapping 62304's activities...



...into Agile's Incremental/Evolutionary life cycle



Credit: Kelly Weyrauch & AAMI TIR Working Group for agile methods.

Conclusion

- Think Lean
- Understand the “intent” of the regulations and/or standards
- Inspect and Adapt
- Construct your quality system to allow tailoring of approach without losing discipline
- Tyranny of the “or” – be agile and compliant

SoftwareCPR Consulting

Successful Software Development in a Regulated Environment

- ✓ Transformation for organizational agility – adaptable, able to change at speed, and compliant
- ✓ Methodologies, key practices and discipline points, and culture

Training

- EN 62304 software development processes and related software standards; Making your agile-type process EN 62304 compliant (or vice versa)
- ISO 14971 Medical Device and Software Risk Analysis; Assist with system and software hazards analysis using the pre- and post- mitigation evaluation; Using Fault Tree Analysis approach for hazards identification and analysis
- Medical Device Software Verification and Validation
- FDA and EU Quality System compliance
- Integration Of Software Usability Engineering Into ISO 14971 Risk Management Using IEC 62366 as a guide

Regulatory support

- Articulation in FDA Terminology; Planning and reviewing
- FDA interaction and negotiation–inspections, submissions, injunctions, and consent decrees; inspection-readiness audits
- Software information sections ; Full submission preparation; Deciding when to submit a new 510(k)
- MDR evaluations, Field Corrections and Recalls
- EU requirements

Website information service and knowledgebase

A subscription to our website provides access to complied FDA software related warning letters and recalls, SoftwareCPR® checklists and example training documents, software related regulatory news, guidance, and standards.

Next Public Training Course

SoftwareCPR® is running its third public offering of its course entitled “Efficient Use of Medical Device Software Standards 62304, 60601-1 PEMS, TIR32, 80001-1, and 80002-1 for Safety and Regulatory Compliance”.

When: Nov 9 - 11, 2010

Where: Burlington, MA USA (near Boston).

IEC 62304 is an EU harmonized standard. FDA is performing internal training on this standard, and other regulatory authorities are adopting it. This makes it important that software, QA, and RA staff have although understanding of this and related medical device software standards and their relationship to FDA requirements as well. This 3 day course will be taught by Sherman Eagles and Alan Kusnitz of SoftwareCPR®. It will provide in-depth practical coverage of both requirements of these standards and practical efficient and effective implementation approaches, including similarities and differences with FDA guidance and expectations. The course includes extensive emphasis on software risk management, including the recently released 80002-1 Medical Device Software Risk Management technical report.

Sherman Eagles was a Medtronic Staff Fellow and chair/convenor for 62304 and 60601-1 PEMS and other standards, so his perspective is excellent and important to fully understand the intent of the standards.

Alan Kusnitz was co-chair for AAMI TIR32 Medical Device Software Risk Management, contributor and reviewer for 80002-1, reviewer for 62304 and was on the committee that developed SW68 the pre-cursor to 62304.



- Contact:

Brian Pate
813.766.0563 (office/cell)
brian@softwarecpr.com

Mike Russell
813.468.9675 (office/cell)
mrussell@softwarecpr.com