

AAMI TIR45

Agile Meets Software Standards,
and We all Win

Brian Shoemaker, Ph.D.
ShoeBar Associates



4th SDMD Europe, 27.-30. January 2014

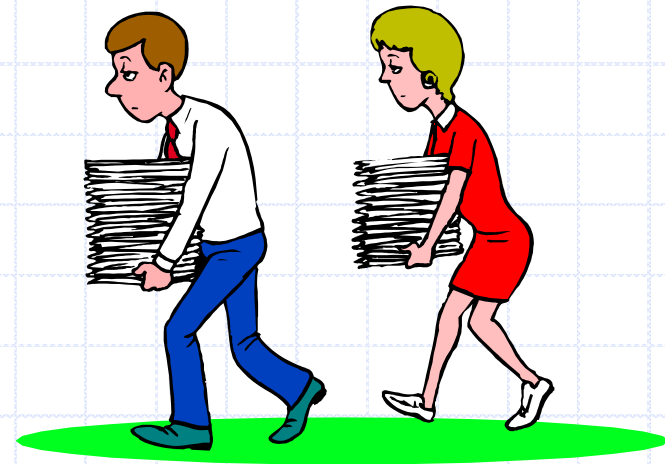
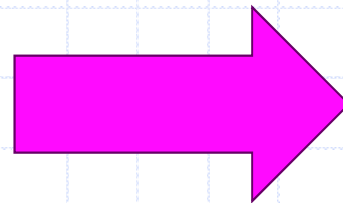
Who I Am

- Originally an analytical chemist
- 15 y in clinical diagnostics (immunoassay):
analytical support → assay development → instrument software validation
- 6 y as SW quality manager (5 in clinical trial related SW)
- 9 y as independent validation consultant to FDA-regulated companies – mostly medical device
- Active in: software validation, Part 11 evaluation, software quality systems, auditing, training

Agile Meets SW Standards, and we all Win

- TIR 45 comes at a much needed time
- *TIR 45 stitches together the important high-level concepts*
- *TIR 45 outlines key practices needed for flexibility and quality*
- *Implementation issues are not ignored*
- *This TIR is actually just a starting point*

SW Quality: All cost, no benefit?



Common view:

- Software quality processes / validation is purely a cost
- The steps are required but add no value

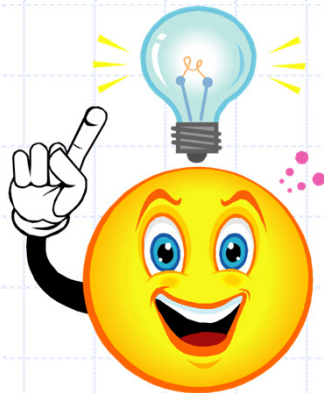
My question was always – isn't there a better way?

The old way is broken!

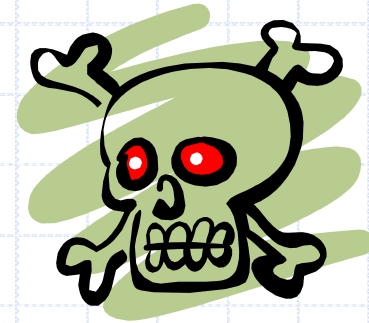
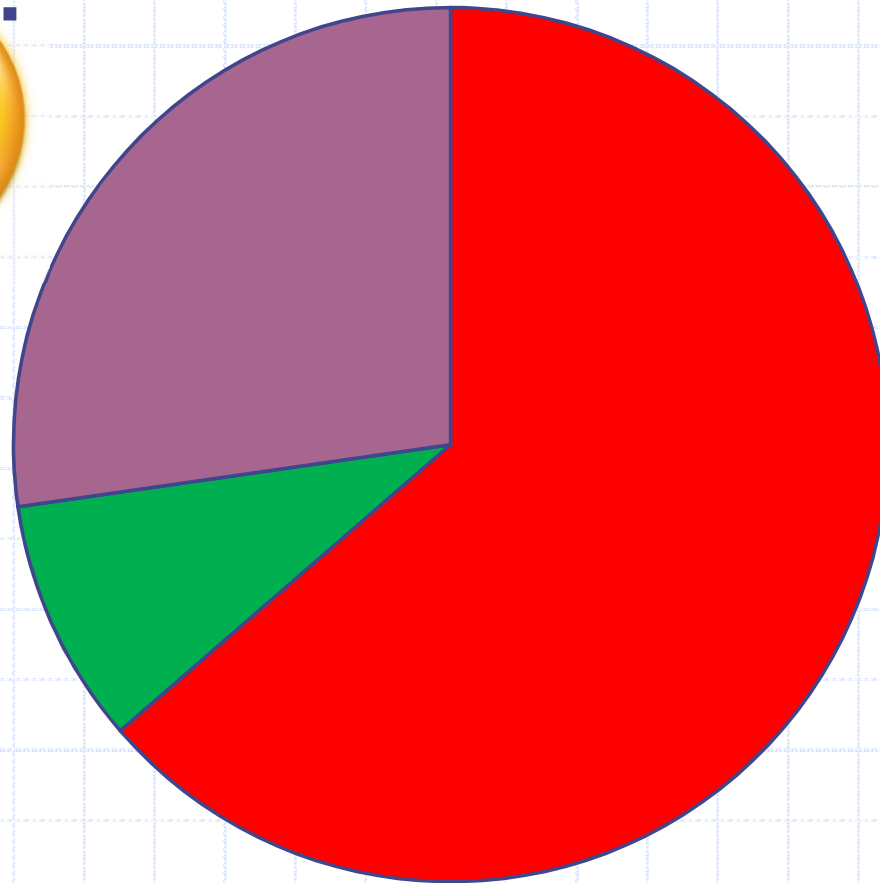


Source: 2011 Annual Report, U.S. FDA Center for Devices & Radiological Health

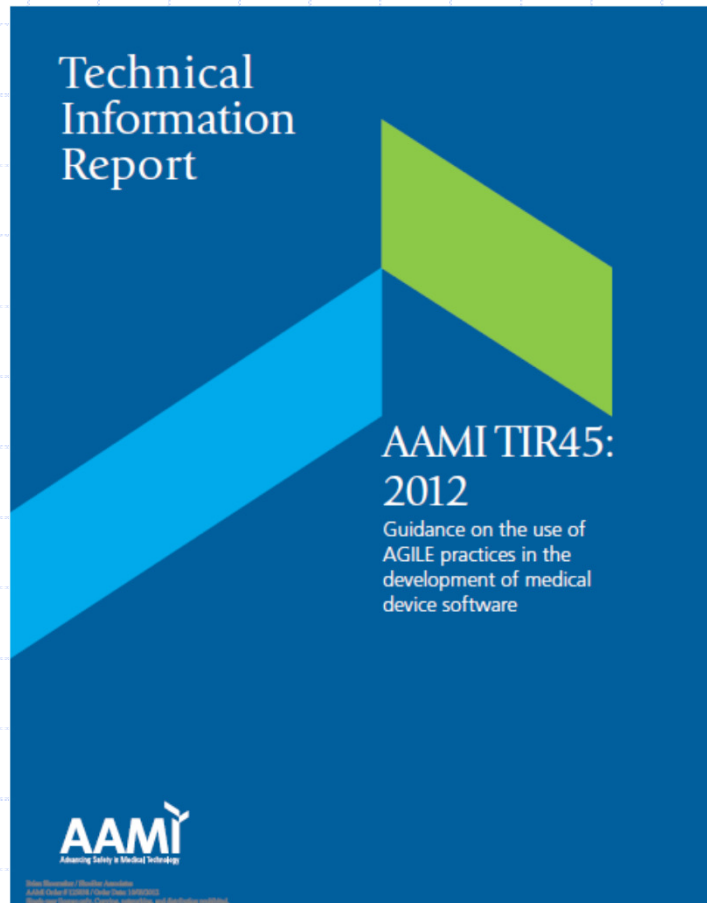
RA/QA Reaction: Mixed



(This is me)



AAMI Agile Doc – game changer?



- I was excited to learn this was being developed
- Even more so when I had the opportunity to review it!
- Document issued in August 2012
- My view: not a moment too soon

Agile Meets SW Standards, and we all Win

- *TIR 45 comes at a much needed time*
- **TIR 45 stitches together the important high-level concepts**
- *TIR 45 outlines key practices needed for flexibility and quality*
- *Implementation issues are not ignored*
- *This TIR is actually just a starting point*

Intro statements say it all

From the introduction to TIR 45:

- Agile can bring value to medical device software.
- Agile can be adapted to the unique needs of medical device software.
- Apply the values of Agile in a way that enhances a robust quality management system.
- Apply the practices of Agile within the context of an established quality management system.
- Set the correct expectations by defining the software development lifecycle model. Demonstrate how an incremental/evolutionary lifecycle satisfies regulatory requirements.
- Establish robust change management systems to manage changes and mitigate risks associated with rapid change.

... But it pays to take a closer look!

NOTE: quotes from TIR 45 are boxed in purple, as above

Agile Manifesto Values

- ***Individuals and interactions***
- ***Working software***
- ***Customer collaboration***
- ***Responding to change***

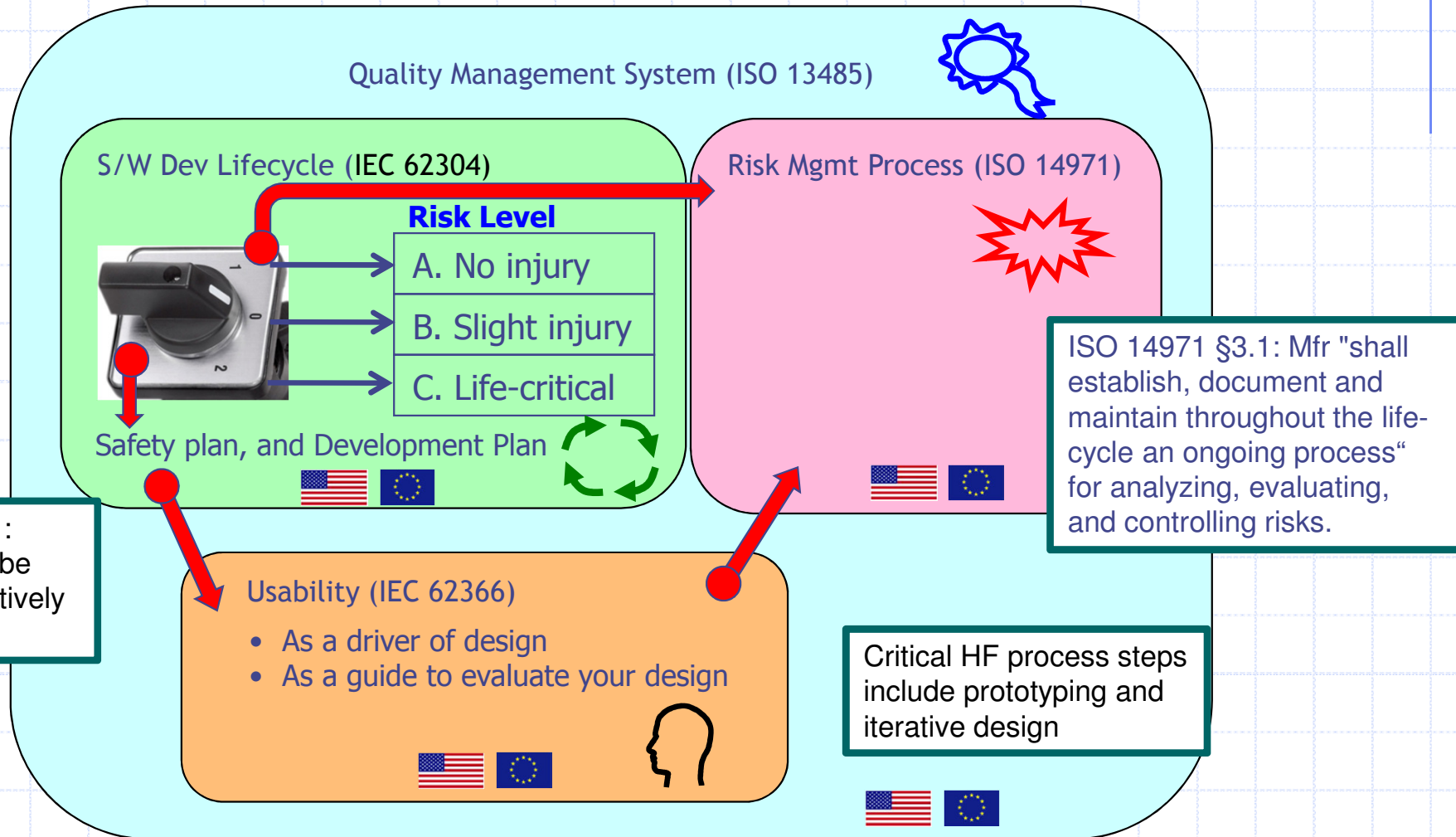
More Value here ...

- ***Processes and tools***
- ***Comprehensive documentation***
- ***Contract negotiation***
- ***Following a plan***

But these still have value!

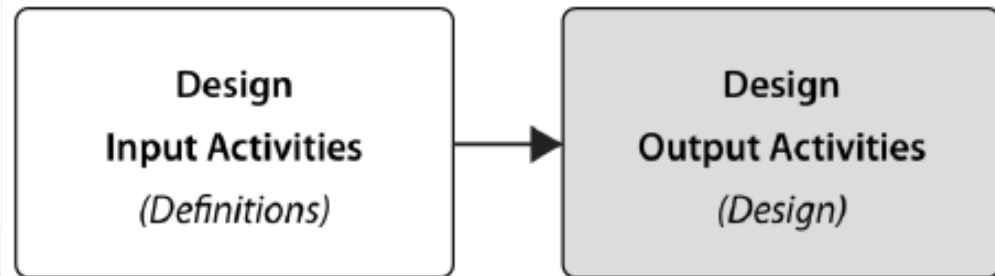
***This does not conflict with device quality standards
(ISO 13485, 21 CFR 820)***

Standards DON'T Dictate Waterfall

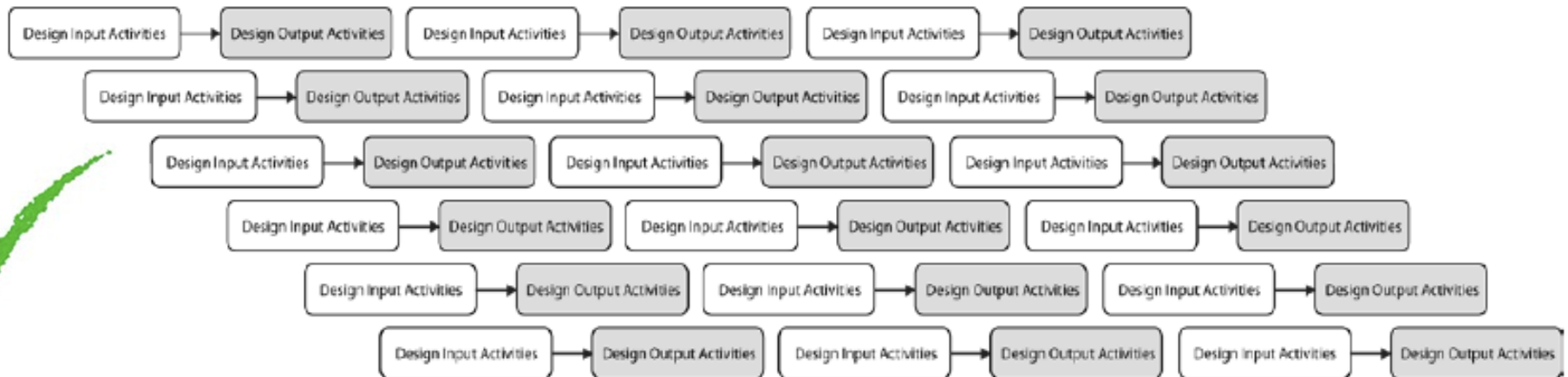


Design Inputs / Outputs: Pieces

Not monolithic:

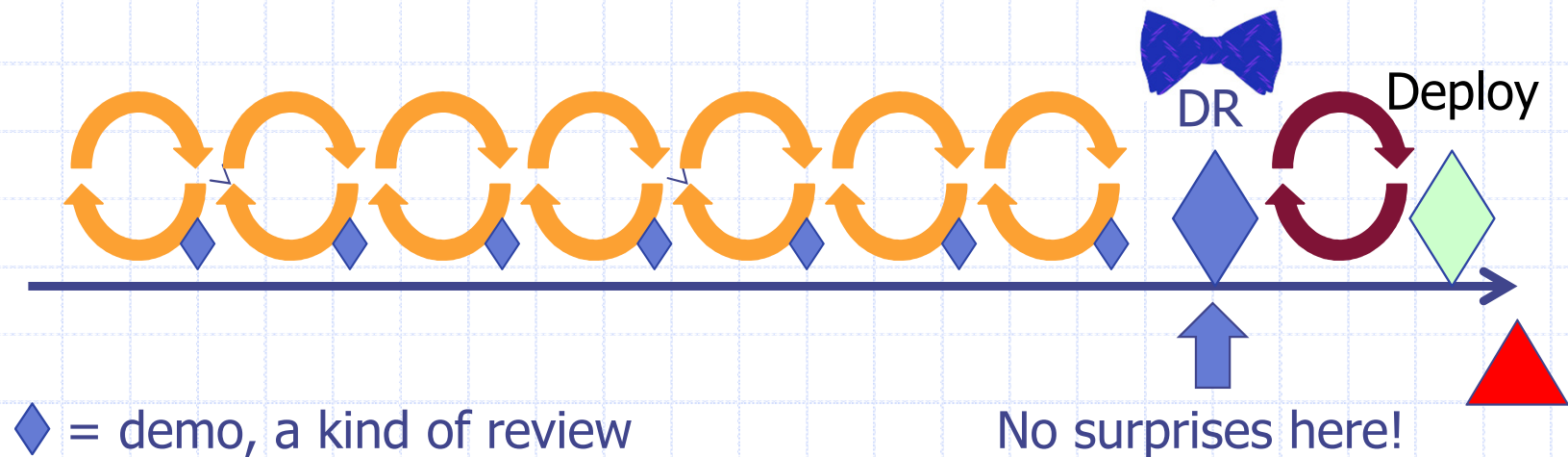


But rather much smaller,
incremental pieces:



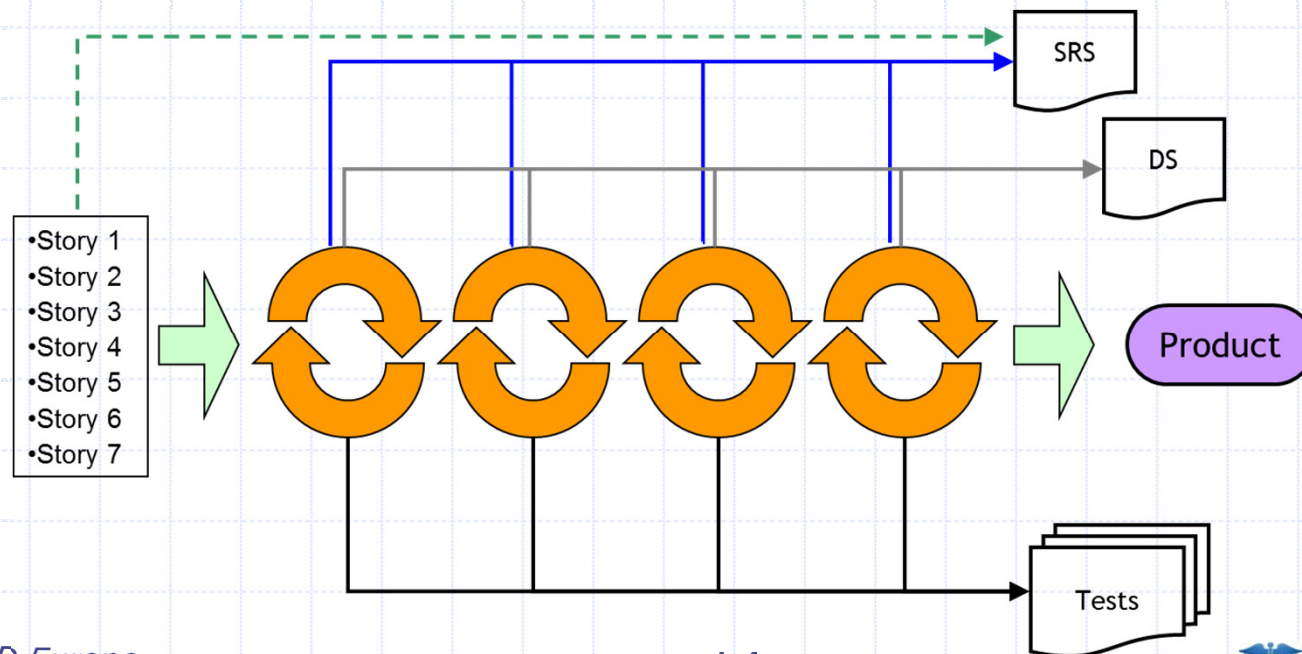
Design Reviews: Part of the Lifecycle

- *In software development plans, define the reviews that take place, showing that they satisfy regulatory requirements.*
- *Plan formal design reviews to be performed at increment and release boundaries.*



Documentation: A Deliverable

- *Produce documentation that communicates information to the intended audience.*
- *Produce documentation at the point in time when it fits the flow of creating it and using it.*
- *Define how documentation is written, controlled, and approved as a sum-of-its-parts.*



Agile Meets SW Standards, and we all Win

- *TIR 45 comes at a much needed time*
- *TIR 45 stitches together the important high-level concepts*
- **TIR 45 outlines key practices needed for flexibility and quality**
- *Implementation issues are not ignored*
- *This TIR is actually just a starting point*

Planning - Incremental

Each Project

5.1 SW Development Planning - *Project*

5.2 SW Requirements Analysis – *High Level Backlog Management*

5.3 SW Architectural Design – *Infrastructure, Spikes*

Each Release (*multiple releases*)

5.1 SW Development Planning – Release

Each Increment (*multiple increments*)

5.1 SW Development Planning – Increment

Each Story (*multiple stories*)

5.1 SW Development Planning - *story*

5.2 SW Requirements Analysis - *story details*

5.3 SW Architectural Design - Emergent

5.4 SW Detailed Design

5.5 SW Unit Implementation & Verification

5.6 SW Integration & Integration Testing

5.7 SW System Testing

5.6 SW
Integration &
Integration Testing

5.7 SW System
Testing & Regression
Testing

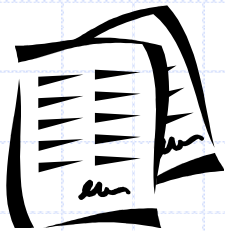
5.6 SW
Integration &
Integration Testing

5.7 SW System
Testing & Regression
Testing

5.8 SW Release

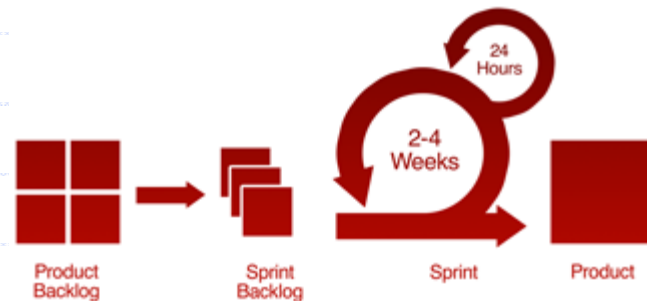
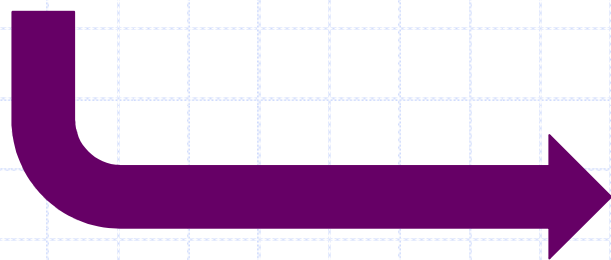
Plans can be more formal or less

Formal – high level



Goals
Resources
Milestones
Deliverables

An Agile team will find that they need more than a backlog and release strategy to cover some of these planning topics. They now will have to write formal plans around such subjects as testing (at all levels), risk management, and software configuration management. A good way to remain Agile is to document the high-level strategy / resources / schedules / milestones and use the story creation / backlog / increment / release management to plan and execute detailed tasks. Together, they form the software development plan for a project.

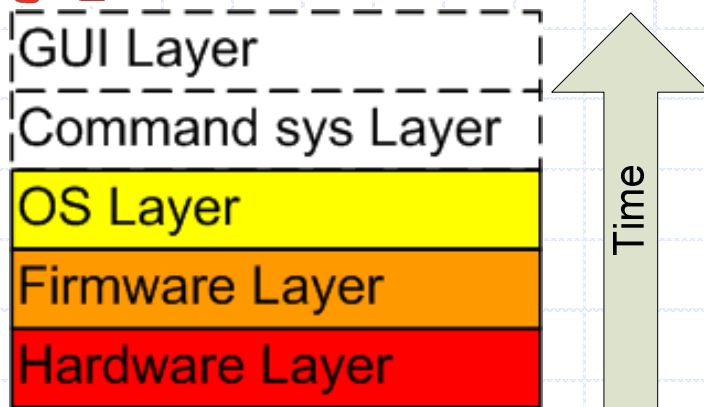


Less formal
(emergent details)

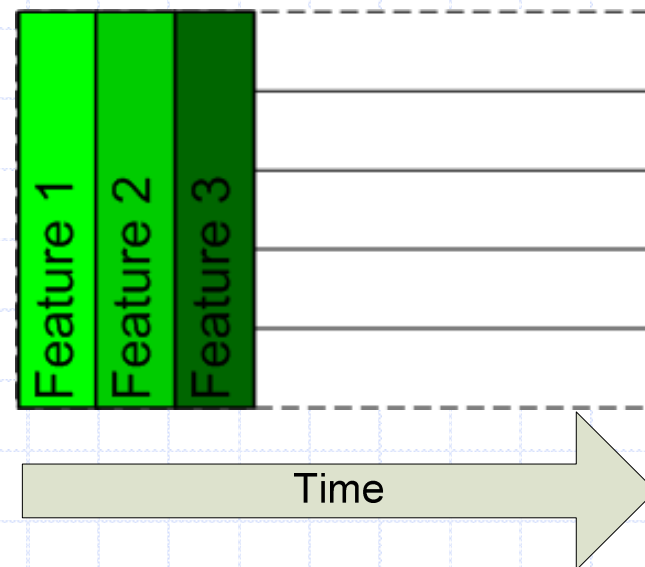
Continuous Integration – for success



Not This:



But This:



Use Agile's continuous integration practices as the core of an effective integration strategy.

Note that this strategy also confirms any OTS software (SOUP)!

Pairing – for more than code



Pairing can be used effectively for requirements, design, and testing as well as coding. Benefits include:

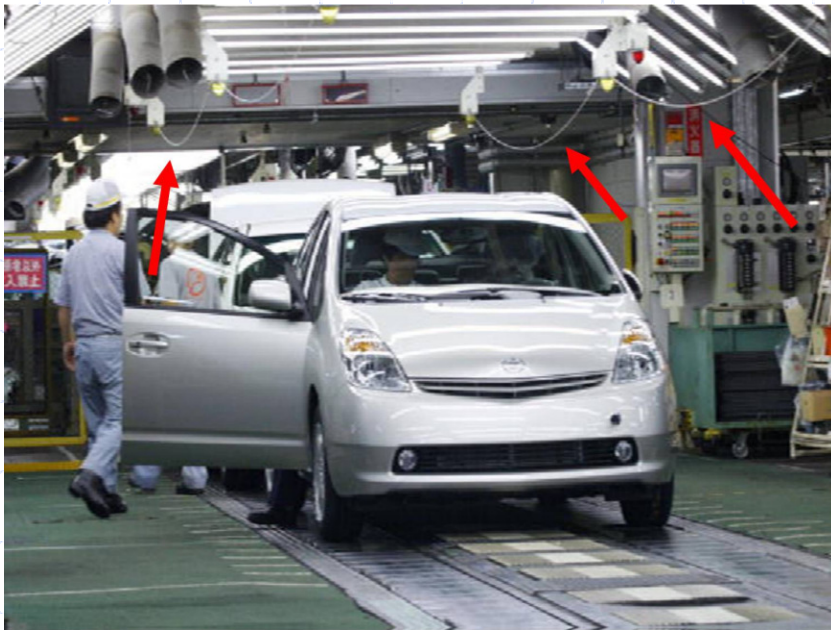
- Better designs
- Effective training / mentoring
- Reduced risk of knowledge loss
- Improved quality via constant review

Pairing can also serve as peer review if the company addresses regulatory concerns:

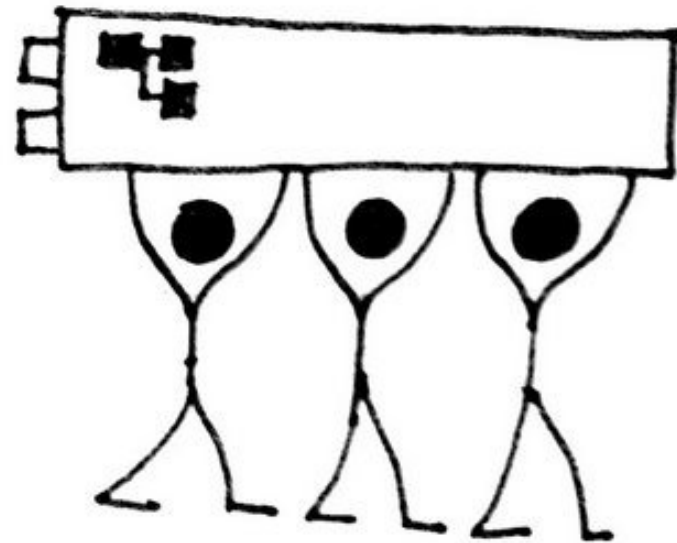
- Established acceptance criteria
- Defined reviewer qualifications
- Documentation of results

Pairing as a form of peer review (as part of the overall verification process) must satisfy the same requirements as any other method of peer review, addressing the considerations described above.

Quality – everyone's job



“Stop the line” (andon)

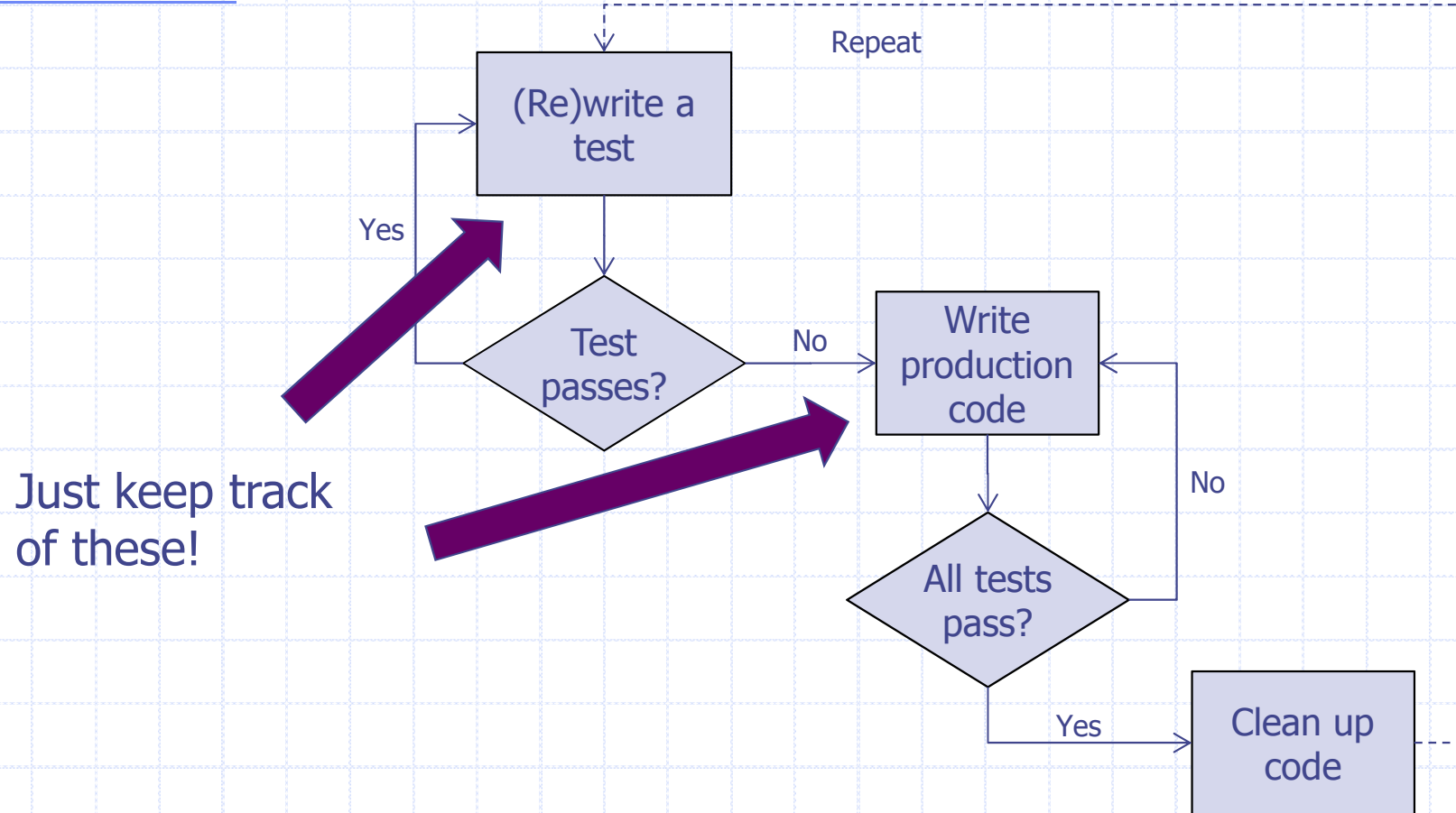


Collective code ownership

Agile Meets SW Standards, and we all Win

- *TIR 45 comes at a much needed time*
- *TIR 45 stitches together the important high-level concepts*
- *TIR 45 outlines key practices needed for flexibility and quality*
- **Implementation issues are not ignored**
- *This TIR is actually just a starting point*

TDD: Traceability comes naturally



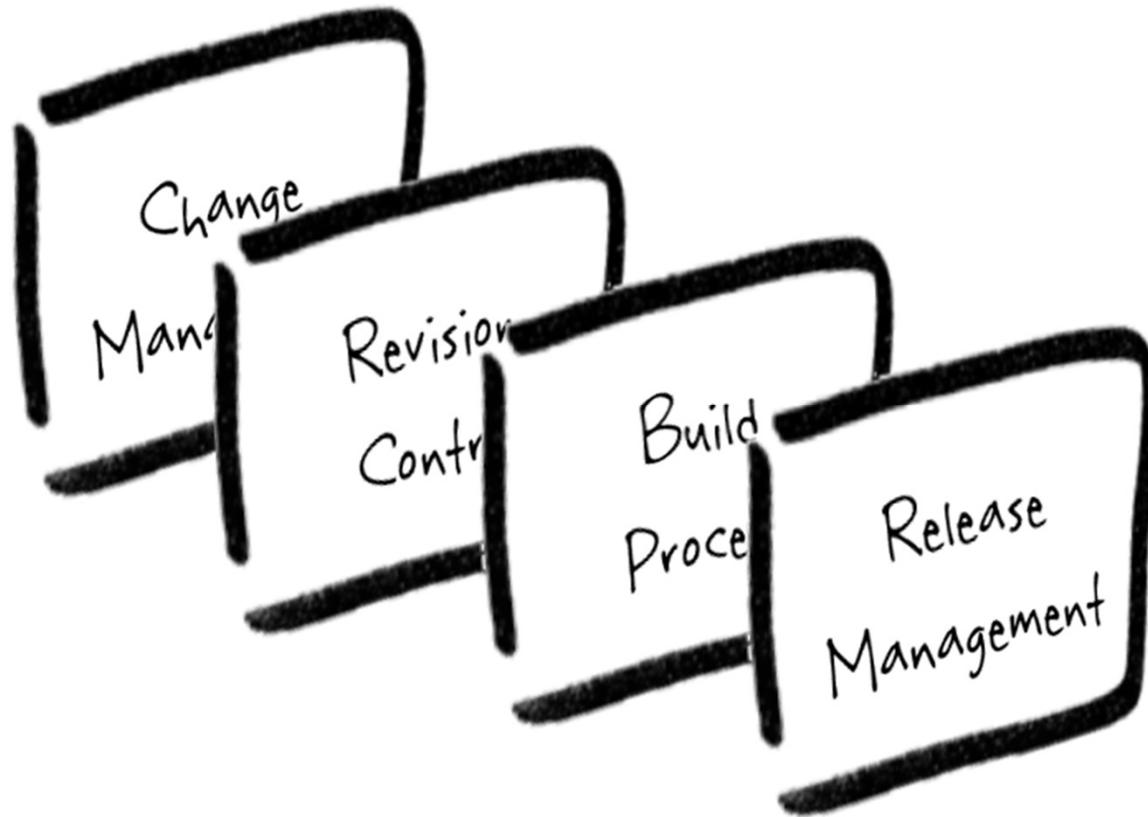
Source: http://en.wikipedia.org/wiki/File:Test-driven_development.PNG

"Release" Often?

A "release" can be:

- **Internal:** intended for a group outside the software development team but within the company
- **Demo:** For limited use outside company - only specific customers, not for therapeutic or diagnostic use
- **Final Evaluation:** Fully developed and tested, prior to regulatory approval, e.g. for Pivotal Trials and/ or Summative Usability testing
- **Approved or Cleared:** commercially available for clinical use.

Control is crucial



Applies to OTS software as well as to code under development!

Agile Meets SW Standards, and we all Win

- *TIR 45 comes at a much needed time*
- *TIR 45 stitches together the important high-level concepts*
- *TIR 45 outlines key practices needed for flexibility and quality*
- *Implementation issues are not ignored*
- **This TIR is actually just a starting point**

Doubt remains, interest grows

- Consider: *"I talk about Agile development in my course and generally advise against it for anything but Class I devices. ... In my experience, there's too much hype and not enough hard data supporting results."*
- Meanwhile, try a Google search on "Agile medical device development"



Martin Bakal

Agile Development in the Medical Device Industry



Mike Dobbles

Can and should agile be used for medical device development? Absolutely!

The Down and Dirty on Scrum in Medical Device Development

By [Bachan Anand](#) / Filed under [Agile](#), [Collaboration](#), [Inspect & Adapt](#), [Scrum](#), [Self Organization](#) / March 23rd, 2011



+1 0



Tweet 13



Share 19



Share



Like 4

By [Bachan Anand](#) and [Tricia Rodewald](#). First published in [March 2011 issue of the MPO magazine](#)

Feature

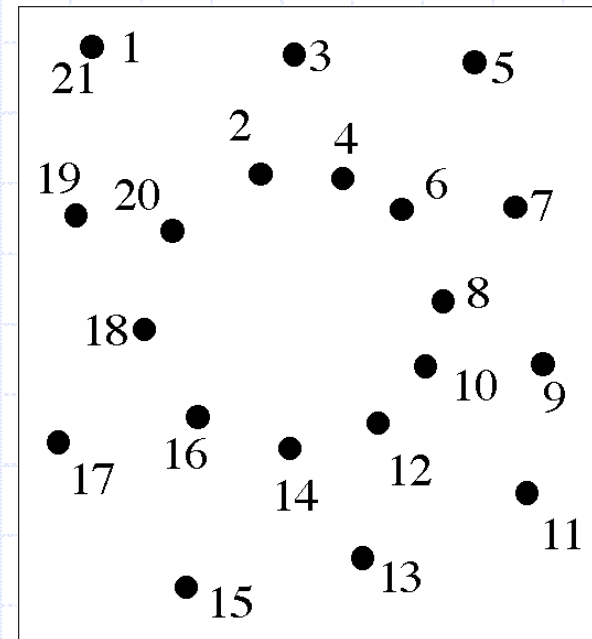
Agile Software Development Streamlines FDA-Regulated Applications

April 14, 2011

By: [Dan Olivier](#), [Certified Compliance Solutions Inc.](#) and [Jeff Dere](#), [Life Technologies](#)

What do we need?

- Success stories
- Hard data – e.g. cost reduction with productivity gain
- Techniques – how to “connect the dots”



Thank you!

Brian Shoemaker, Ph.D.
Principal Consultant, ShoeBar Associates
199 Needham St, Dedham MA 02026 USA
+1 781-929-5927
bshoemaker@shoobarassoc.com
<http://www.shoobarassoc.com>

