



Agile is More Than Software

SDMD Europe

Nancy Van Schooenderwoert

Lean-Agile Partners

Brian Shoemaker

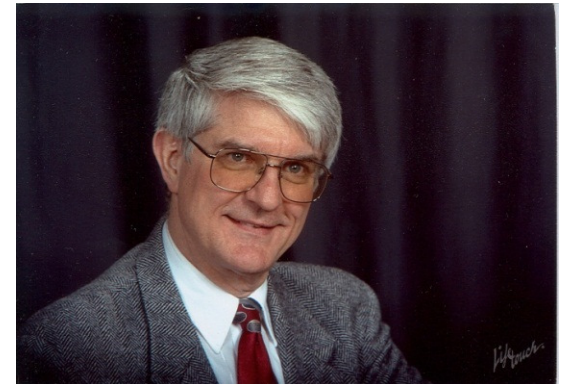
ShoeBar Associates

© 2009-16 Lean-Agile Partners Inc. and ShoeBar Associates
All rights reserved.

Who We Are

Brian Shoemaker

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



Nancy Van Schooenderwoert



- *Originally an electronics and software designer*
- *15 years safety-critical embedded systems development experience*
- *Since 2002: Agile coaching of teams and managers in regulated industries*
- *Industries: Aerospace (Flight simulation), Medical Devices, Sonar Weaponry, Scientific Instruments, Industrial Controls, Financial Services*
- *BSCE (Computer Engineering) from Rochester Institute of Technology*
- *Active in Agile New England & Agile Alliance; speaker at conferences worldwide*



Adopting “Agile” is incremental

- More like learning a language than like following a recipe
- Language *fluency* exists at 5 levels
 - Elementary proficiency
 - Limited working proficiency
 - Professional working proficiency
 - Full professional proficiency
 - Native or bilingual proficiency

Source: **Interagency Language Roundtable scale** of the United States Foreign Service Institute. https://en.wikipedia.org/wiki/ILR_scale



Start Off: Pick Your Interest

- A. Have your QA people accommodated the Agile approach in the software SOPs?
- B. Does your risk management track along with your iterative development?
- C. Do your hardware development people acknowledge and cooperate with your Agile approach?

Discuss with your group – report out issue identified and solution you think would work



“Slices” We’ll Discuss

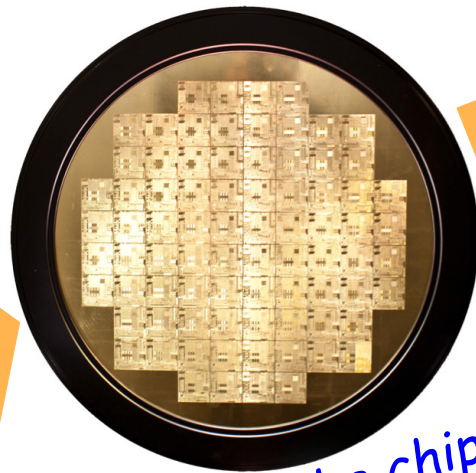
- Consider Agile for ***Hardware***
- Agile = quality improvement
- Risk management – iterative by nature
- Take-away concepts

Example: FPGA design

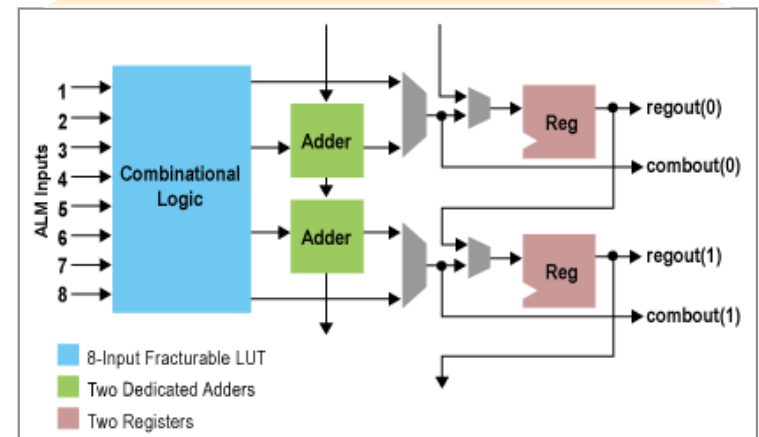
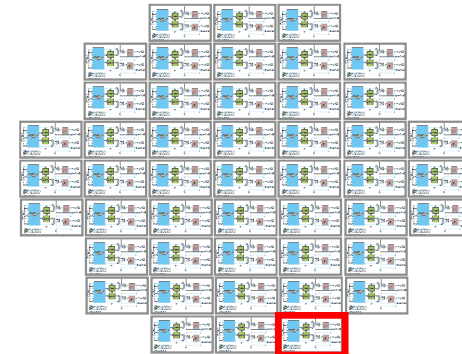
- Field Programmable Gate Array



FPGA chip on ckt board



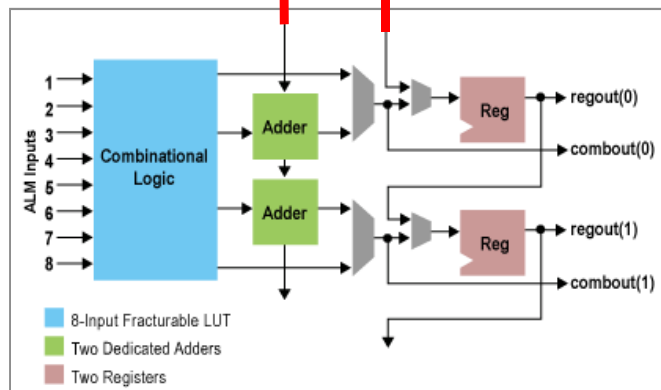
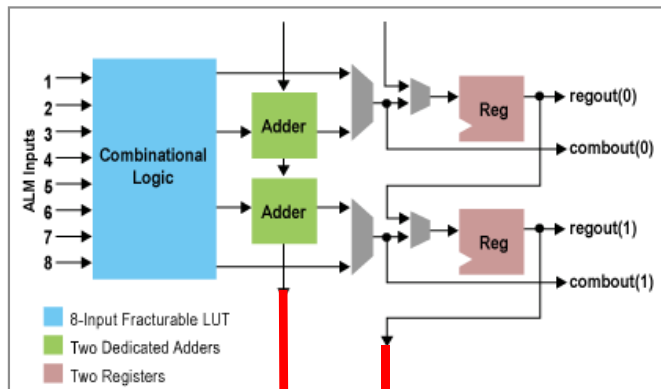
Inside the chip



Inside each circuit

FPGA circuits 'wired' by s/w

- Separate circuits on the wafer are connected by statements in HDL, a type of s/w



```
const logic [1:0] W_ENABLE = 1;  
const logic [1:0] R_ENABLE = 2;
```

```
// SETUP -> ENABLE  
always @(negedge rst_n or posedge clk) begin  
    if (rst_n == 0) begin  
        apb_st <= 0;  
        prdata <= 0;  
    end
```

```
else begin  
    case (apb_st)  
        SETUP : begin  
            // clear the prdata  
            prdata <= 0;
```

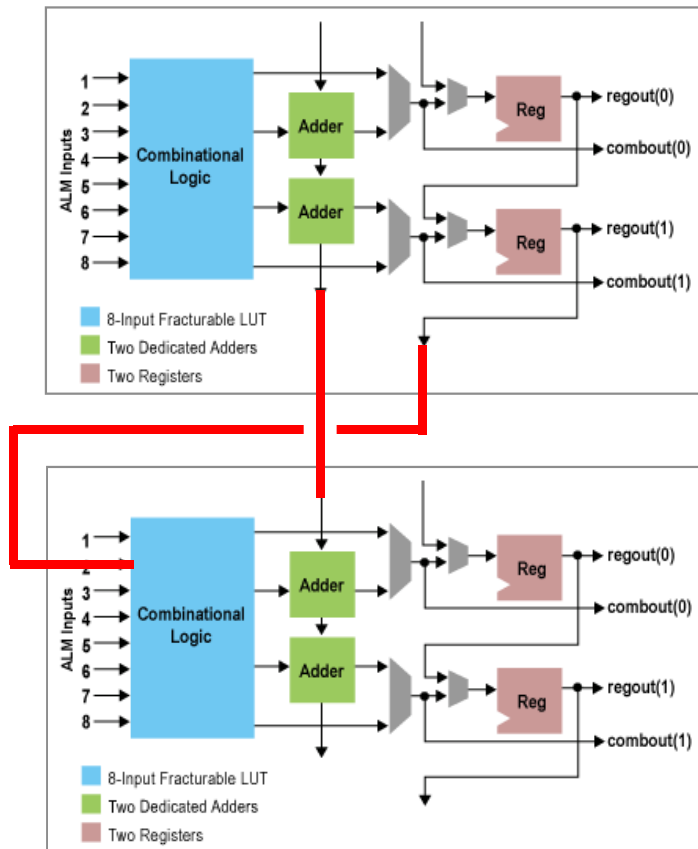


Hardware Description Language (HDL)



FPGA circuits 'wired' by s/w

- Changes to the FPGA can take a few minutes to a few hours



- FPGA Reprogrammed - new *bitfile* downloaded in seconds...
- Bitfile is rebuilt: this *changes* the circuit connections, takes time
- Result: h/w can do the work faster than s/w – by orders of magnitude

HDL testing...

Before Agile

- Write the HDL code
 - (optional) simulate the HDL
- Deploy to the FPGA circuitry
- Check the circuitry behavior manually
- *Very time-consuming*

After Agile (TDD)

- Write the HDL code
 - Write a unit test in HDL
 - Ensure unit test fails
 - Write HDL code
 - Ensure unit test passes
- Deploy to the FPGA circuitry
- Check the circuitry behavior manually
- *Much faster*

SVUnit →

Result: have code

Result: have code and tests

Using SVUnit TDD framework



What can we conclude?

- What are the safety implications of hardware built using TDD?
- Is TDD cost-effective?

Example: Auto Design

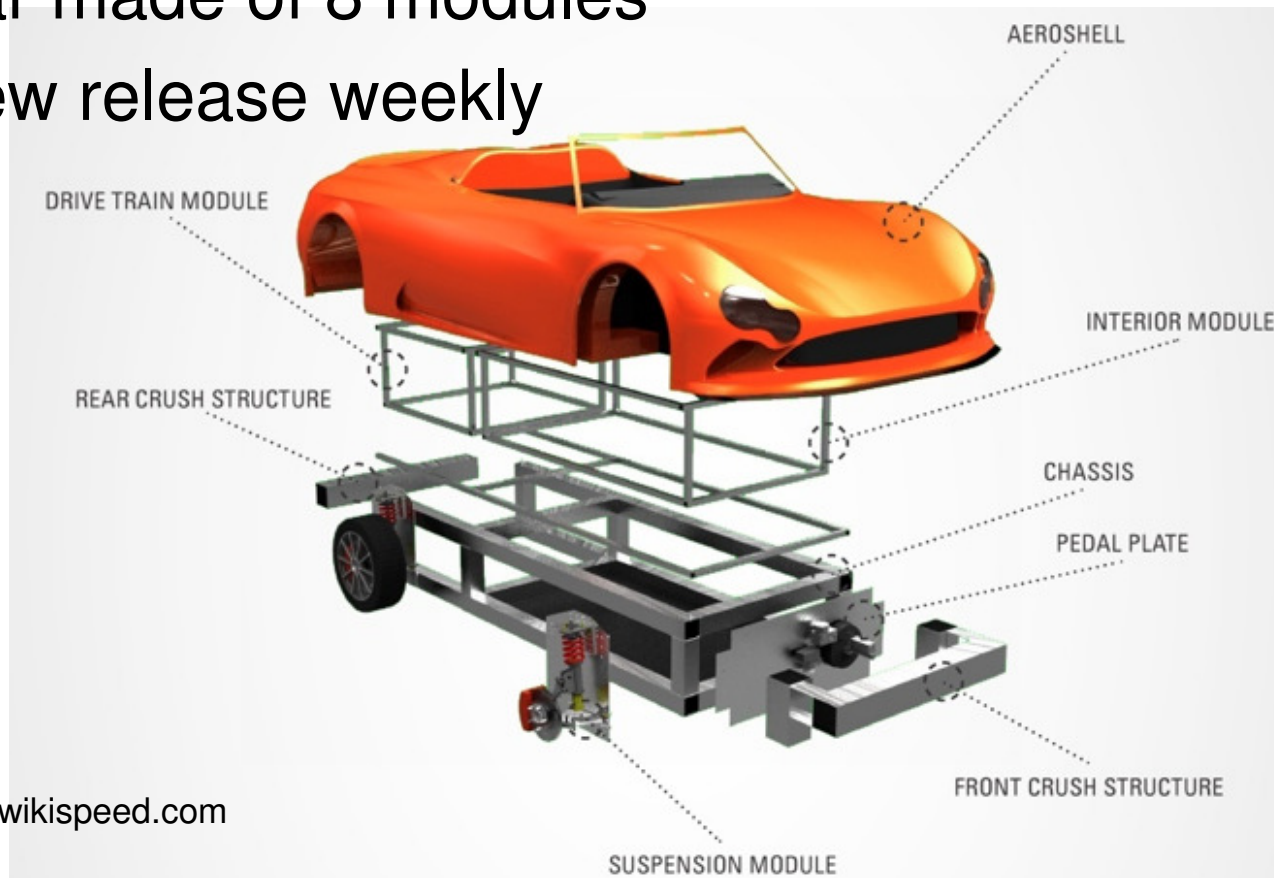
- Wikispeed 100 mpg car built by volunteers
- Agile 'software' practices used:
 - Pairing
 - Swarming
 - TDD (Test-driven Development)



Source: www.wikispeed.com

Modularity is key

- Modularity → dependency management
- Car made of 8 modules
- New release weekly



Source: www.wikispeed.com



What can we conclude?

- What advantages are gained by having a *working product* at regular short intervals?
- Why is a modular design important?

Example: Grain Monitor System

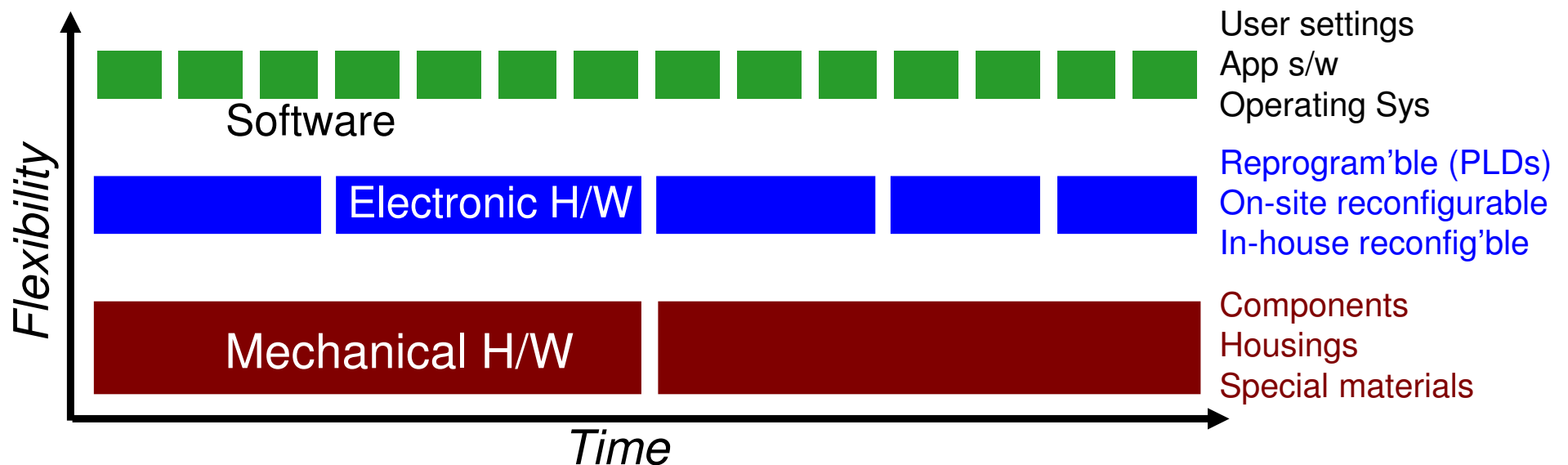


Spectrometer system

- Measures protein, oil in corn, wheat, etc. in seconds, in the field
- Evaluated for **medical** application
- New science, new CPU, new OS port, new NIR sensor, new algorithm...
- Agile team delivered 1st field units in 6 months
- In 3 years – 60+ s/w iterations,
 - approx. 9 electronic iterations
 - approx. 5 mechanical iterations
 - 51 s/w defects post-unit-tests, 3 yr total

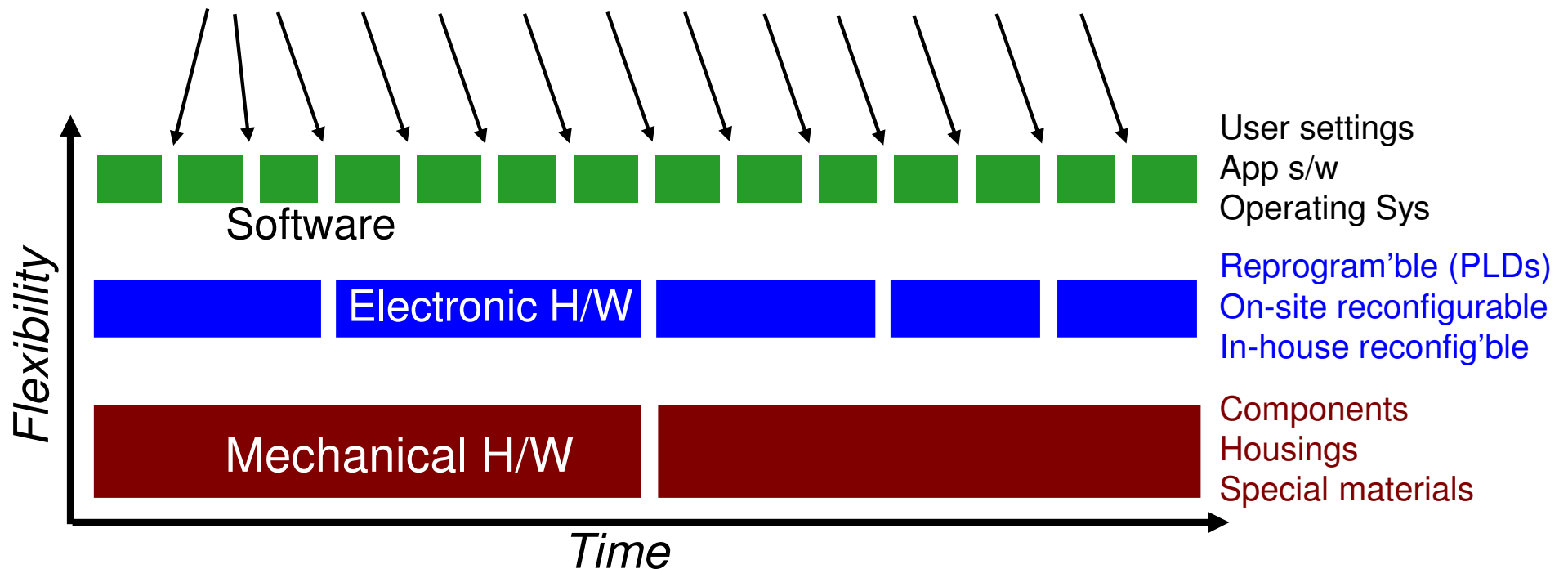
Iterations work differently

- Less frequent iterations for hard-to-change items
- Aim for *working hardware* at each iteration boundary
- Misconception: To be Agile, h/w dev has to fit inside of 2-wk or 4- wk iterations



Iterations work differently

- Each junction gives tangible baseline each person sees





GMS Agile s/w helped h/w

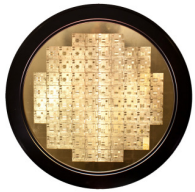
- Only the s/w team was using Agile practices, but...
- Frequent s/w releases created many more opportunities to improve h/w-s/w interaction
 - Some measurements inconclusive due to voltages out of range – so added **s/w monitoring** of h/w key areas
 - Field problems that could not be isolated to one area (opto, sensor, electronics) could be investigated thru special s/w releases for **troubleshooting**
 - Hand assembly of field units improved by downloadable collection of **s/w drivers** with command-line menu
- Result was h/w became more Agile “without trying”



What can we conclude?

- Must longer h/w iterations conflict with shorter s/w iterations?
- Is it important for product engineers to be co-located?

What do these examples tell us?



- FPGA example: Predictability



- Wikispeed Car example: Fast learning



- Most projects: Predictability for coordination, Fast learning to handle unknowns



“Slices” We’ll Discuss

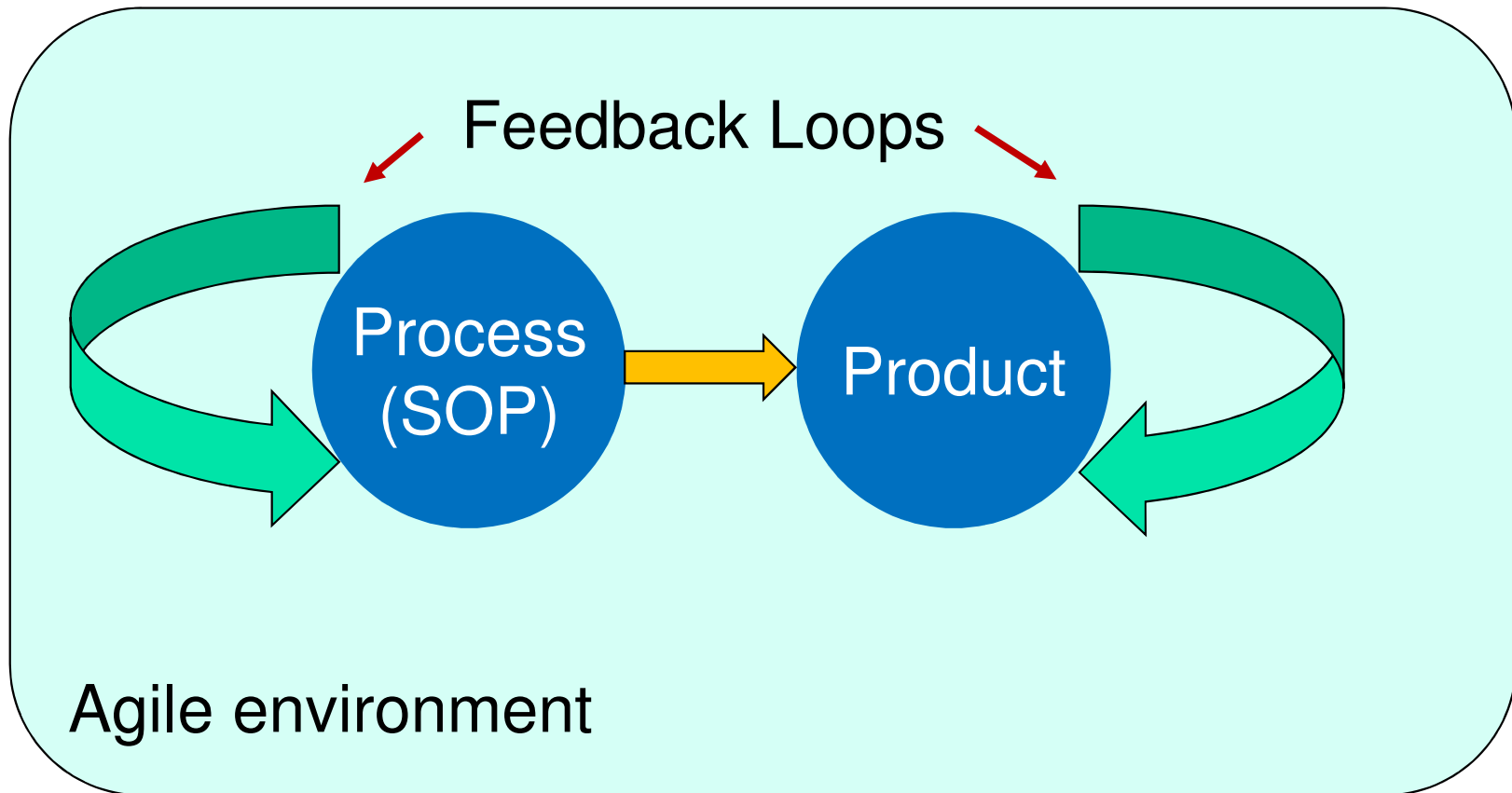
- Consider Agile for *Hardware*
- Agile = quality improvement
- Risk management – iterative by nature
- Take-away concepts



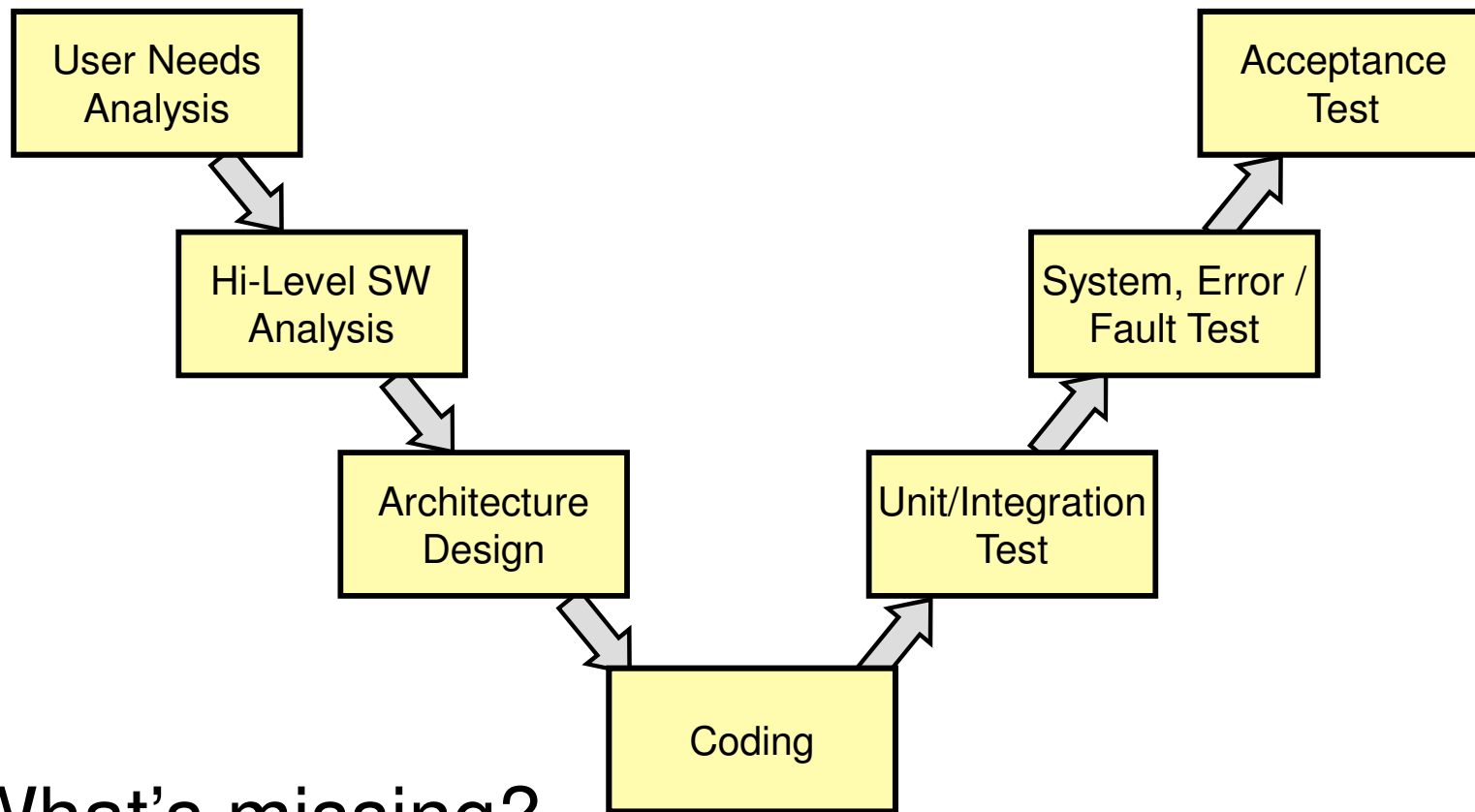
Agile = Quality Improvement

In Agile, SOPs ***add value***
because they are an integral
part of feedback loops

Agile = Quality Improvement



Are these steps linear?



- What's missing?
- How are these boxes related to each other?

Original “Waterfall”

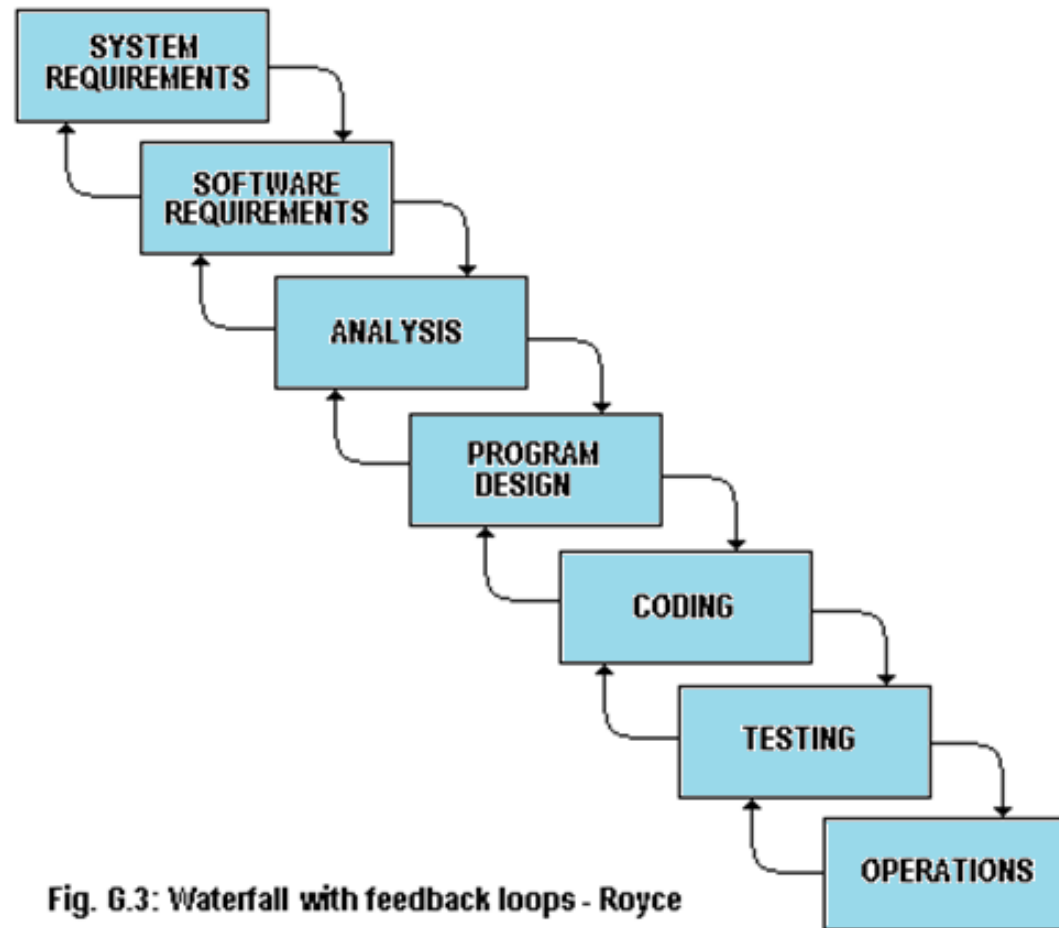
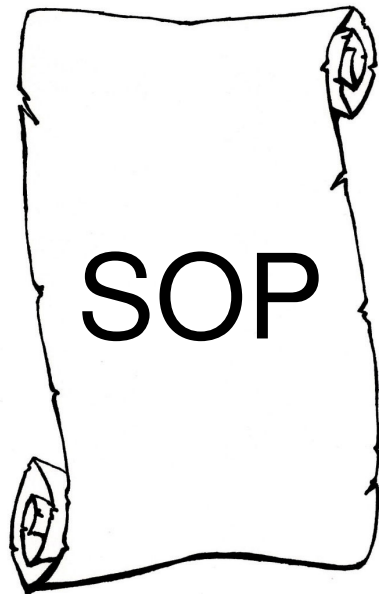


Fig. G.3: Waterfall with feedback loops - Royce

What do you think?



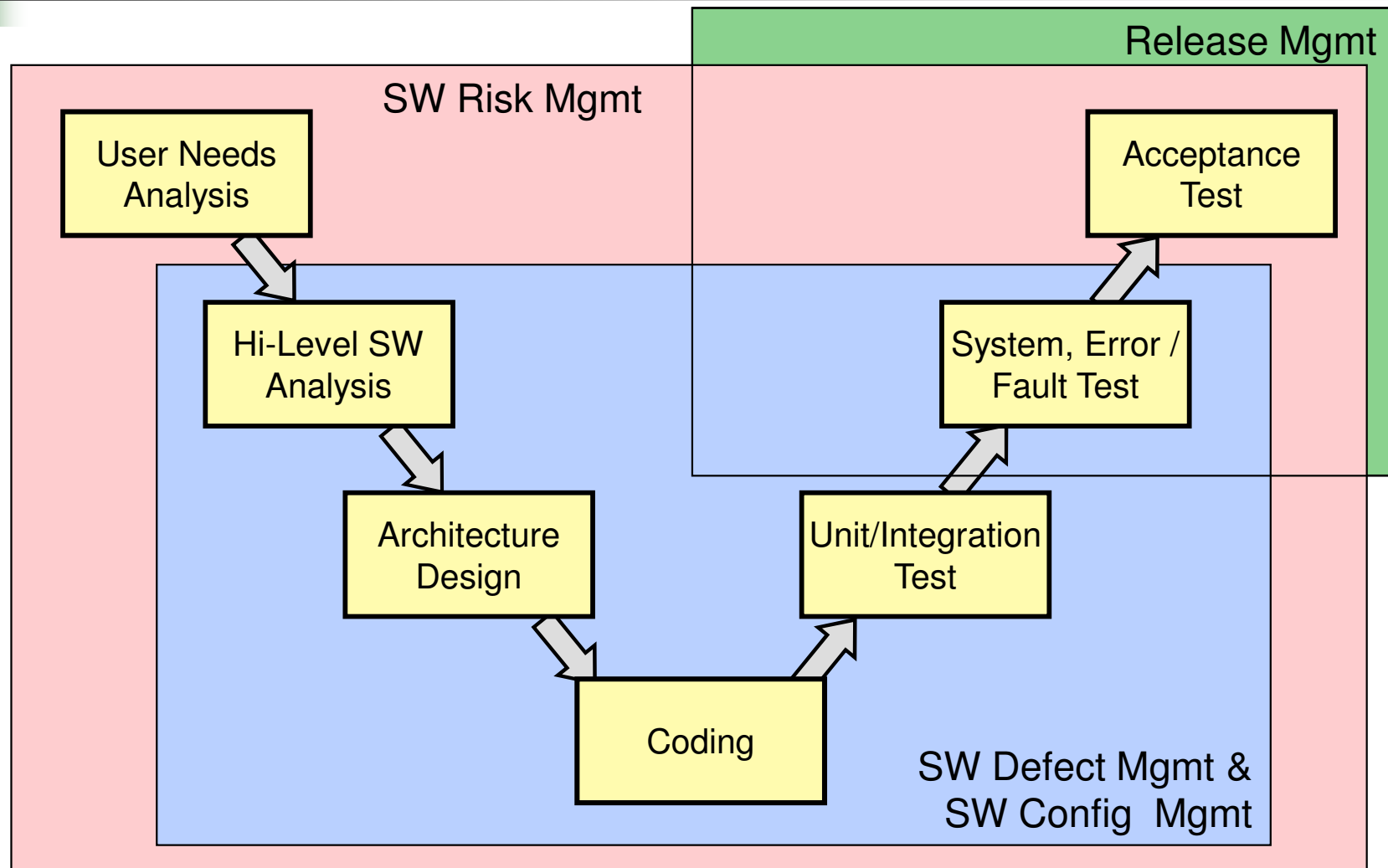
=



?

Are our procedures perfect and permanent?
Can we ever improve them?

SOPs: Support Processes



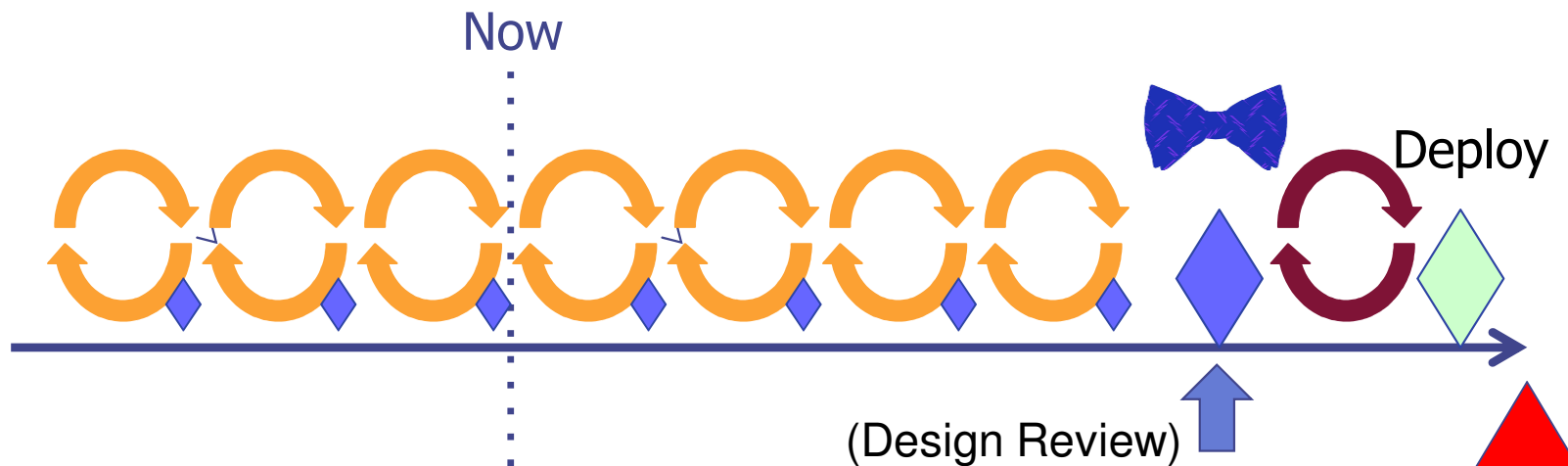


Processes?

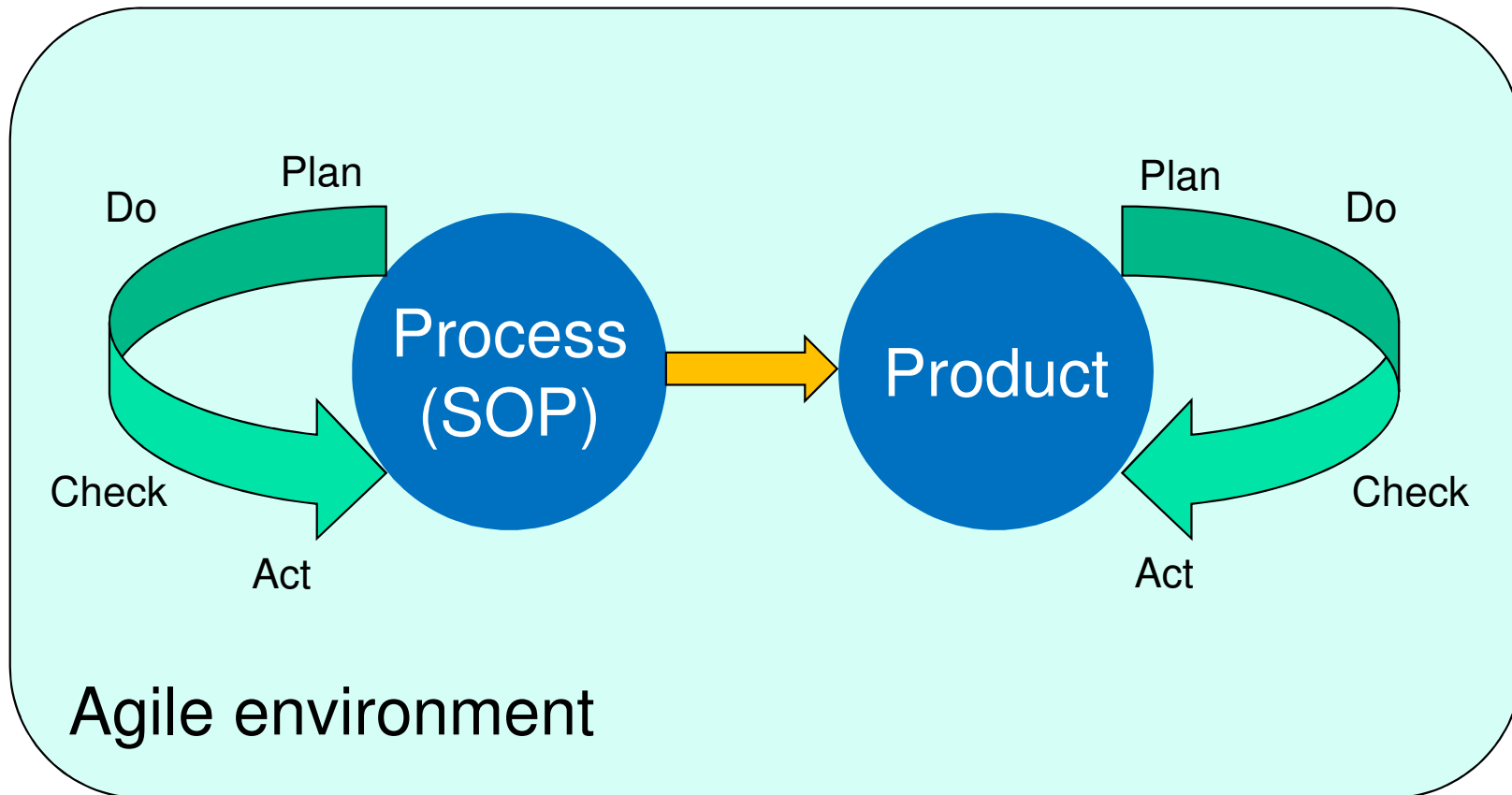
- Which processes should be supported by written procedures?
- What advantage do we get from standard procedures?
- Do SOPs themselves ever create a problem?
- When can a team provide feedback on their SOPs?

Continuous Learning

- Accept that we need to learn
- Each iteration has design, dev, test, demo (◆)
- Consider SOPs as well as the design
- Plan multiple iterations!



Where can feedback occur?

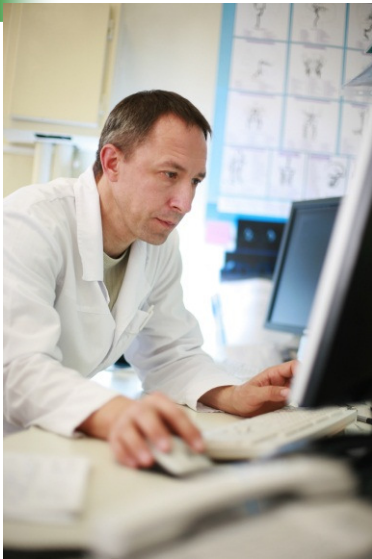




“Slices” We’ll Discuss

- Consider Agile for *Hardware*
- Agile = quality improvement
- Risk management – iterative by nature
- Take-away concepts

Who are you designing for?

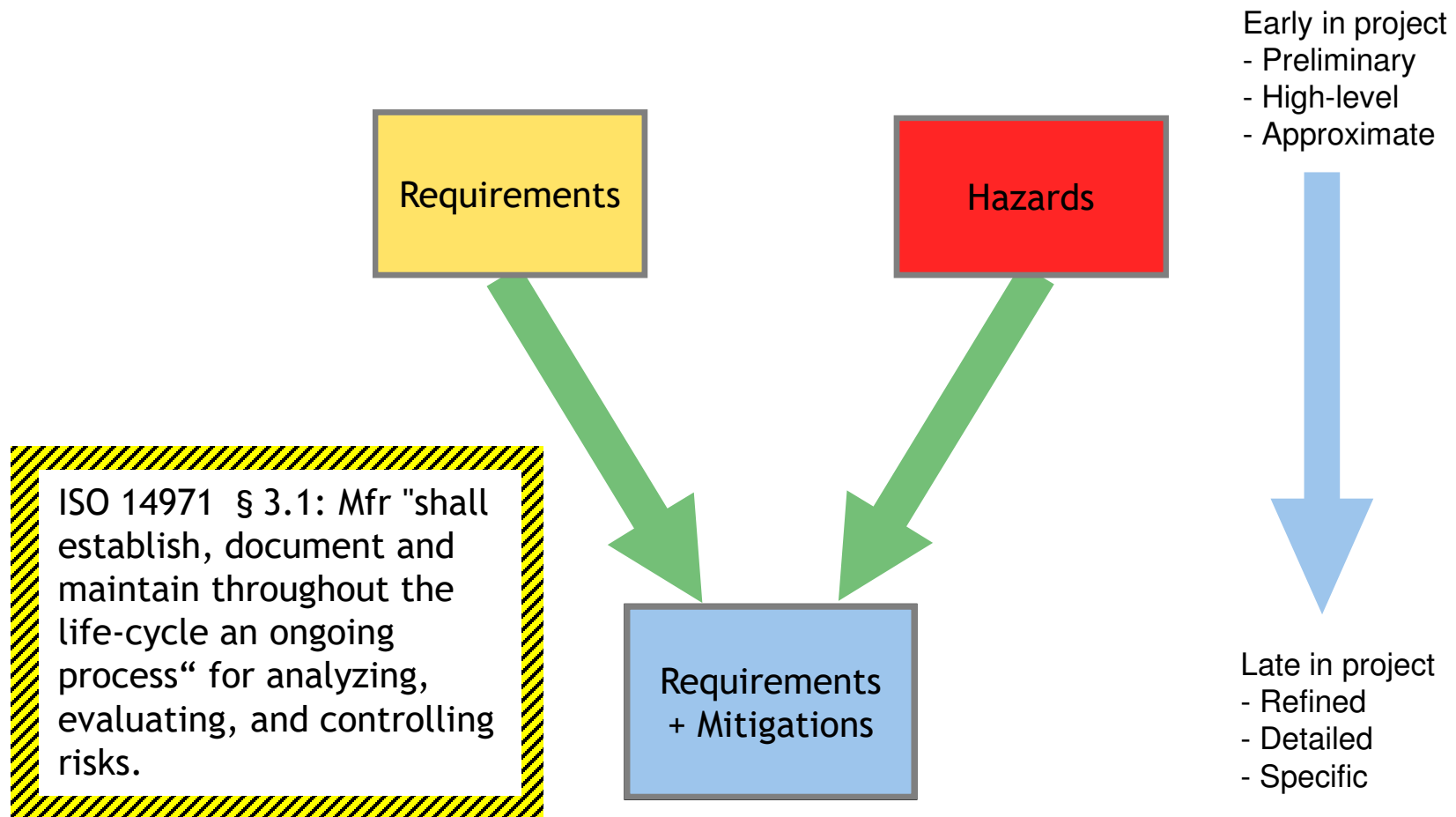


- Who will actually operate your system?
- Do you know what jobs they have to do every day? Where and under what conditions?



- **Could they misuse the system in a way that would hurt or kill the patient, the user, or a bystander?**

Risk Management MUST Iterate



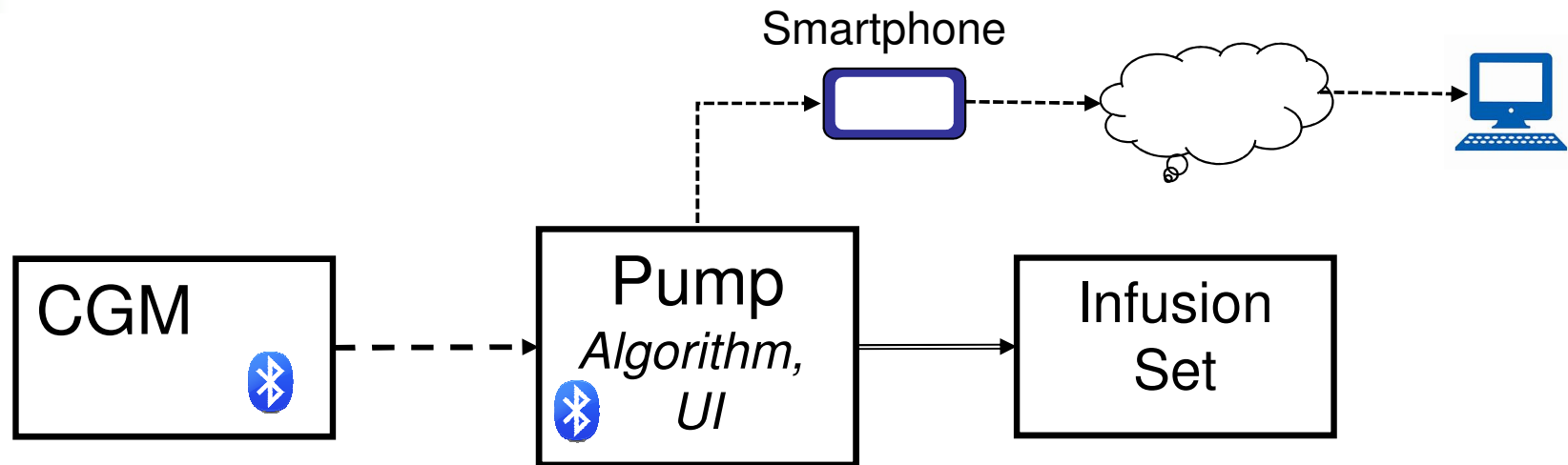
Consider an example



“Artificial Pancreas” – closed loop control of blood glucose for type I diabetics

Image: <http://www.wired.co.uk/news/archive/2012-06/11/artificial-pancreas>

Think about hazards



- Who should take part in the hazard analysis?
- What sorts of bad things could happen, i.e. what could go wrong?
- What could we learn from animal studies and trials?
- Will we have other risks to consider, on adding this feature?



Challenge: Exploding Tech



Integrated Health IT Networks

Image:
<http://chandra.harvard.edu/photo/category/snr.htm>

Interoperable devices

Home Care devices

Mobile medical devices

Software as a medical device

Ability to respond quickly is key – having a team that isn't in constant firefighting mode, but which has the currently-implemented features well in hand.



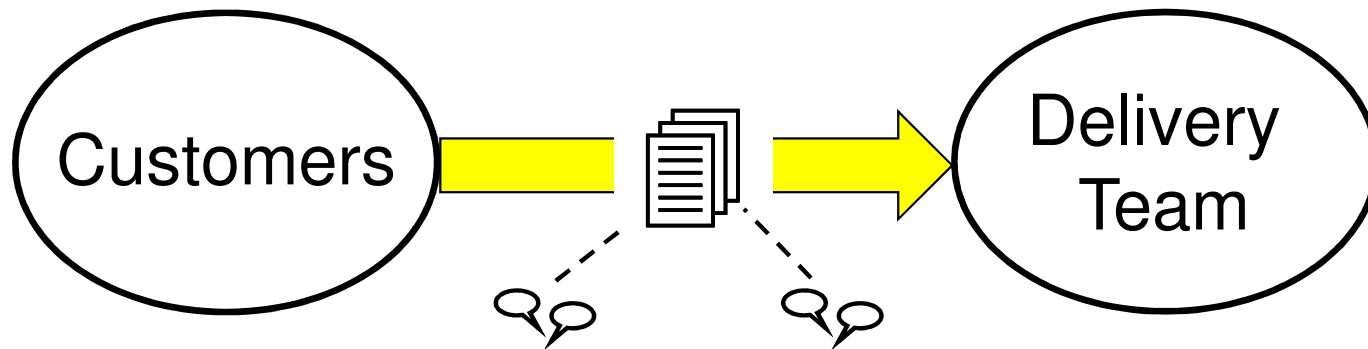
Consider Security As Well

Recent industry news:

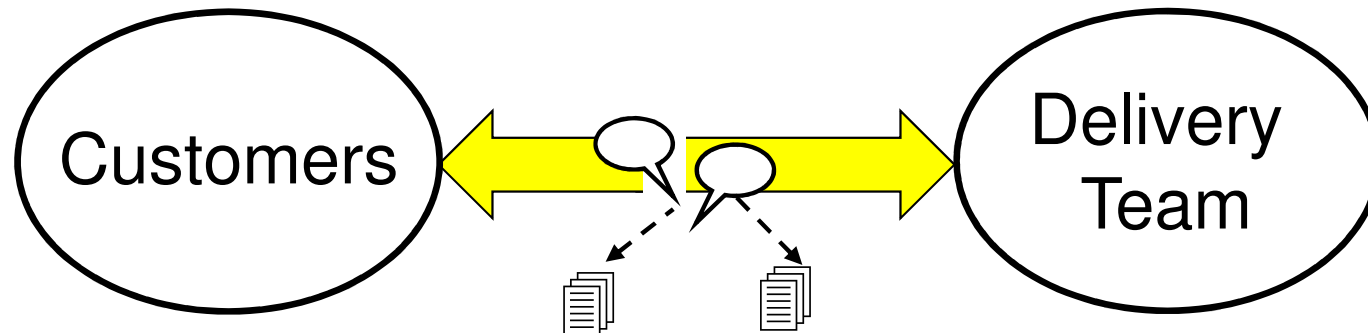
- FDA issues cybersecurity recommendations for device manufacturers
- Mach7 granted U.S. patent for medical image capture via mobile device
- LifeWatch mobile cardiac telemetry patch cleared
- Philips / Validic to integrate consumer-generated health data from 3rd party apps, devices
- Boston Scientific, Accenture develop data-driven digital health solution

RM Requires Conversations

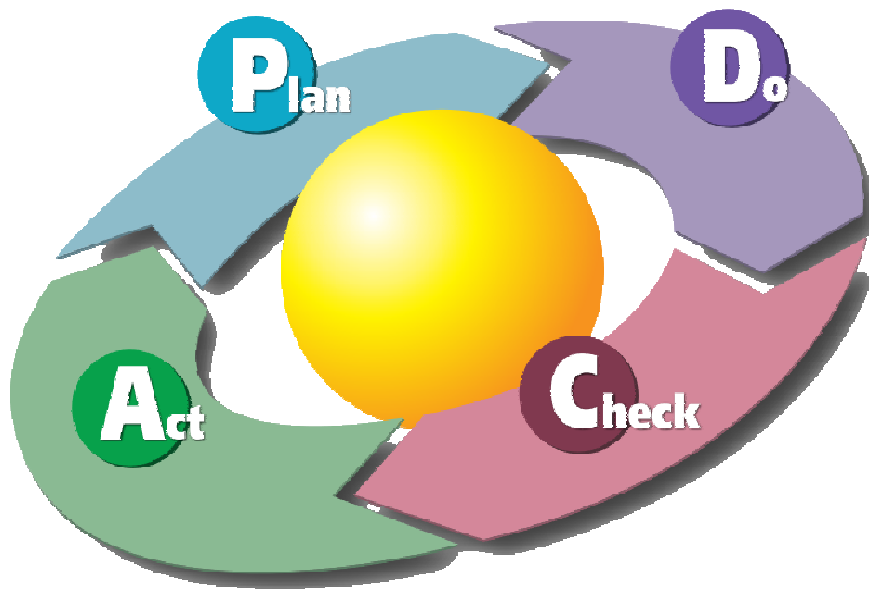
- From **Document-centric**, supported by Conversation



- To **Conversation-centric**, supported by documents



Work iteratively!



- Good risk management is the same as ever – Agile hasn't changed that
- Early analysis is not static – review & revise as iterations proceed

Image: <http://www.bulsuk.com/2009/02/taking-first-step-with-pdca.html>



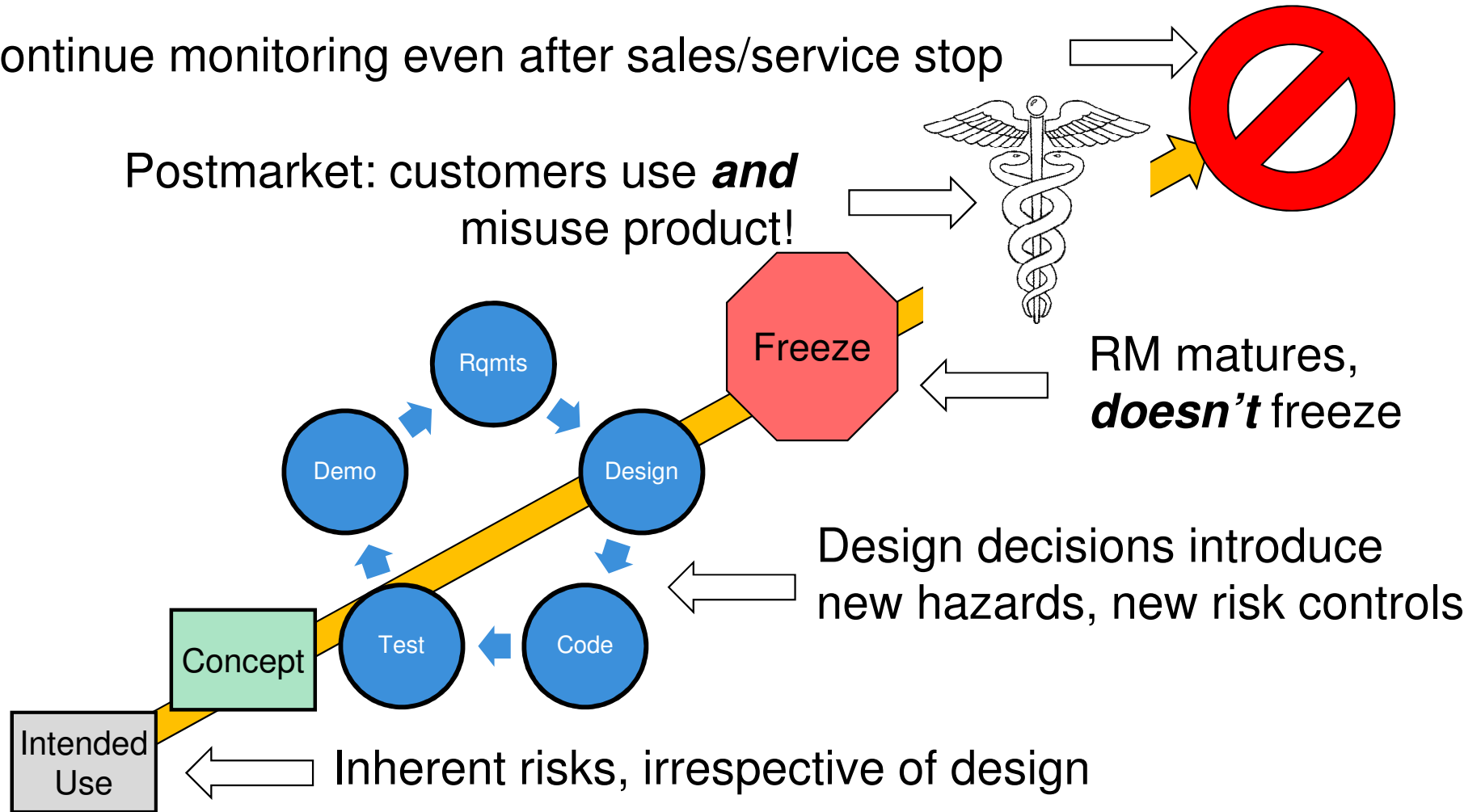
Is this process ever “complete”?

- Do we know enough about hazards when a project begins?
- Will we learn as potential users try out our design?
- What other analyses can we do when we have a detailed design?
- Might we bring in other stakeholders later in development?

RM In the Product Lifecycle

Continue monitoring even after sales/service stop

Postmarket: customers use *and* misuse product!





“Slices” We’ll Discuss

- Consider Agile for *Hardware*
- Agile = quality improvement
- Risk management – iterative by nature
- Take-away concepts



Key Thoughts

- The “moment of truth” comes early and often - not just at end of project!
- “Zero Defects” is already a practical achievement for Agile software
- Different kinds of work can “flow” and seek its own level much like a river: Controlling WIP is key!
- The mindset affects MANY other areas of a company besides software; others we haven't discussed include
 - Marketing
 - Budgeting
 - Project planning
 - Human resources
 - Management

Agile Trends

Trends



Source: Stefan Rock, slideshare "Agile Trends", used with permission. Pub.2015 by <http://www.slideshare.net/roock/agile-trends-46357024>



Apply what we've discussed

Gather in same groups

- **Hardware**: Can SW add something to help HW group accelerate progress? - OR - Can HW work to produce demonstrable HW more regularly?
- **Hazard**: How to make hazard analysis lighter weight so it can be done incrementally?
- **SOPs** - Draft a high-level description of the SW development process: avoid "linear" mindset, but address customer needs, quality goals, and necessary documentation
- Report out: **What will you bring back to improve your "fluency"?**



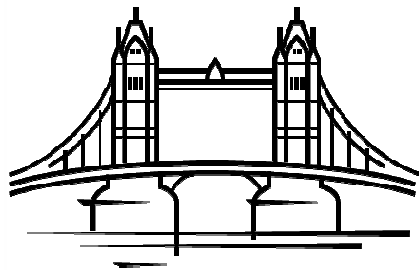
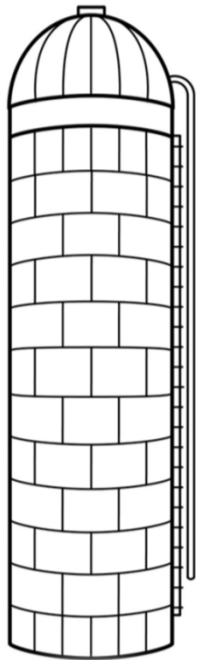
Agile fluency

- “Language” fluency for your Agile journey
 - Elementary proficiency: some practices, s/w only
 - Limited working proficiency: more consistent, s/w only
 - Professional working proficiency: apply to h/w, QA, RA
 - Full professional proficiency: apply to some mgmt areas
 - Native or bilingual proficiency: whole company Agile

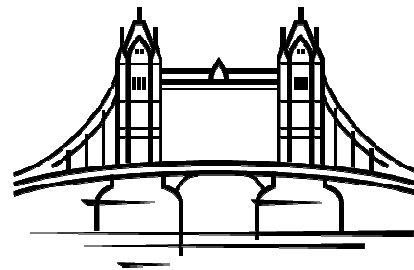
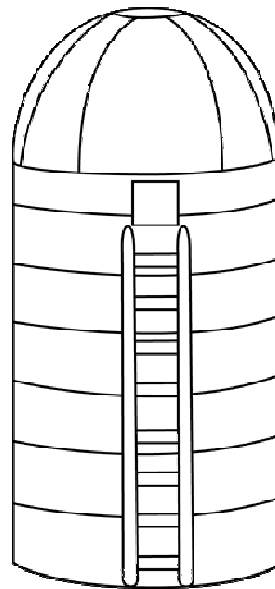
Source: **Interagency Language Roundtable scale** of the United States Foreign Service Institute. https://en.wikipedia.org/wiki/ILR_scale

Effective Agile – Bridge Silos!

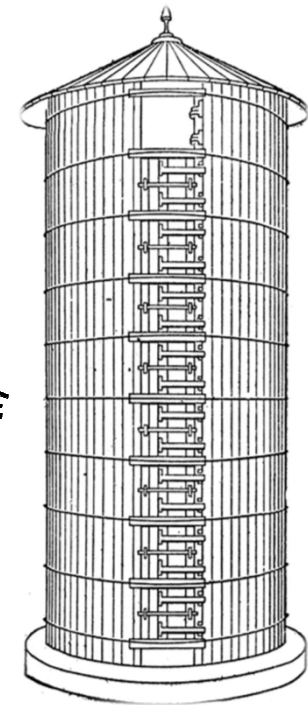
R&D / Engineering



Clinical / Support



Marketing



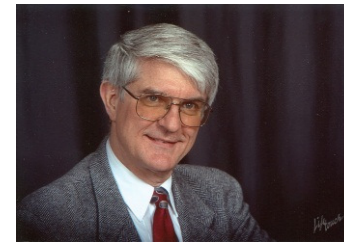


Contact Us



Nancy Van Schoenderwoert
LeanAgilePartners
nancyv@leanagilepartners.com

@vanschoo



Brian Shoemaker
ShoeBar Associates
bshoemaker@shoobarassoc.com

@brian_shoe



Our Services



Nancy Van Schooenderwoert

Lean-Agile Partners, Inc.
162 Marrett Rd., Lexington, MA 02421
+1 781-860-0212
NancyV@leanagilepartners.com
<http://www.leanagilepartners.com>

- Lean-Agile coaching for software and hardware teams
- Safety-critical, regulated coaching is our specialty
- Lean-Agile coaching for stakeholders and senior managers



Brian Shoemaker, Ph.D.

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

- Software quality systems
- Software validation consulting
- Software documentation
- Software quality auditing
- Electronic records & signatures consulting



References

- Neil Johnson, *TDD And A New Paradigm For Hardware Verification*, Proceedings of the Agile 2012 Conference, 2012
- TDD framework by Neil Johnson: SVUnit framework, <http://www.agilesoc.com/open-source-projects/svunit/>
- Wikispeed Project: www.wikispeed.com
- Royce, Winston W., “Managing the development of large software systems: Concepts and techniques,” in: Proceedings, IEEE WESCON (August 1970). (original “Waterfall” diagram)
- ISO 14971:2007 (2nd ed) Medical devices – Application of risk management to medical devices
- ANSI/AAMI/IEC TIR80002-1:2009, Medical device software - Part 1: Guidance on the application of ISO 14971 to medical device software, 26-Oct-2009.
- FDA Announcement, draft cybersecurity guidance:
<http://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm481968.htm>
- Current security examples:
<http://www.mach7t.com/company/news/mach7-technologies-awarded-patent-mobile-medical-imaging/>
<http://www.prnewswire.com/news-releases/lifewatch-receives-fda-clearance-for-its-mobile-cardiac-telemetry-1-lead-patch-565609491.html>
<http://hitconsultant.net/2016/02/03/31719/>
<http://www.clinical-innovation.com/press-releases/boston-scientific-and-accenture-develop-data-driven-digital-health-solution-help-improve-patient-outcomes-and-reduce-cost-treating-chronic-cardiovascular-conditions>



Further Info

FURTHER INFO & Credits:

More examples of Agile hardware: N. Van Schooenderwoert, “Yes, Hardware Can Be Agile!”, InfoQ, March 2015, <http://www.infoq.com/articles/hardware-can-be-agile>

Discussion of Agile for predictability vs. Agile for fast learning; Jeff Patton, “Common Agile Isn’t for Startups”, <http://jpattonassociates.com/common-agile-isnt-for-startups/>

More on Extreme Manufacturing (XM); Peter Stevens, Blog *Extreme Manufacturing 10 Principles*, <http://www.scrum-breakfast.com/search/label/xm>

Experience report from GMS project: ‘Embedded Agile Project by the Numbers With Newbies’ paper by N. Van Schooenderwoert. Available no charge at <http://www.leanagilepartners.com/publications.html>

Version One 2014 survey. Ref <http://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf>