Lean-Agile Partners

Hardware-Software Collaboration in Agile Organizations

Nov. 2, 2017

Nancy Van Schooenderwoert

http://www.leanagilepartners.com/

NancyV@LeanAgilePartners.com

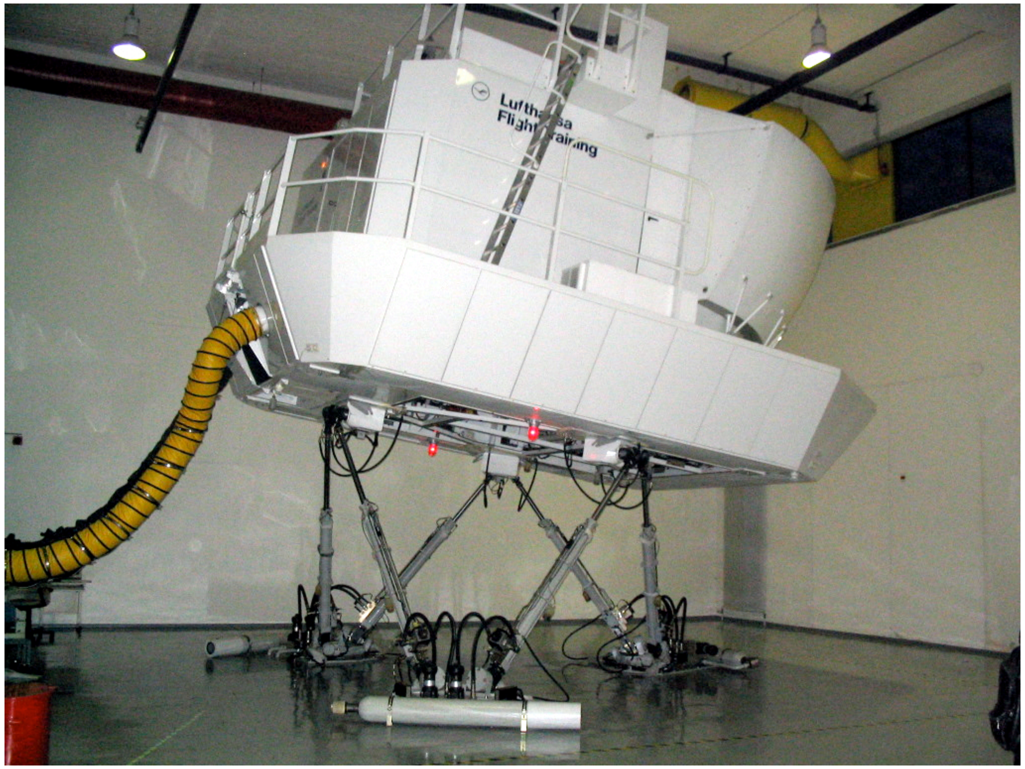
© 2017 Lean-Agile Partners Inc. All rights reserved.

Nancy V’s Background

- 15 years safety-critical systems experience
  - Electronic circuit design
  - Embedded software development
  - Requirements analyst & Team leader
- Pioneered Agile methods for embedded dev
- Coaching Agile teams & leaders since 1998
- Founder, Lean-Agile Partners Inc., 2006
- Industries: Aerospace – flight simulation, Medical Devices, Sonar Weaponry, Scientific Instruments, Industrial Controls, Financial Services
- Active in Agile Alliance and board member for Agile New England in Boston, USA

Lean-Agile Partners

Flight simulators – for Airlines, NASA



- Long before “Agile”…
- Singer-Link flight simulation
- Same group of electronic engineers designed flight instrument panels, and the software to drive them
- No separation between s/w and h/w

Flight simulator image courtesy wikimedia.org

Lean-Agile Partners

Questions

- Ask questions as they come to you
- We may "park" some to cover later

"For the things we have to learn before we can do, we learn by doing."

- Aristotle

Outline

- What prevents H/W – S/W collaboration?
  - Deeper specializations
  - Long H/W cycle times
  - Rework economics are opposite
- Benefits that surprised us
- Tips and Caveats

Lean-Agile Partners

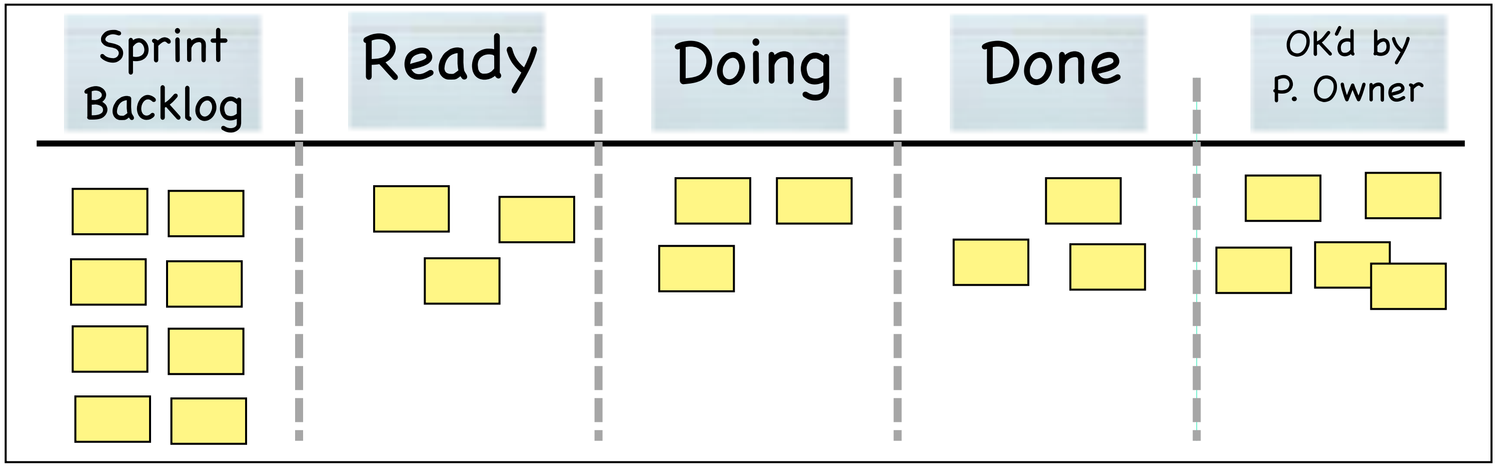
Issue: Deeper specializations

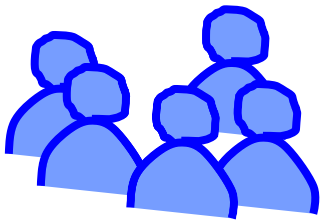
- Cross-skilling in Agile s/w teams
  - Is never 100%
  - Works for skills that overlap
    - development, architecture, test, etc.
  - “Story board” or a kanban board shows the shared work…

Lean-Agile Partners

Story board assumption…

- All team members share all work



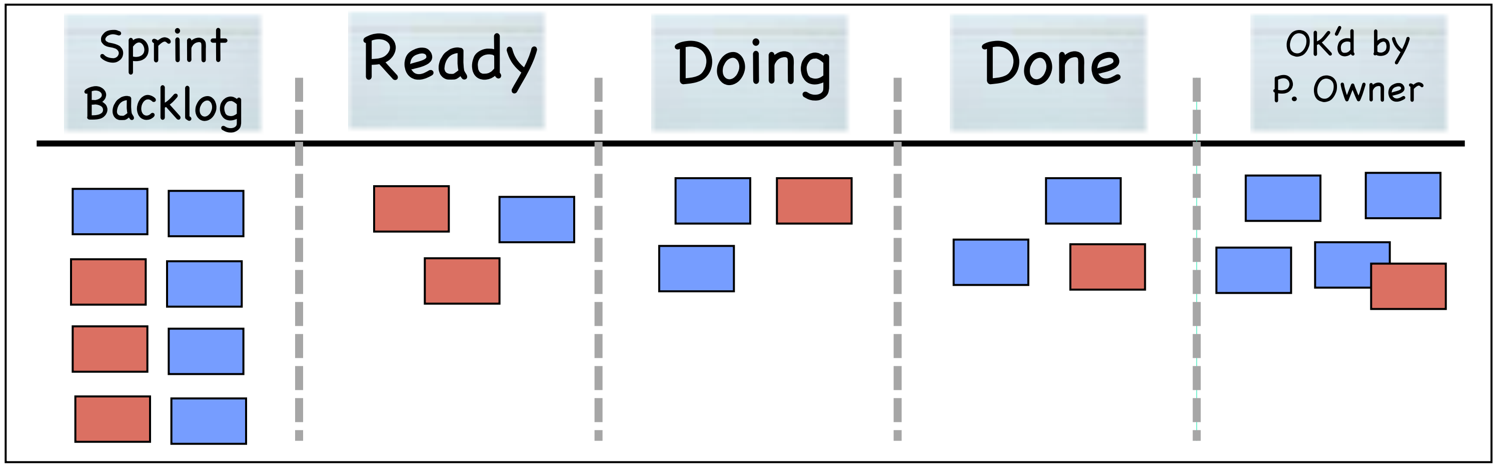


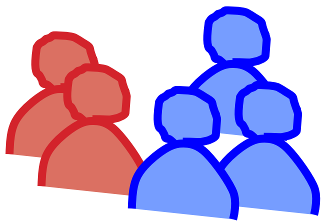
But what if that isn't true?

Lean-Agile Partners

Mixed team skills

- Skills are divided



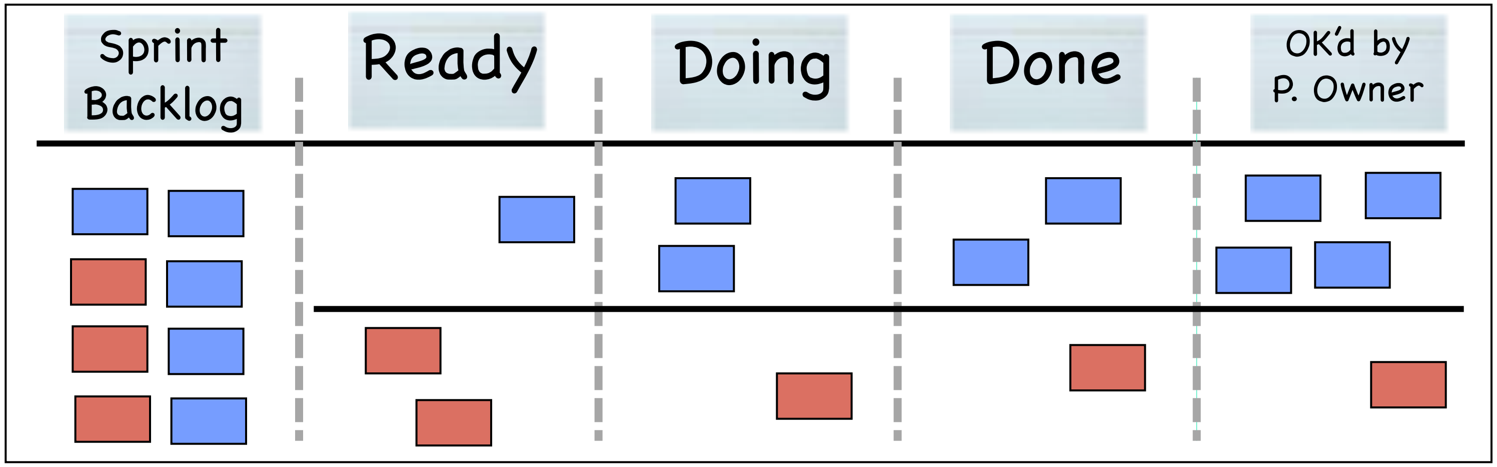


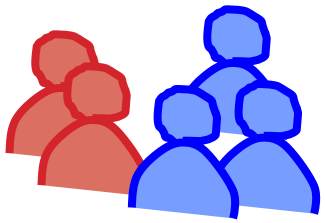
Members work tasks matching their skill, e.g. software + electronics

Lean-Agile Partners

Mixed team skills

- Story board is divided…





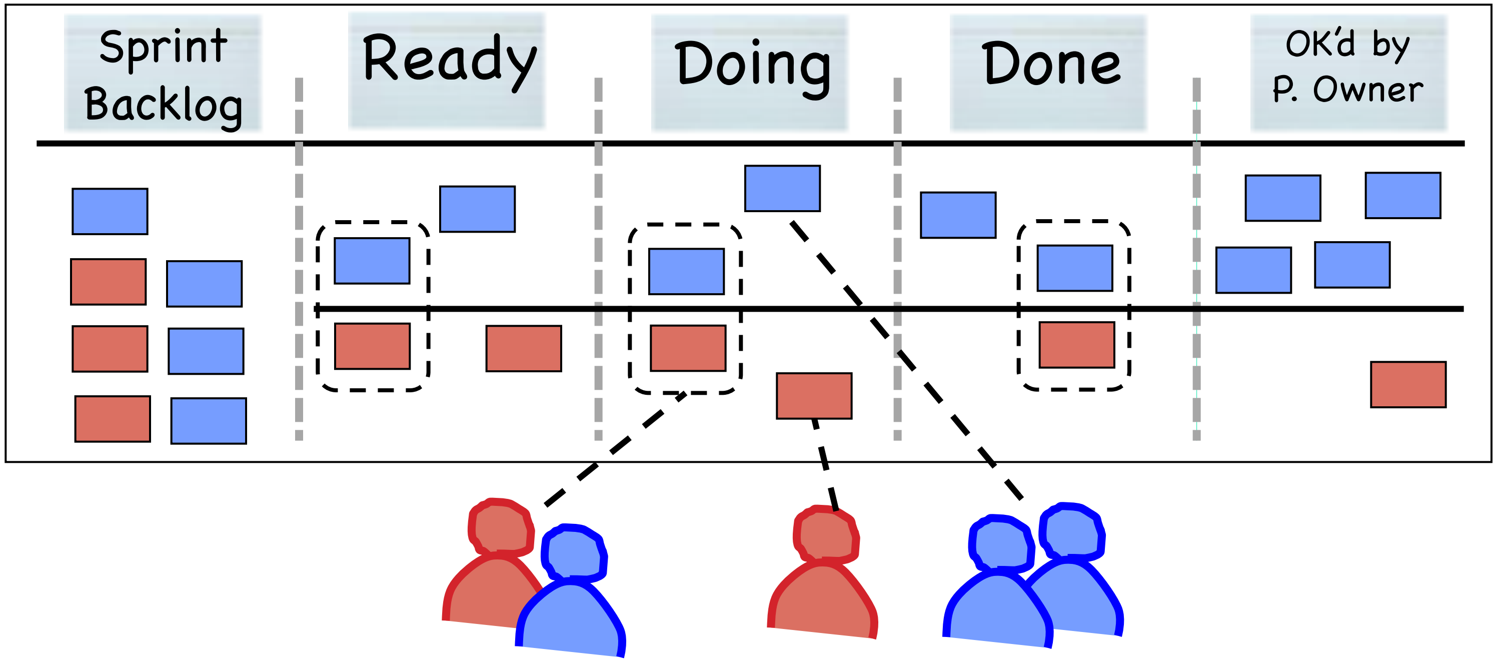
But we're still *one* team, now with different workstreams visible

A story board is simply one type of 'kanban' signal device

Lean-Agile Partners

Lanes not independent

- Keep focus on whole features; don't merely fit work to skill siloes



People pair to do their parts of features that span disciplines

Lean-Agile Partners

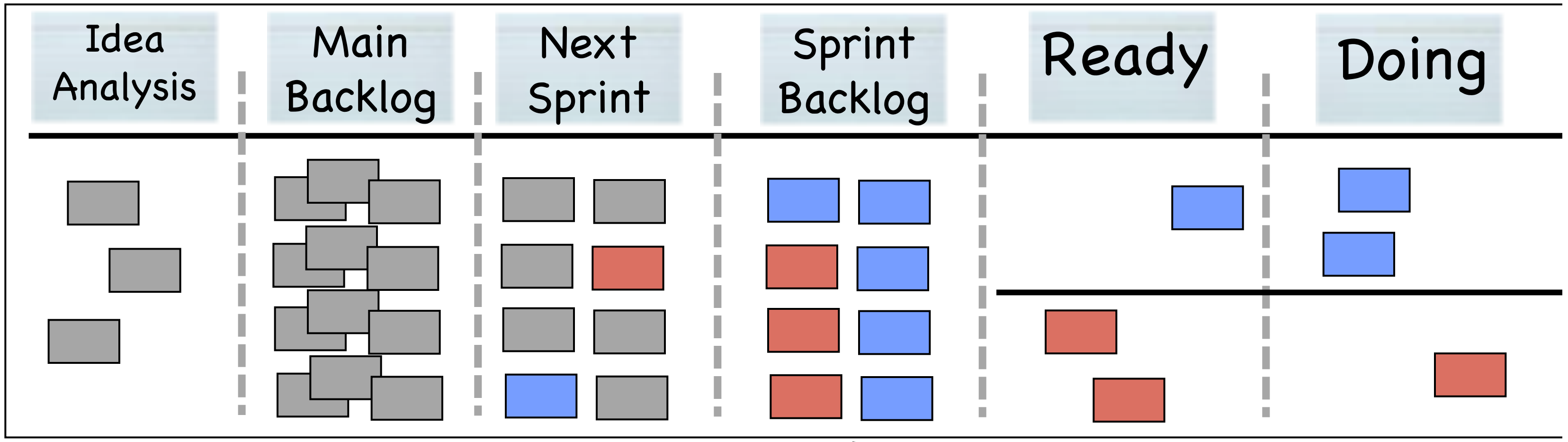
Even deeper specializations

- Example: team designing a sensor
  - Electronic engineer, Materials scientist, Mechanical engineer, Physicist
  - Don’t cross-skill. Communicate! Often!
  - Make a kanban lane for each person, and synch daily – “inspect and adapt”

Lean-Agile Partners

Upstream kanban board

- Architecture decisions steer features



At sprint planning, each story is assigned to a feature team

Team signals their desire to own a story in next sprint

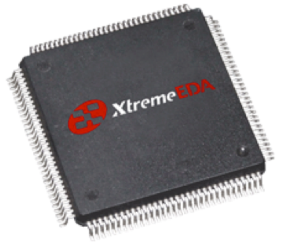
Tip: It's important to learn about & use WIP limits – without these, your kanban board will become just another messy "to do" list!

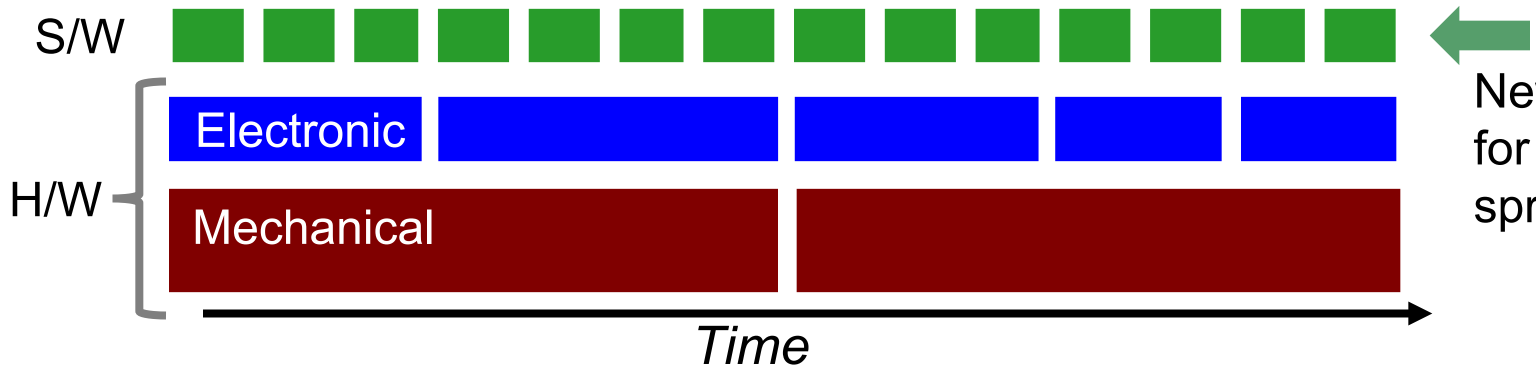
WIP = Work in progress

Lean-Agile Partners

Issue: Long h/w cycle times

- Many ways to mitigate
  - Simulation
  - Programmable devices – PLD, FPGA…
- Faster s/w cycles allow s/w team to support the electronics team, e.g. monitor test points

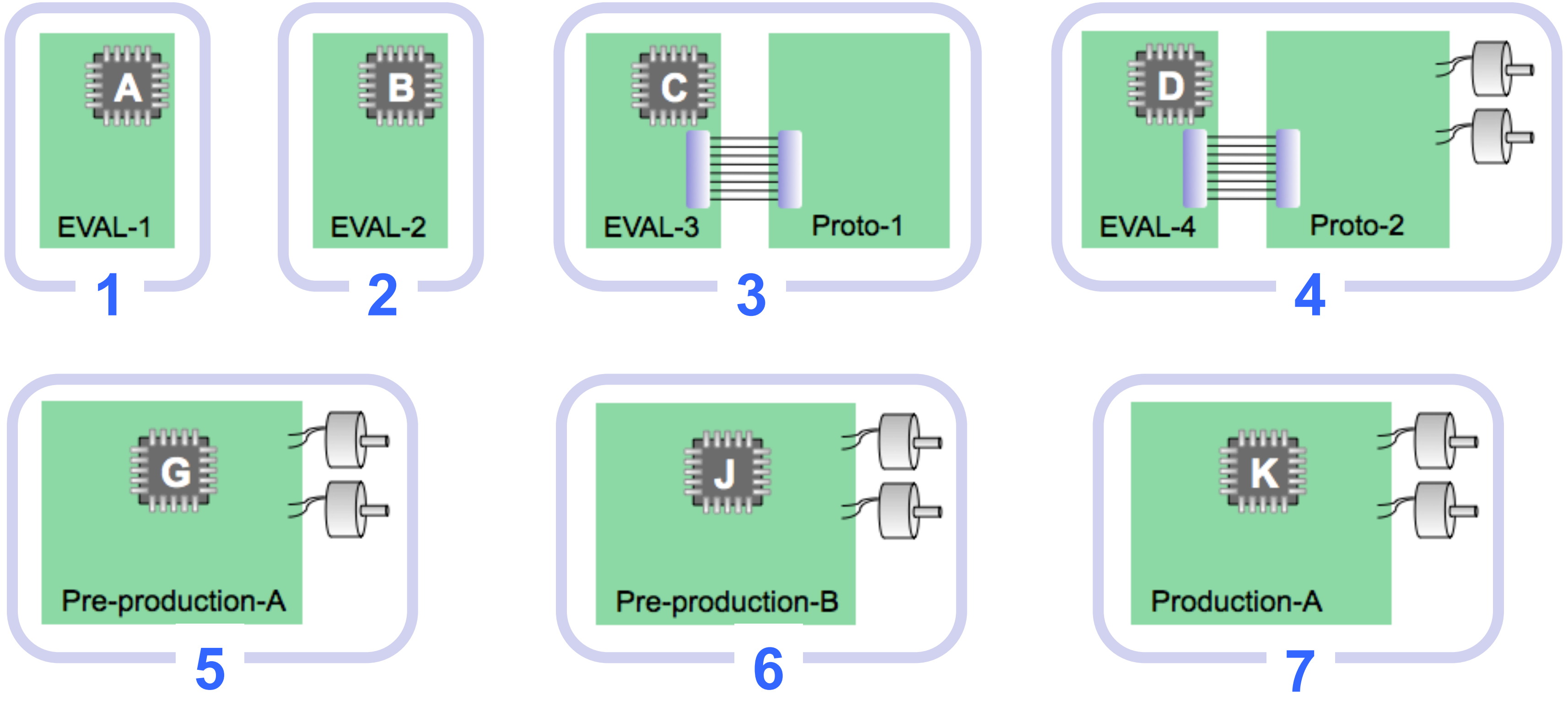




New learning for *all* at each sprint by any

Lean-Agile Partners

Hardware Evolving…



Hardware for a spectrometer instrument began with a manufacturer’s evaluation board, then added a hand-built “Prototype-A” board, etc.

Lean-Agile Partners

Making s/w ready to respond

- Software has to be ready to support not just its own growth but the evolving hardware
- Must be built to make troubleshooting easy!

Is it a s/w or h/w problem?

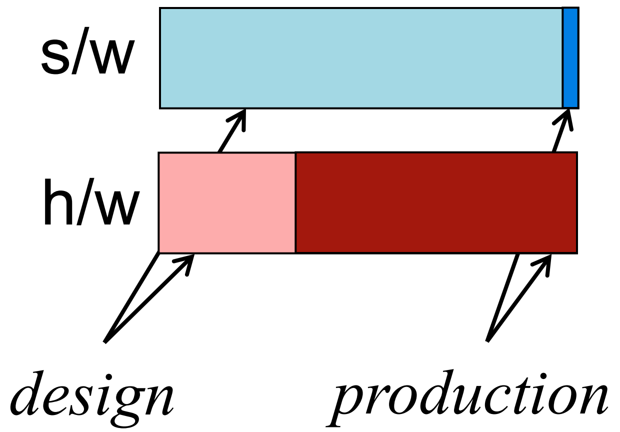
Spectrometer's embedded software defect prevention strategy

	Target CPU	Desktop PC
System	Full system able to run on target hardware. [The deliverable s/w]	Full system able to run on PC with hardware presence faked.
Domain	Example domain = the OS task that is the math algorithm. Can run alone.	A domain is not the whole system, but is more than a C function.
Unit	C function routine, or C++ class method – the "unit" in 'unit testing'.	C function routine, with #defines to fake the presence of hardware.

Lean-Agile Partners

Issue: Rework Economics

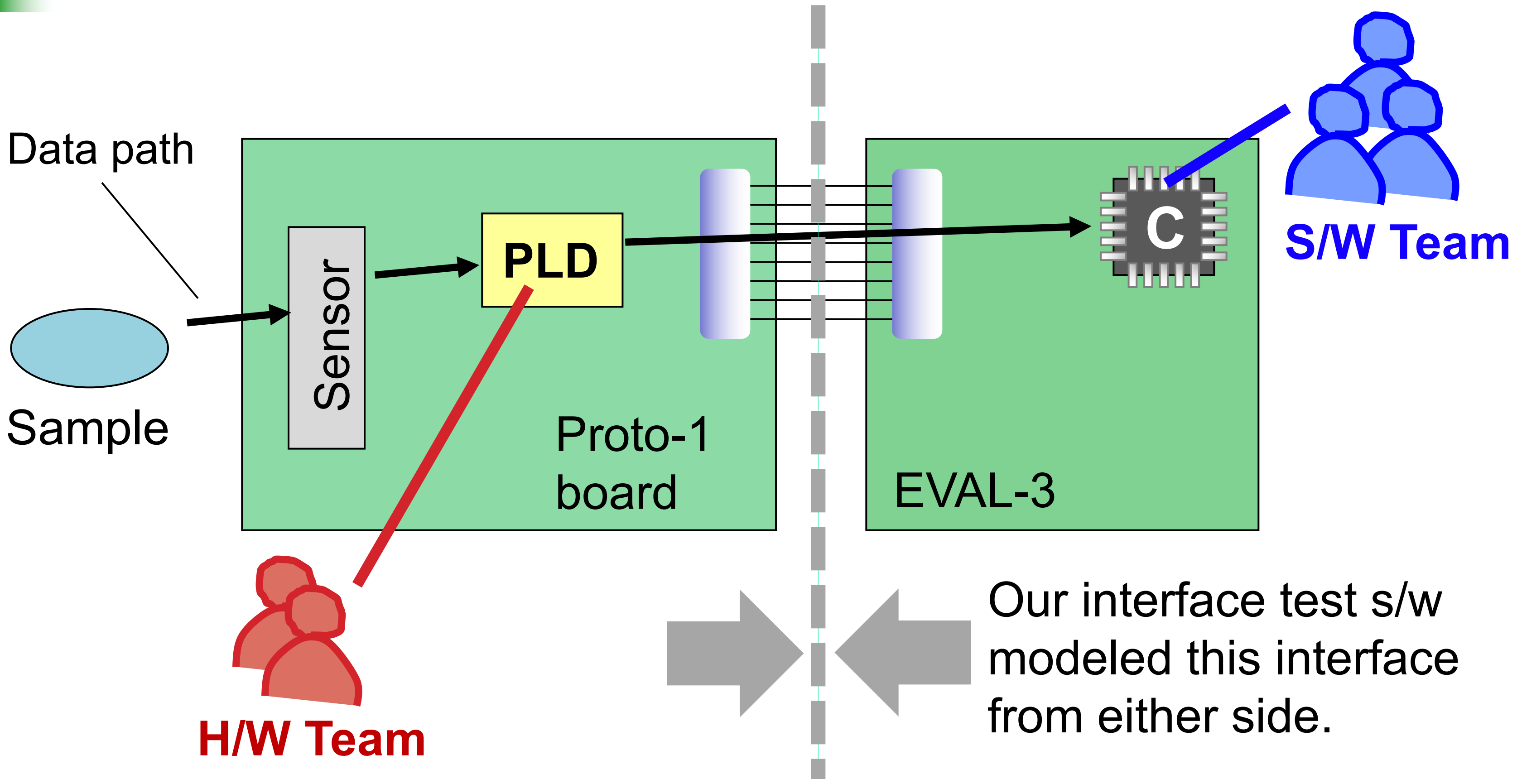
- Agile methods are meant for software
- Rework/ final production more constrained for h/w



Economics of final production are inverted for software compared to physical products

Lean-Agile Partners

Input Data Conditioning



Our interface test s/w modeled this interface from either side.

GOAL: For production, get an ASIC to replace the PLD

Input Data Conditioning

- Challenge
  - Architecture risk identified early by h/w and s/w leads – h/w data conditioning as mitigation
- Action
  - Collaboration to design a format for data from front end interface, and create interface test software to enforce it.
- Result
  - Change by either h/w or s/w side of intfc can be fully tested in less than 30 minutes (Old way would have identified risk late and over-designed)
  - Rework avoided! ASIC for data conditioning was bought only once; vendor used our interface test software in their verification step.

Lean-Agile Partners

Outline

- What prevents H/W – S/W collaboration?
  - Deeper specializations
  - Long H/W cycle times
  - Rework economics are opposite
- Benefits that surprised us
- Tips and Caveats

Lean-Agile Partners

Benefits that surprised us

- 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

Lean-Agile Partners

Benefits that surprised us

- Old rule: when there is a “mystery bug” s/w must prove it’s not a s/w bug before h/w will check it –
  - s/w guilty till proven innocent
  - Became h/w guilty till proven innocent!
- Only the s/w team was using Agile practices, but…
- Result was h/w became more Agile “without trying”

Lean-Agile Partners

Outline

- What prevents H/W – S/W collaboration?
  - Deeper specializations
  - Long H/W cycle times
  - Rework economics are opposite
- Benefits that surprised us
- Tips and Caveats

Lean-Agile Partners

Tips and Caveats

- Always have working hardware every sprint – it's everyone's shared reality
- Build key interfaces early; simulate them if necessary
- Use the simplest tooling possible & keep it under team's control
- Have team lab space for h/w and for s/w

Lean-Agile Partners

Contact Info

Nancy Van Schooenderwoert
LeanAgilePartners
nancyv@leanagilepartners.com

@vanschoo

Services

- Remote coaching for managers and tech leaders
- Training/ coaching for Agile hardware development
- Specialty services for medical device companies:
  - Incremental risk management
  - Incremental documentation
  - Intro to Agile course for mid-level managers
- Customized training/ coaching

Lean-Agile Partners

Appearances

Dec 6, 2017 11am – noon EDT
Webinar: "Agile is More Than Software"
By N. Van Schooenderwoert and Brian Shoemaker
Email bshoemaker@shoebarassoc.com to sign up

Jan 22-24, 2018
SDMD (Software Design for Medical Devices) in Boston!
https://sdmdconference.iqpc.com/

Feb 19-22, 2018
SDMD (Software Design for Medical Devices) in Munich
https://sdmdglobal.iqpc.co.uk/

Mar 12-14, 2018
Medical Device Pathways conference in Munich
http://medicaldevicepathways.iqpc.co.uk/

Lean-Agile Partners