ShoeBar Associates

Lean-Agile Partners

On Being Nimble:

Lean Development Principles in Medical Device Software Development

Brian Shoemaker
ShoeBar Associates
Nancy Van Schooenderwoert
Lean-Agile Partners Inc.

Thesis

Applying lean principles to software development, done properly, leads to higher productivity with far lower defect rates than traditional "linear" models, without loss of rigor or documentation.

Key Concepts

- Design vs. Production
- Lean principles applied to development
- Agile practices: iteration, feedback, cumulative testing, tracking
- Types of risk: project (technical, planning) vs. product (hazards)
- Hazard Analysis: FTA, FMECA
- Software development lifecycle

Nancy’s Background

- 15 years safety-critical systems experience
- 10 years agile team coaching
- 3 years agile enterprise coaching
- Industries: Aerospace, Medical Devices, Sonar Weaponry, Scientific Instruments, Financial Services
- Electrical Engineering and Software Engineering, embedded systems

Brian's Background

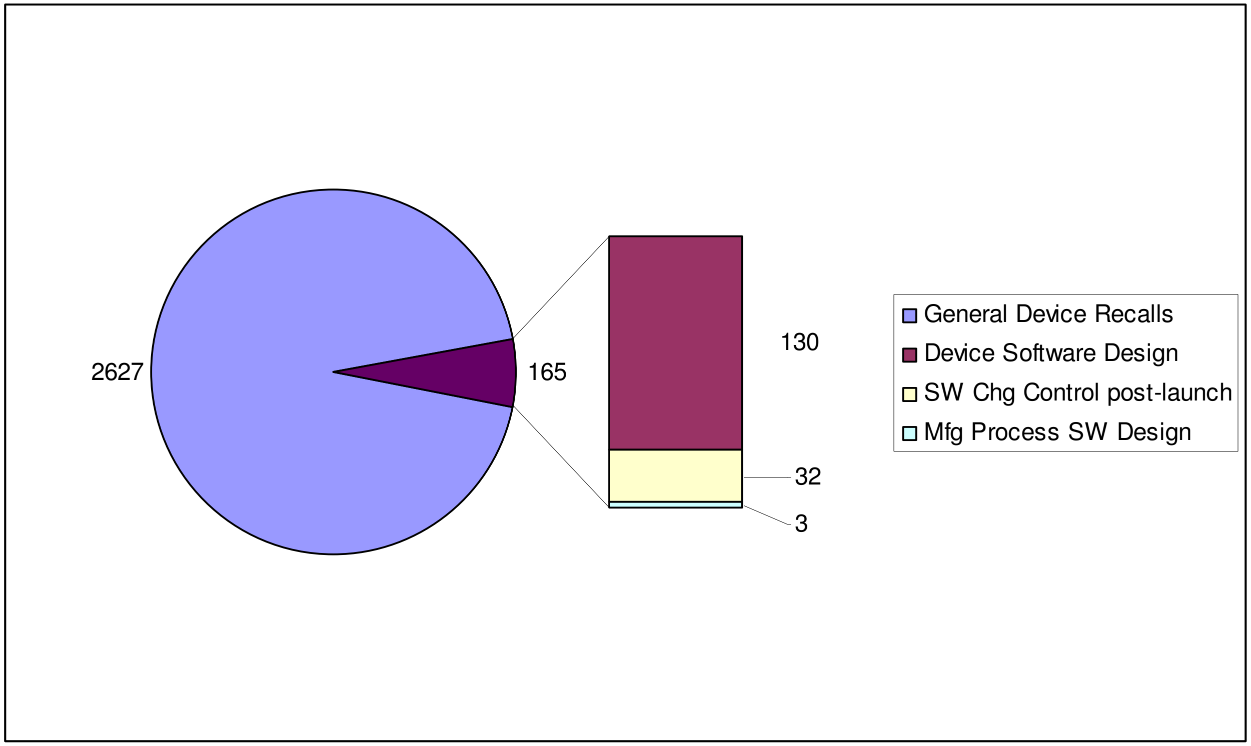
- Originally an analytical chemist
- 15 y in clinical diagnostics (immunoassay):
  analytical support → assay development → instrument software validation
- 6 y as SW quality manager (5 in clinical trial related SW)
- 4 y as independent validation consultant to FDA-regulated companies – mostly medical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

On Being Nimble: Lean Principles in Med Device SW Development

- **Why a new lifecycle?**
- What are the roots of this new lifecycle?
- How does the new lifecycle handle long range planning?
- How can hazard management fit into the new lifecycle?
- What are the keys to successful use in medical device development?
- Teams report positive experiences

Consider FDA software recall data

CDRH study: 2792 total medical device recalls, 1983-1991



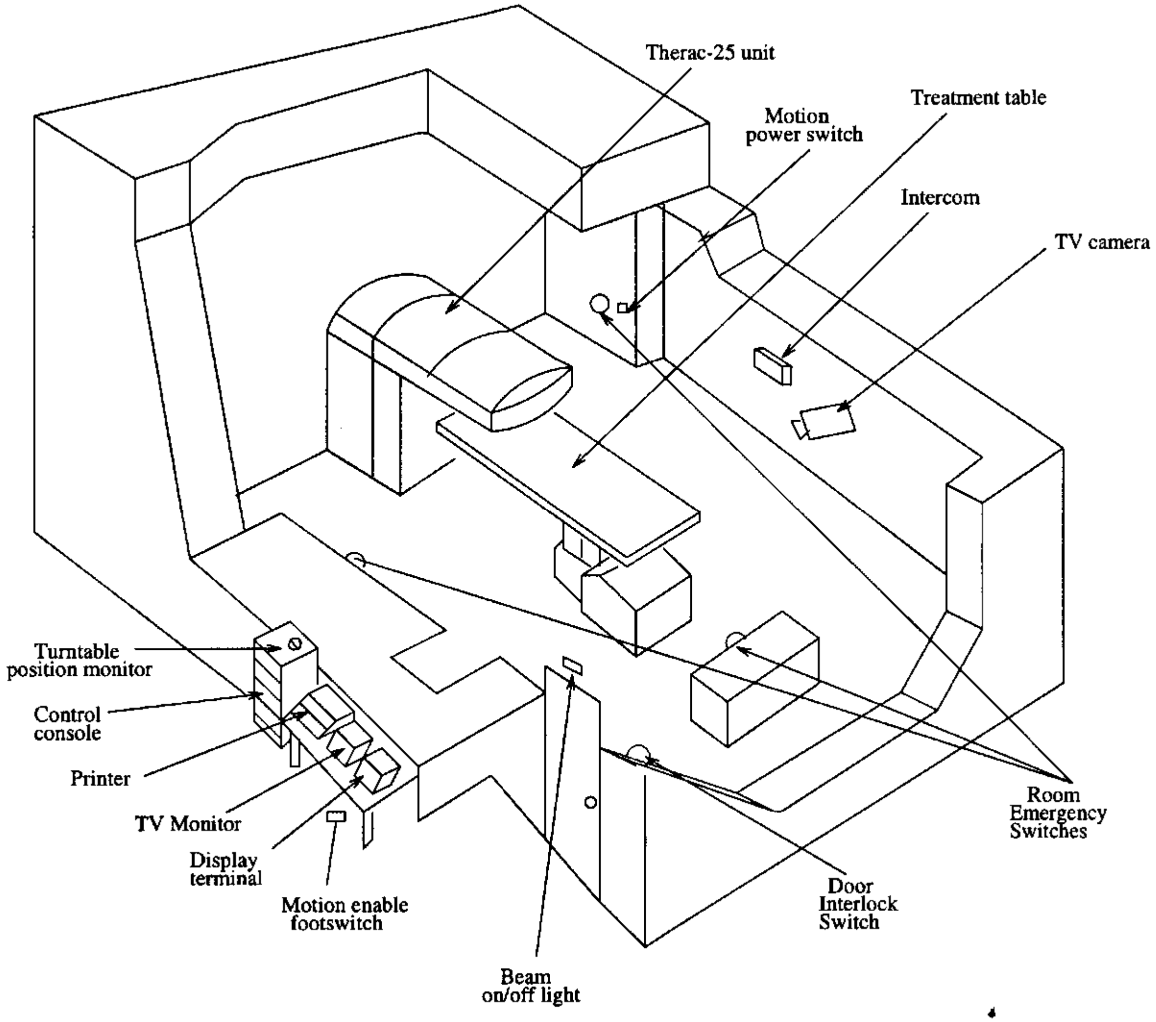
- 165 recalls (6% of total) software-related

Of the 165:

- 133 (81%) Inadequate software design
- 32 (19%) Post-launch software change control

Source: FDA CDRH, 1992.

Software Can Do Dangerous Things

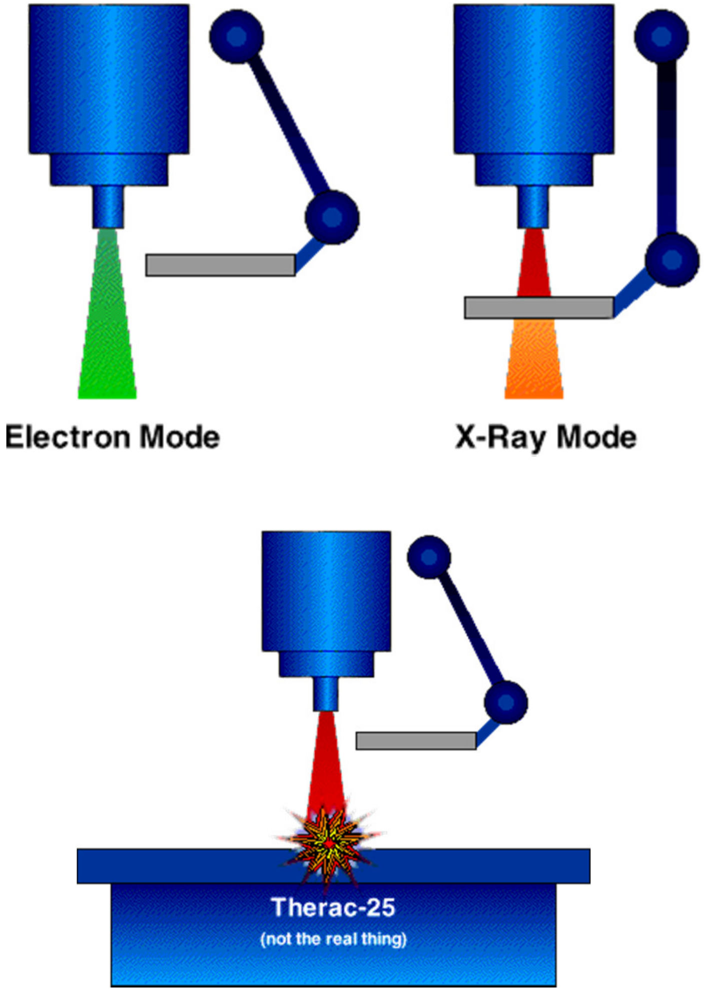


Therac-25: Brief Summary

- Linear accelerator system built for cancer therapy
- Instrument was further advancement of earlier models (controlled entirely through software)
- 11 Units installed in US / Canada; hundreds of patients treated (thousands of treatments)
- Mechanism: radiation beam destroys cancer tissue
  - Electron beam treats shallow tissue
  - X-rays penetrate deeper, minimal damage to overlying area
  - X-rays produced by hitting metal target with high-energy electrons
- Six overdose accidents (3 fatal): June 1985, July 1985, December 1985, March 1986, April 1986, January 1987
- Overdoses (~100x intended dose, ~20x lethal whole-body dose) traced to two specific software errors

The Therac-25 Safety Issue

- Single electron gun produces both modes
- In x-ray mode, electron energy must be ~100x higher (target is a good attenuator)
- Low energy + target = underdose
  High energy + no target = huge overdose



More Recent Examples Abound

- **Oct 24, 2012 – Automated Clinical Chemistry System**
  http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRES/res.cfm?id=112676
  Shuttle to Barcode Reader and Cap Piercer alignment procedure cannot be performed on certain DxC configurations with v5.0.11 software installed – if misaligned, operator cannot align.
- **Sep 26, 2012 – Radiological Image Processing System**
  http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRES/res.cfm?id=111796
  During renumbering of the patient database, if multiple patients with an identical name are present, the software will assign all subsequent records to the first patient record.
- **May 08, 2012 - Stereotactic Radiosurgery System**
  http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRES/res.cfm?id=108734
  When using respiratory gating with patients in the prone position, the gating curve is inverted.
- **April 09, 2012 - Pharmacy compounding system recalled**
  http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRES/res.cfm?id=81930
  Software issue with compounding pump resulting in over delivery of component ingredients during routine operations.
- **February 15, 2012 - Telehealth Monitor recalled**
  http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfRES/res.cfm?id=106852
  System collects patient vital signs data and periodically transmits to remote physician. Blood glucose (BG) results from a specific meter that the home user tags with either the meter hypoglycemic or general "asterisk" flags (*) will not be automatically transmitted through the monitor, or displayed to the health care provider through the data server.

Are glitches unique?

- Specific computed tomography scanner:
  9 software recalls, March 2011 - August 2012
- Specific clinical chemistry system:
  7 software recalls, March 2011 – July 2012
- Overall, January 2011 – Oct 31 2012:
  259 recalls with “software” in reason field

Right Problem, Wrong Solution

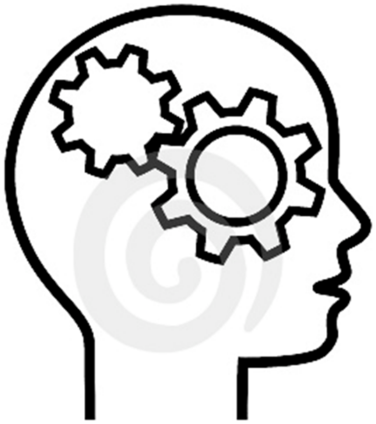
- Software issues prompt significant number of recalls
- Many still claim solution lies in rigorous, stepwise development
- Our position is that a different development lifecycle is needed
- But we still arrive at same goal!
- GPSV, IEC 62304 talk about *outputs* but do not dictate a specific *process*

Why the wrong solution?

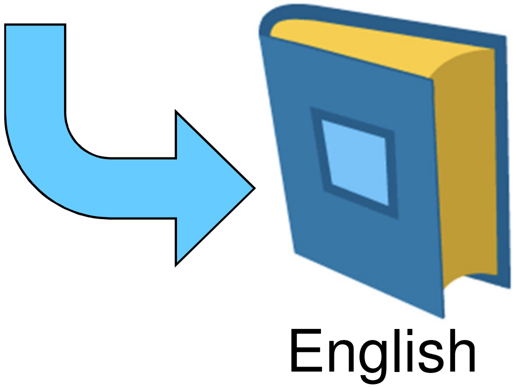
- Fallacy: coding software = production
- If all design was complete when spec was written then we need only to conform to spec
- But does that happen for software?
- Software is a *complexity magnet* -
  - For things expensive to do in hardware
  - When it can compensate for h/w issues
  - For complex user configurations
  - Capturing business rules

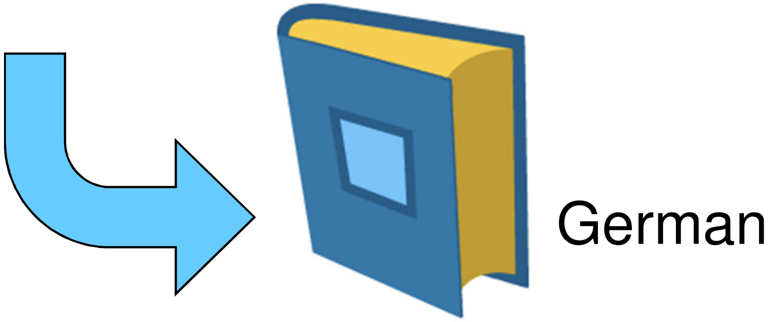
Design work or Production work?

- Writing a book is Design
  - Requires creativity
  - Expected to try multiple times before its right
  - Test is whether it satisfies customers



- Translate to German is Production
  - Should get it right the first time
  - Test is whether meaning conforms to original book’s text (i.e. as spec)





Design or Production?

- “Real” engineering:
- Output, e.g. circuit boards ready to ship
- Build, done by factory workers; big % of costs
- Design spec = circuit layouts, fabrication notes…
- Design is all the activities that create correct inputs to the build (fabrication) work

- Software engineering:
- Output, executable image ready for download
- Build, done by linker & compiler; almost free
- Design spec = software listings (the final version of the listings)
- Design therefore is what precedes the correct, final software listings: coding, architecture, analysis, reqmts.

Ref. ‘What is Software Design?’ article by Jack W. Reeves for C++ Journal, 1992.

Questions?

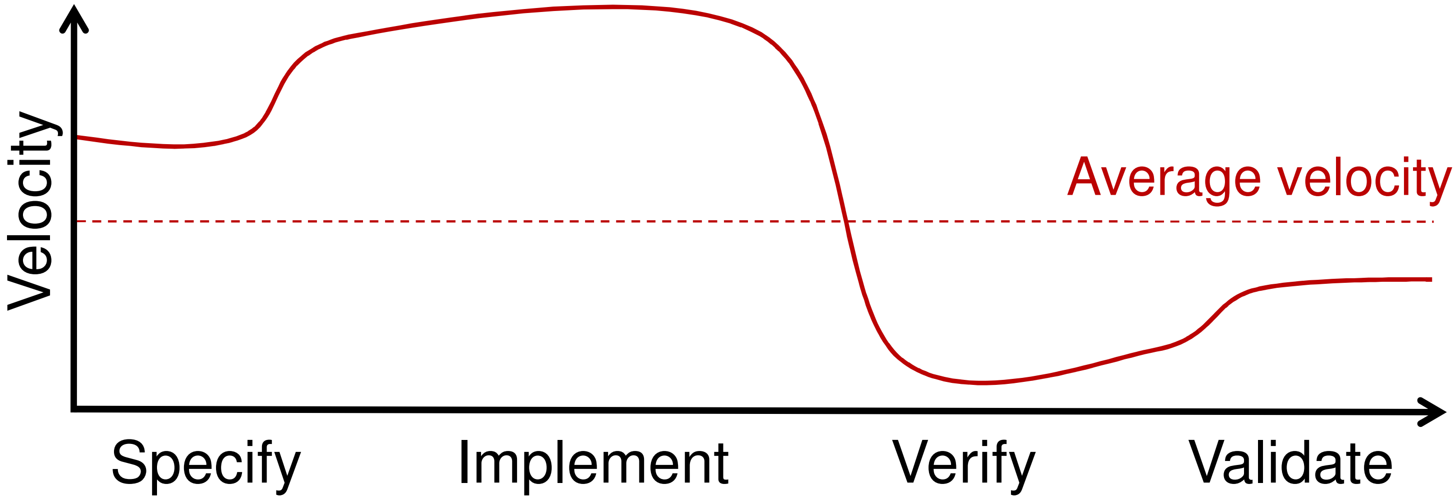
On Being Nimble: Lean Principles in Med Device SW Development

- *Why a new lifecycle?*
- **What are the roots of this new lifecycle?**
- How does the new lifecycle handle long range planning?
- How can hazard management fit into the new lifecycle?
- What are the keys to successful use in medical device development?
- Teams report positive experiences

ShoeBar Associates Lean-Agile Partners

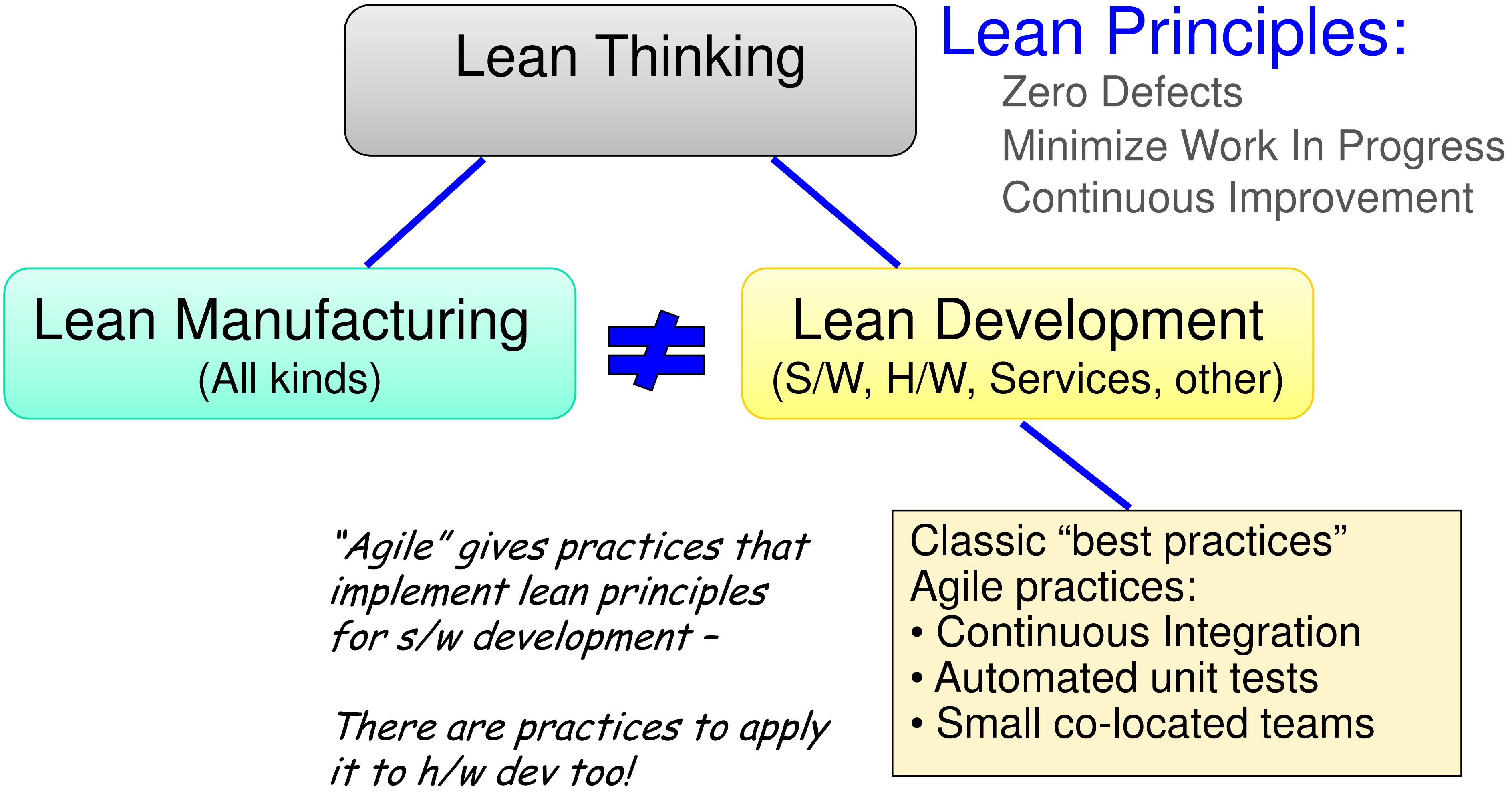
Common scenario…

- Project velocity varies greatly
- Much slower at integration time



Solution: Pace yourself! It's a marathon, not a sprint

Lean Thinking



Lean Thinking

Lean Principles:

Zero Defects
Minimize Work In Progress
Continuous Improvement

Lean Manufacturing
(All kinds)

≠

Lean Development
(S/W, H/W, Services, other)

"Agile" gives practices that implement lean principles for s/w development –

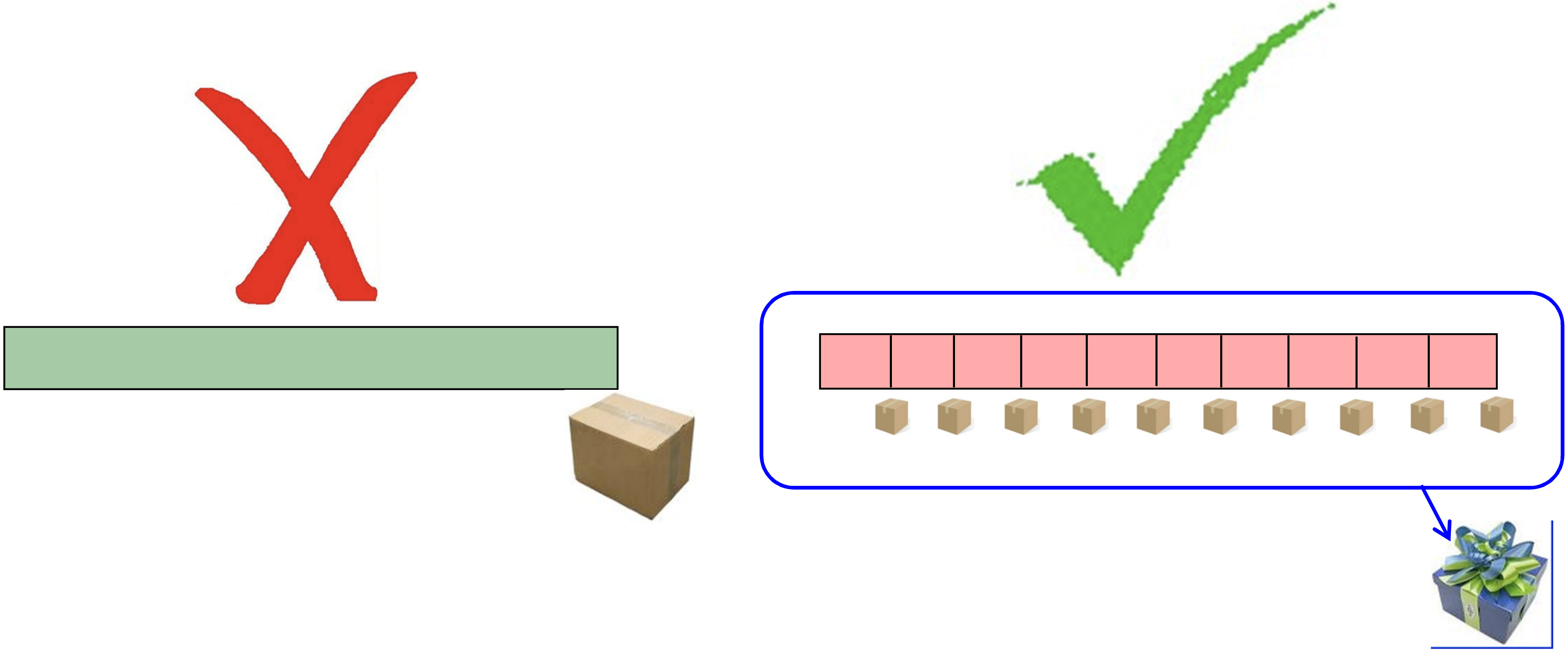
There are practices to apply it to h/w dev too!

Classic "best practices"
Agile practices:

- Continuous Integration
- Automated unit tests
- Small co-located teams

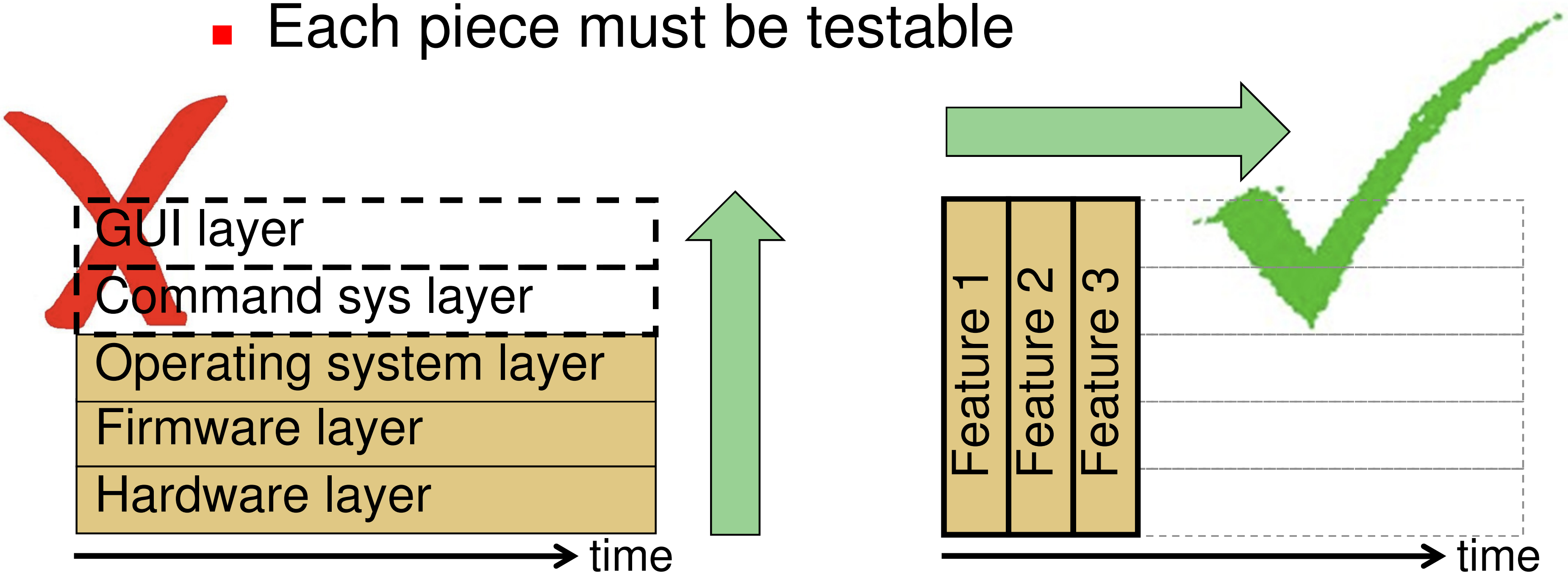
Avoid late integration

- Integrate new work as you go
- Incremental deliveries early & often



Deliver incrementally

- To deliver incrementally you must:
  - Carve the work into functional pieces
  - Each piece must be small
  - Each piece must be testable



ShoeBar Associates Lean-Agile Partners

Example user story

- Story – Card, Conversation, Confirmation –

  headline, narrative, test

Story

Cards have the headline

System can read a single HART value at a fixed address

Narrative details captured in documents

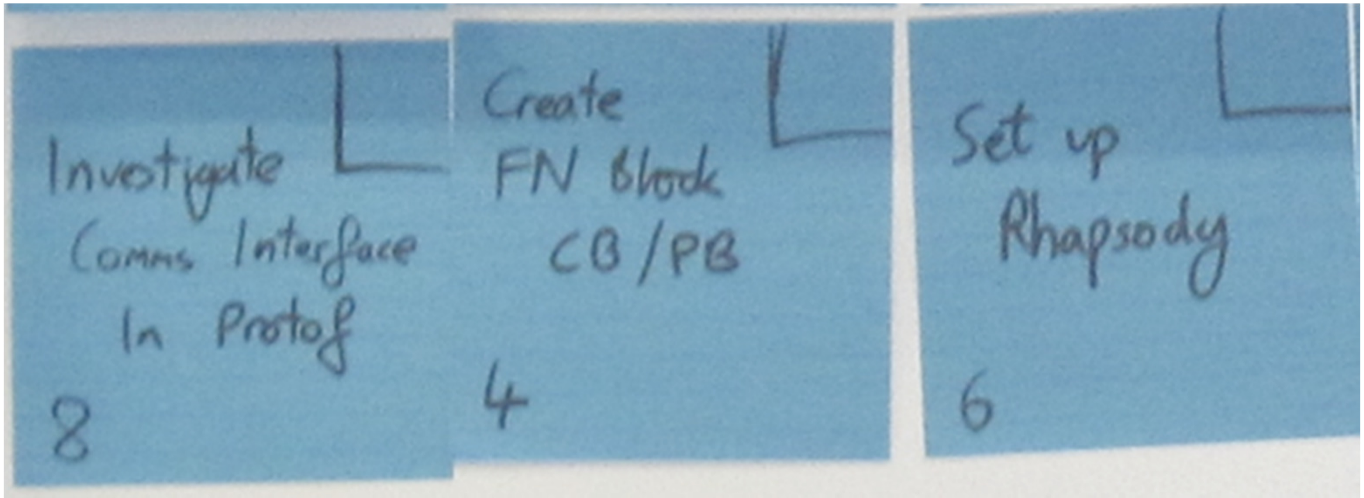
Conditions of Satisfaction

Get expected response to Cmd #1 with
- Single master
- Using present hardware
- Update < 1 second

CoS becomes the root of story acceptance test

An old idea: If you have a clear goal, you are much more likely to achieve it.

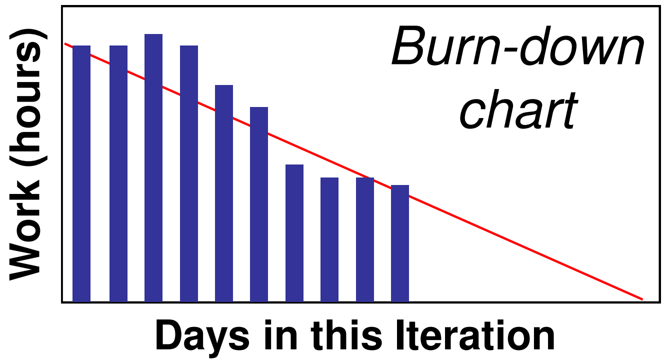
Tasks within the user story



- Tasks are estimated in hours
- Tasks are not assigned to people in advance
- Estimates are team-owned, assume avg. performer
- At end of each day, all future work hours are summed

Total transparency

- Status reporting is not separate from team's own way of tracking their work

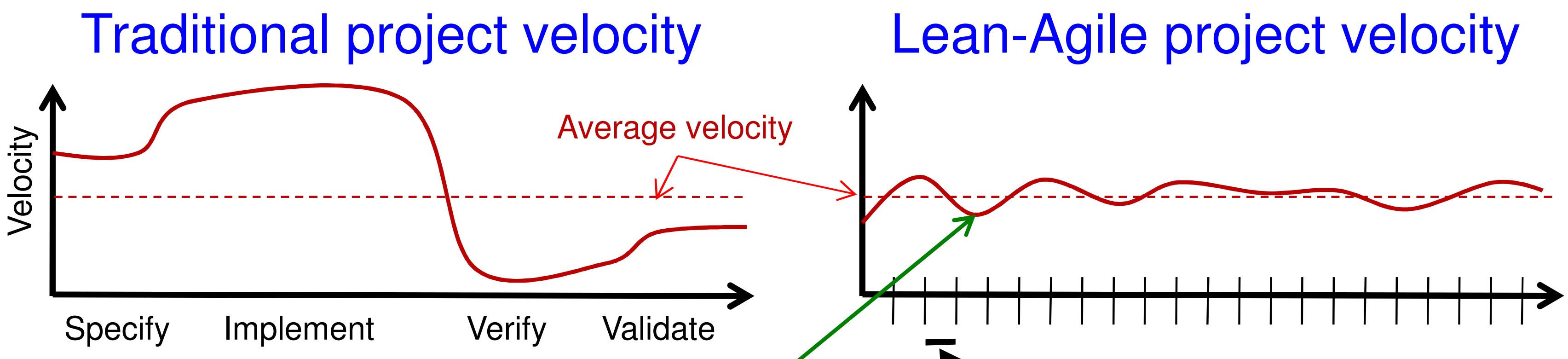


Each day:

- Team estimates hours remaining for each task
- All remaining hours are summed
- That total is today's data point on burn-down chart

How we avoid bad news late

- Traditional project has varying speeds
- Lean-Agile project's control mechanisms keep it close to its average velocity (in story points)



Traditional project velocity

Lean-Agile project velocity

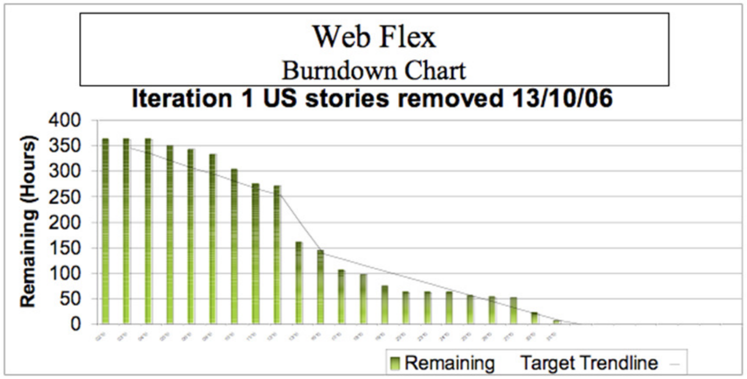
Velocity represents # Stories per unit time

Each Iteration does all 4 of
- Specify
- Implement
- Verify
- Validate

For every story in the iteration

When things go wrong

- Renegotiating iteration goal
- De-scoping stories from iteration
- “Could we have anticipated this?”
  - Consider root causes of issues
  - Then do continuous improvement



- End of iteration - “Yes or No: Did you build trust with your product owner?”

Questions?

On Being Nimble: Lean Principles in Med Device SW Development

- *Why a new lifecycle?*
- *What are the roots of this new lifecycle?*
- **How does the new lifecycle handle long range planning?**
- How can hazard management fit into the new lifecycle?
- What are the keys to successful use in medical device development?
- Teams report positive experiences

ShoeBar Associates Lean-Agile Partners

Exercise: Thermostat function

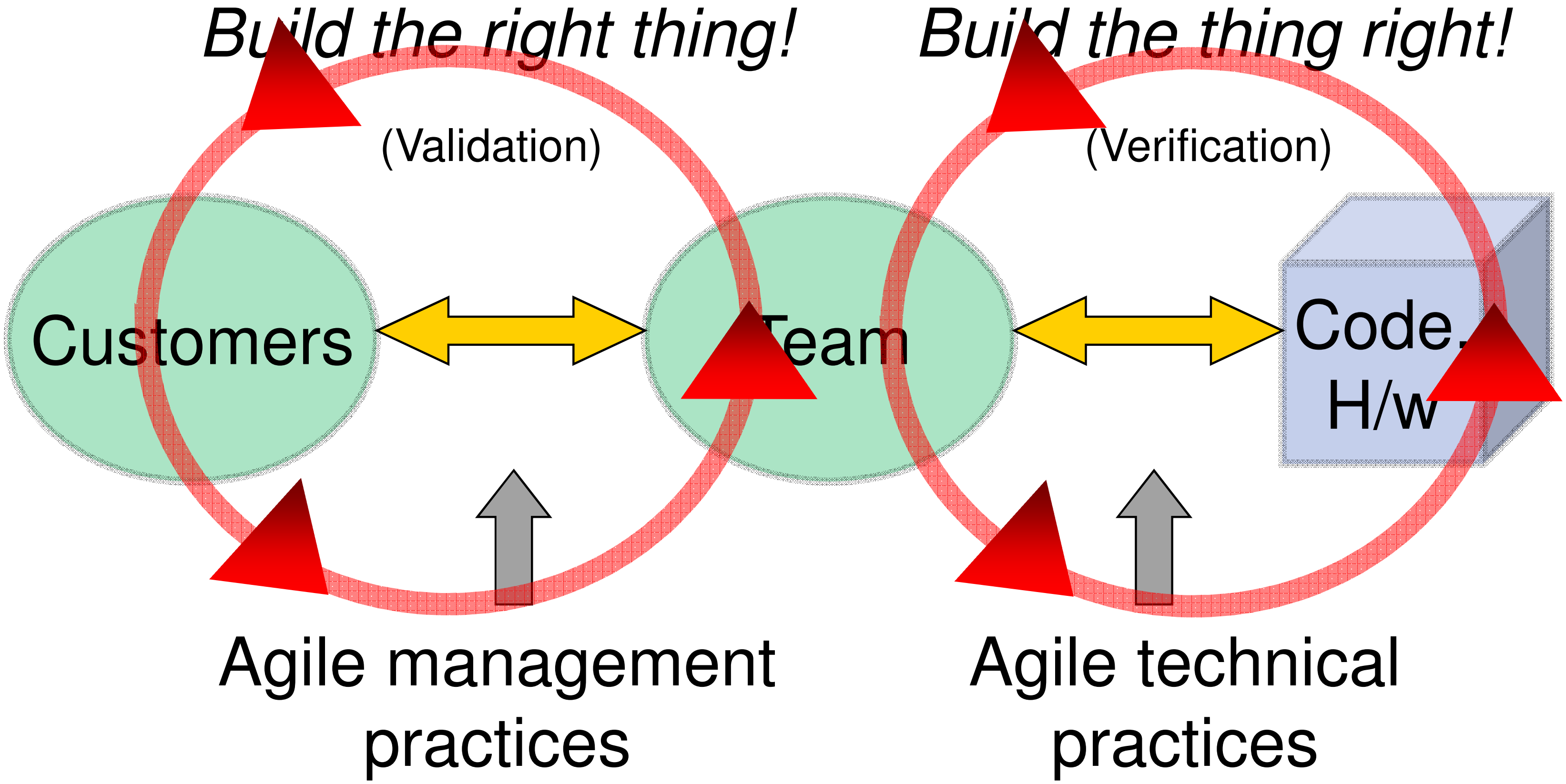
- Thought experiment: You have no thermostat for your home’s furnace, and it’s winter
  - You must use a switch to turn the furnace on or off
  - It’s either all the way on, or it’s off
  - This is called “open loop” control
- Question: What percent of the time will you probably be uncomfortable?

ShoeBar Associates Lean-Agile Partners

Process risk

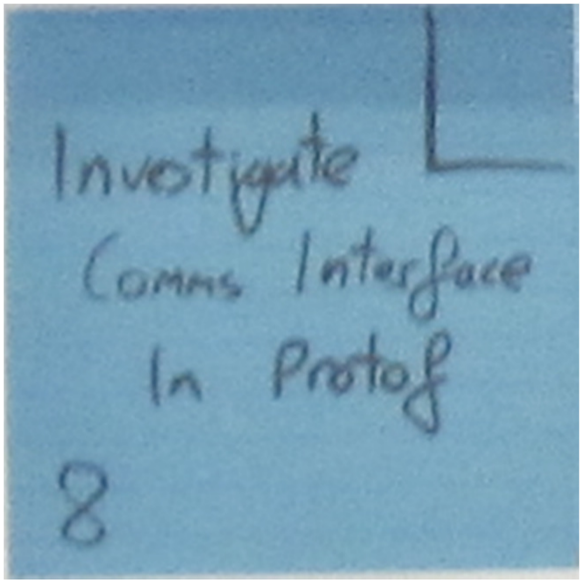
- Furnace with no thermostat ensures house nearly always too cold or too hot
    - Because you're an active control element
    - Analogous to managers who must stomp out "brushfires" all the time
- With thermostat back – comfortable all the time
    - *Because you're no longer an active control element – you're freed to do more important things*
- Engineering solution to poor control: Closed Loop
    - Adds feedback: Agile process has 2 primary feedback loops

Partnership: Business - Technical



Technical risk: estimation

- Story: “Read a single HART bus value”



Investigate Comms Interface In Protof

8

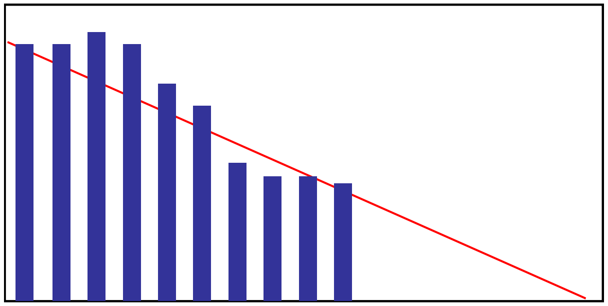
- 8 hours????
- Don’t know what the investigation will uncover!

- If your estimate will really be only a *guess*, you have an R&D task

Tracking R&D tasks

Regular task in a Story

~~10~~ 8



10 hrs estimated. 6 hrs worked. But 8 hrs remaining. 8 summed into burn down chart to compute the remaining work.

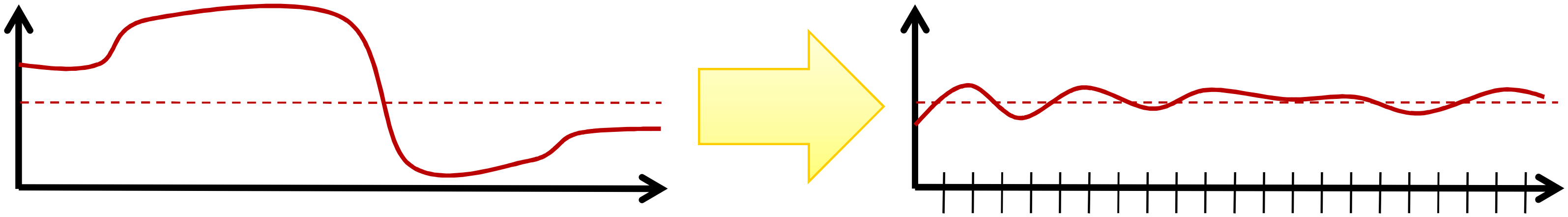
R&D task in a Story

~~10~~ 4

10 hrs ~~estimated~~ Time-boxed. 6 hrs worked. *Subtract* to get 4 hrs remaining. 4 summed into burn down chart to compute the remaining work.

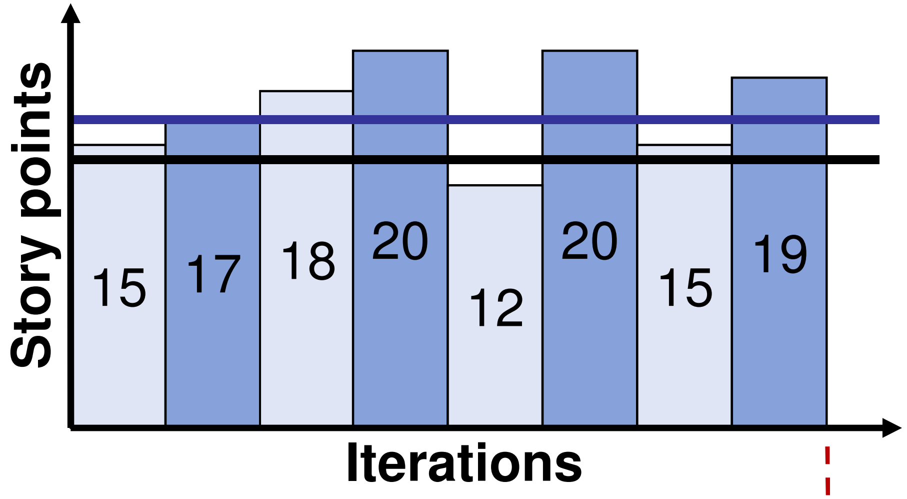
Addressing the 'planning risk'

- Incorporate new knowledge at each iteration
- Teams pace themselves: no "integration traffic jam"



- Project Managers can cover more projects for same effort –
  - No need to 'fight fires' late in their projects; no need to be the *active control element*

Predictable project speed

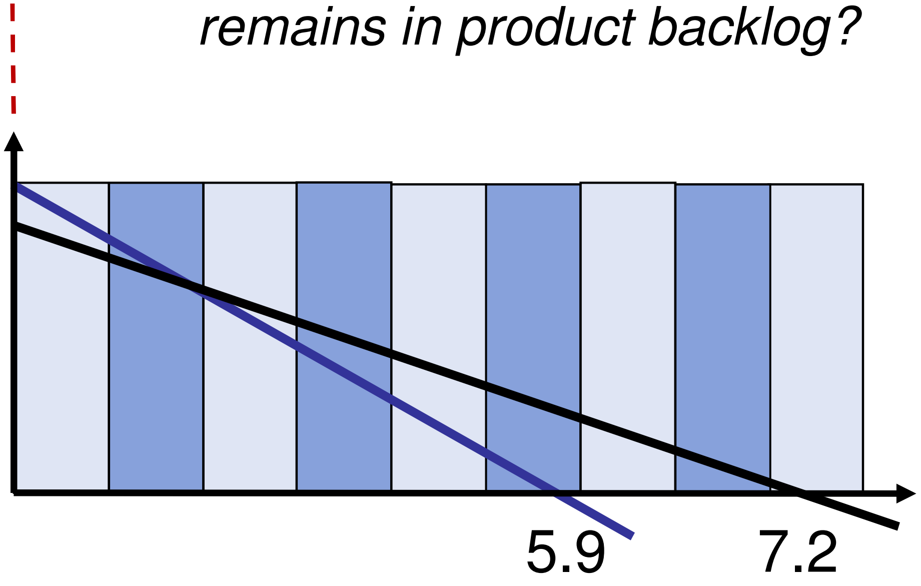


Mean (Last 8) = 17

Mean (Worst 3) = 14

Q. How long to finish project if 100 story points of work remains in product backlog?

A. If it’s 14 points/iter, then it takes 7.2 iterations. If it’s 17 points/iter. then it will take 5.9 iterations. You can be conservative or not, as appropriate.



Questions?

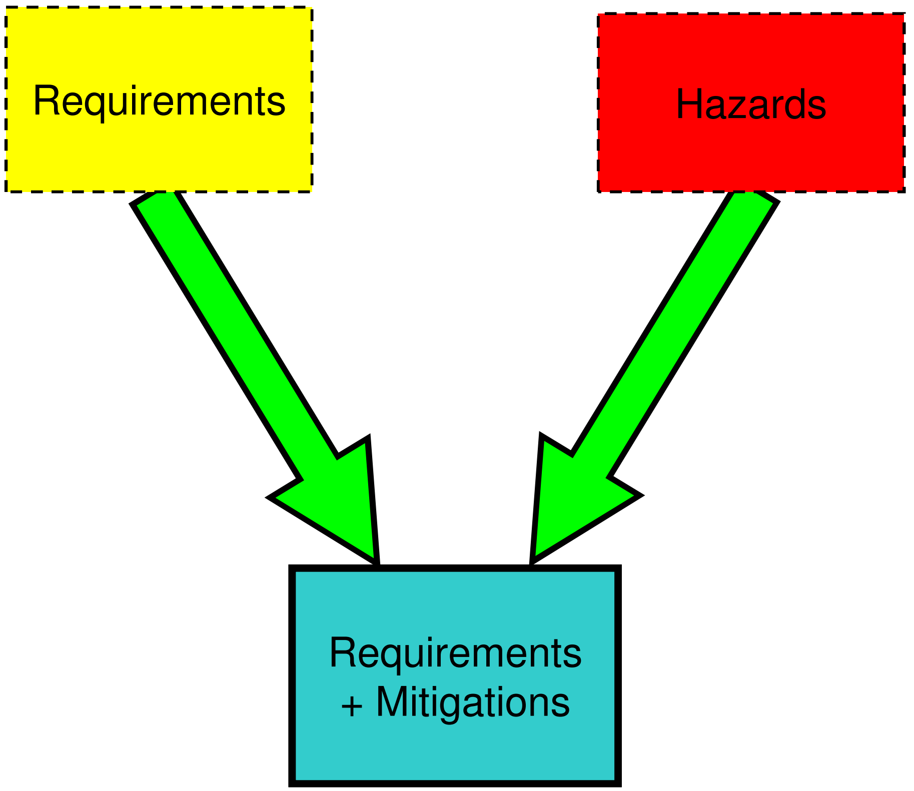
On Being Nimble: Lean Principles in Med Device SW Development

- *Why a new lifecycle?*
- *What are the roots of this new lifecycle?*
- *How does the new lifecycle handle long range planning?*
- **How can hazard management fit into the new lifecycle?**
- What are the keys to successful use in medical device development?
- Teams report positive experiences

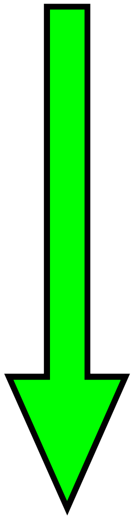
ShoeBar Associates

Lean-Agile Partners

Requirements / Hazards: Converging Analyses



Early in project
Preliminary
High-level
Approximate



Late in project
Refined
Detailed
Specific

Review: rank hazards by impact

	Probability:			
Severity:	High	Occasional	Low	Remote
Major	U	U	U	A
Moderate	U	A	A	N
Minor	A	A	N	N

Acceptability is ranked as follows:

U = unacceptable – mitigation required

A = ALARP (as low as reasonably practicable) – mitigate as reasonable; risk decision must be documented and reviewed

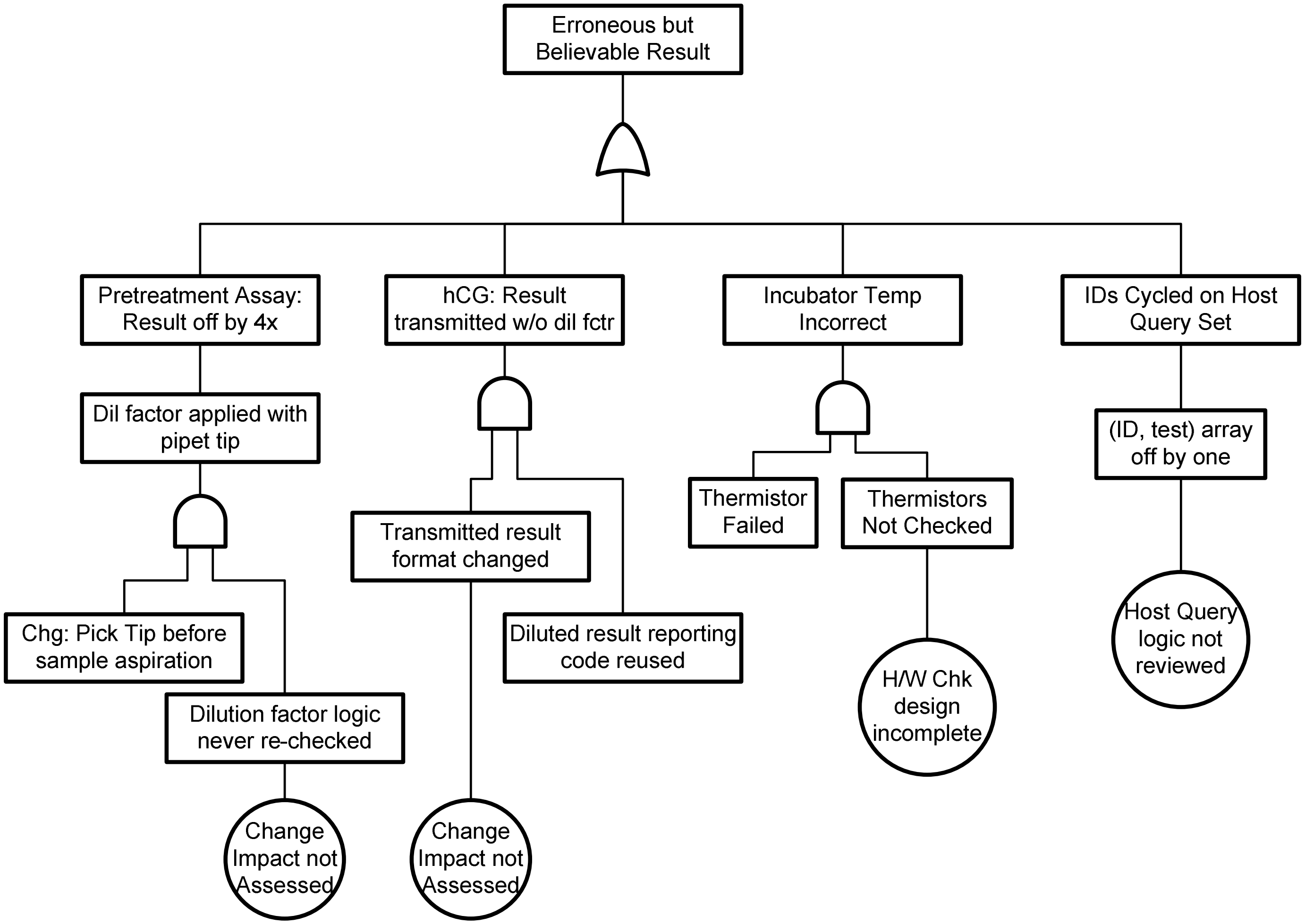
N = negligible – acceptable without review

However, for software the probability axis disappears.

Hazards: analyze early and often

- Systematic methods (FMEA / FMECA, FTA) help analyze potential hazards
- Evaluate hazards repeatedly throughout project
- Just as requirements (aka User Stories) become more refined as design evolves -
- So identifying <u>hazard mitigations</u> is changing or adding to requirements
- Think of a hazard as a negative user story

FTA: Work Back from Potential Hazards


Erroneous but Believable Result
Pretreatment Assay: Result off by 4x
hCG: Result transmitted w/o dil fctr
Incubator Temp Incorrect
IDs Cycled on Host Query Set
Dil factor applied with pipet tip
(ID, test) array off by one
Thermistor Failed
Thermistors Not Checked
Transmitted result format changed
Chg: Pick Tip before sample aspiration
Diluted result reporting code reused
Host Query logic not reviewed
H/W Chk design incomplete
Dilution factor logic never re-checked
Change Impact not Assessed
Change Impact not Assessed

ShoeBar Associates Lean-Agile Partners

FMEA: Build Up from Component Failures

Failure Mode	Effect	Causes	S1	Mitigation	S2
Sample ID / results array off by one	Wrong results reported	Inconsistent array logic; incorrect initialization	5	Optional – operator approve results before saving	2
Initialization fails to warm up lamp	Can't perform analyses	Startup logic can be set to skip steps and left that way	4	Reset all startup parameters on initialization	1
Dilution factor associated with pipet tip, picked in advance	Wrong result reported (dilution applied to wrong sample)	Counting logic not rechecked when pick-in-advance process introduced	5	(a) Track dilution and pipet tip separately; (b) show dilution with reported result	1

S1 = Severity rating before mitigation; S2 = severity rating after mitigation

Severity (sample values only): 5 = critical; 1 = nuisance

Note this analysis does not include two of the standard engineering estimates: occurrence (probability) or detection.

Hazards: Often Caught in Context

- Direct failure
  Software flaw in normal, correct use of system causes or permits incorrect dosage or energy to be delivered to patient.
- Permitted misuse
  Software does not reject or prevent entry of data in a way that (a) is incorrect according to user instructions, and (b) can result in incorrect calculation or logic, and consequent life-threatening or damaging therapeutic action.
- User Complacency
  Although software or system clearly notes that users must verify results, common use leads to over-reliance on software output and failure to cross-check calculations or results.

Hazards: Often Caught in Context

- User Interface confusion

  Software instructions, prompts, input labels, or other information is frequently confusing or misleading, and can result in incorrect user actions with potentially harmful or fatal outcome.

- Security vulnerability

  Attack by malicious code causes device to transmit incorrect information, control therapy incorrectly, or cease operating. No examples in medical-device software known at this time, but experience in personal computers and "smart" cellular phones suggests this is a serious possibility.

Lean-Agile adapts well to hazard mitigation

- Early analysis not static – review & revise as iterations proceed
- Users / product owner have multiple chances to uncover hazard situations
- Hazards can be simulated via “mock objects” in test suite
- Flexible, adaptive method can react to hazards learned during development (considered “negative user stories”)

Questions?

On Being Nimble: Lean Principles in Med Device SW Development

- *Why a new lifecycle?*
- *What are the roots of this new lifecycle?*
- *How does the new lifecycle handle long range planning?*
- *How can hazard management fit into the new lifecycle?*
- **What are the keys to successful use in medical device development?**
- Teams report positive experiences

Know the objections & benefits

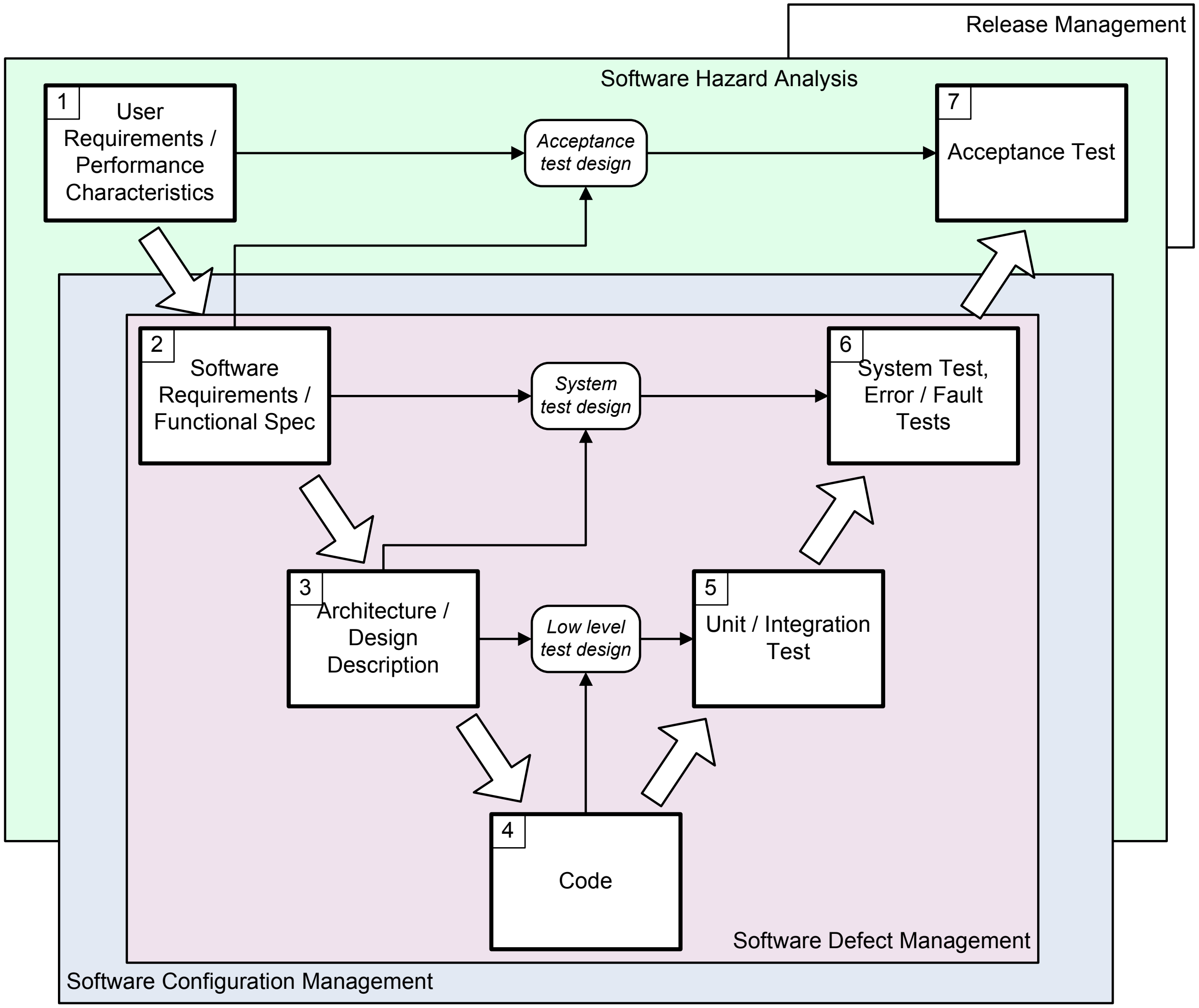
Points to counter:

- Lack of defined requirements
- Lack of structured review/release cycles
- Lack of documentation

Advantages to offer:

- Ability to resolve incomplete / conflicting requirements
- Ability to reprioritize requirements (mitigations) as system takes shape
- Many chances to identify hazards (controls not frozen too soon)

Know your SW lifecycle

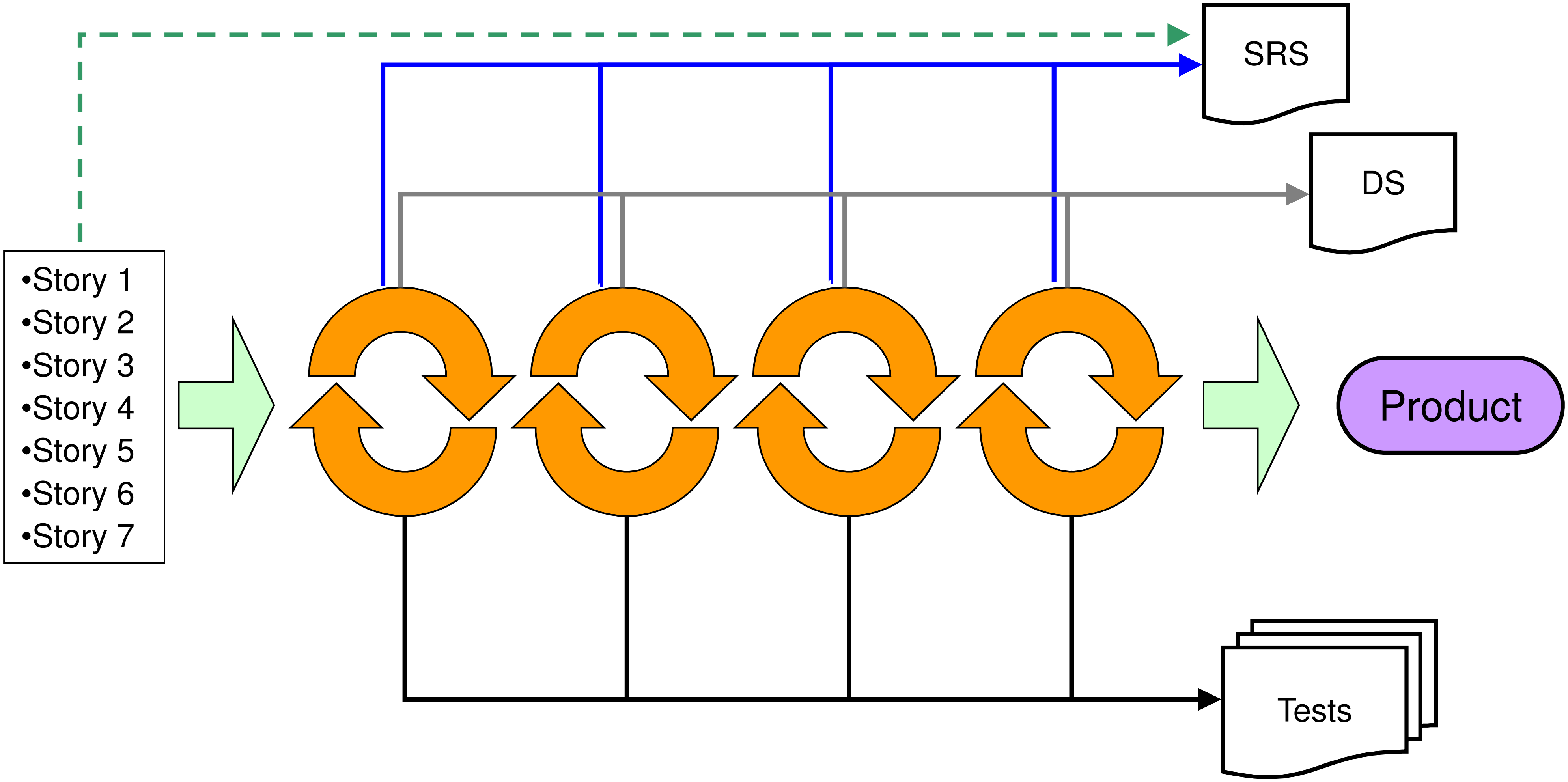

Release Management
Software Hazard Analysis
1
User Requirements / Performance Characteristics
Acceptance test design
7
Acceptance Test
2
Software Requirements / Functional Spec
System test design
6
System Test, Error / Fault Tests
3
Architecture / Design Description
Low level test design
5
Unit / Integration Test
4
Code
Software Defect Management
Software Configuration Management

ShoeBar Associates Lean-Agile Partners

Document Effectively but Flexibly

- SOPs: focus on deliverables
  - Cover all required areas
  - Specify outputs, not strict order of completion
- Development outputs: focus on information
  - Requirements, architecture/design, hazard analysis
  - View as deliverables rather than process support
  - Consider nontraditional form if this makes information capture easier or more automatic

Capture knowledge as work proceeds



- Story 1
- Story 2
- Story 3
- Story 4
- Story 5
- Story 6
- Story 7

Use appropriate tools

- Initial user stories – may simply be index cards
- Requirements manager as they’re elaborated
- Unit test harness
- Code-comment document extraction
- User-focused functional / system test engine – best if tied to requirements, e.g. FitNesse

Don’t forget communication

- With customer / product owner – for input and ongoing feedback
  - Iter. end Demo; discussions during iteration
- Among team members – frequent but brief, to build team dynamics
  - Daily stand-up meeting; team room conversations
- With management – to show progress and build trust
  - Information radiators; Iter. end Demo

Advanced Topics

- Creative work *on a deadline*!?!
- What about documentation?
- Why we measure in “story points”
- Using “planning poker” for estimating
- “Epic points” for quick forecasting in large projects (>2 y)
- Specific practices for lean H/W development
- Job roles in a lean-agile team

ShoeBar Associates Lean-Agile Partners

Questions?

On Being Nimble: Lean Principles in Med Device SW Development

- *Why a new lifecycle?*
- *What are the roots of this new lifecycle?*
- *How does the new lifecycle handle long range planning?*
- *How can hazard management fit into the new lifecycle?*
- *What are the keys to successful use in medical device development?*
- **Teams report positive experiences**

What Do Defect Outcomes Suggest?

Team	Defects/Function Point	
Follett Software[1]	0.0128	agile
BMC Software[1]	0.048	agile
GMS[2]	0.22	agile
Industry Best[3]	2.0	traditional
Industry average[3]	4.5	traditional

[1] Computed from data reported in Cutter IT Journal, Vol. 9, No. 9 (Sept 2008), page 10
[2] “Newbies” paper presented at Agile 2006. See “References” slide for full citation.
[3] Capers Jones presentation for Boston SPIN, Oct., 2002

Agile Productivity

	"Biotech" re-implemented, as Agile (1)	"Biotech" original, as Waterfall (1)	SirsiDynix, as Agile (Scrum) (2)
Person Months	54	540	827
Lines of Java	51,000	58,000	671,688
Function Points (FP)	959	900	12,673
FP per Dev/month	17.8	2.0	15.3

1. M. Cohn, User Stories Applied for Agile Development, p. 175. Addison-Wesley, 2004 (Reported without giving company or project name, but it was a life sciences application.)
2. J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams," in HICSS'40, Hawaii International Conference on Software Systems, Big Island, Hawaii, describing SirsiDynix team.

ShoeBar Associates

Lean-Agile Partners

Case 1: Clinical Trial Tool

- Presented at Drug Information Assn annual mtg, June 09
- Collaborative development of data system for use in clinical trial
- Initial requirements written for RFP; project began ~11 months later (changed!)
- Initial goals (user stories) elaborated into lightweight form of Use Cases
- Elaborating also created RBE (requirements by example) which resulted in a form of system test script
- Design frozen only just before final iteration; Requirements frozen only after final iteration

Source: Vogel, DIA 2009.

CT Tool Approaches

- Each feature in an iteration designed/built based on one or more RBE
- Feature “Done” includes automated unit & system tests – no Big Bang testing at the end
- RBE is directly automated; all automated unit / system tests run regularly
- By start of UAT, had generated documentation equivalent to a traditional waterfall project – finalizing was relatively quick because of interim reviews
- Virtually no defects in UAT because of ongoing testing and review

Source: Vogel, DIA 2009.

CT Tool Lessons Learned

- Design changes even more than requirements and tests, so agility is key
- Create just enough design to *responsibly* start coding
- The more test driven, the less design documentation needed. Test driven = extensive unit test automation
- Active collaboration (biz analysts, testers, developers, and compliance) to create & automate RBE was powerful
- Automated, comprehensive system test freed test & compliance staff to focus on higher value activities

Case 2: Device Software

- Authors compared two projects (one Agile, one not): found that Agile gave lower cost, shorter development time, better accommodation of change, better test cases, and higher quality
- Used FDA's concept of "least burdensome approach" as part of their justification for using the Agile method
- Considered risk as integral part of development
- Iterative approach helped manage scope and limit feature creep

Source: Jenks & Rasmussen, Agile 2009.

Case 2: Comments

Developer:
“Control what you know, don’t let it control you.”

Client:
“At time of commercial launch, a number of features, once thought to be essential, were not included. Some were deferred as long as three years. Nonetheless, the product was considered highly successful and trading off nice to have features for three years of sales is an easy choice.”

Case 2: Reported Results

- High visibility – few surprises, able to manage and control
- Cost / duration: Agile project required 20-30% smaller team ***and*** shorter time, saved 35-50% cost, vs. non-Agile project
- Agile project gave higher quality – fewer overall defects, especially at end of project
- Agile project involved far better work-life balance and team morale (issues surfaced and managed in course of project, not saved for the end)

Epilogue

- AAMI has published a technical information report on use of Agile practices for medical device software
- Both industry and FDA representatives on committee
- Points echo many of those made here
- Released August 2012

  "AAMI TIR45:2012, Guidance on the use of AGILE practices in the development of medical device software"

Questions?

Quote for the Day

“It is not the strongest of the species that survive, not the most intelligent, but the one most responsive to change.”

- Charles Darwin

ShoeBar Associates Lean-Agile Partners

References

- FDA, CDRH “Software Related Recalls for Fiscal Years 1983-1991” (May 1992)
- Vogel, M. (EMC Consulting), “What it takes to support a Validated Agile Project,” presented at 45th annual meeting of Drug Information Association, San Diego CA, June 2009.
- Jenks, J.R. (AgileTek), and R. Rasmussen (Abbott), “Moving to Agile in an FDA Environment: An Experience Report,” presented at Agile 2009, Chicago IL, August 2009.
- Reeves, Jack W. “What is Software Design?” C++ Journal, 1992. http://www.bleading-edge.com/Publications/C++Journal/Cpjour2.htm
- “Newbies” paper full citation: Nancy Van Schooenderwoert, “Embedded Agile Project By the Numbers with Newbies” presented at Agile 2006, Minneapolis MN, July 2006
- Michael Mah “How Agile Projects Measure Up, and What This Means to You”, Cutter IT Journal, Vol. 9, No. 9 (Sept 2008), for data on BMC and Follett Software projects.
- Capers Jones, “Software Quality in 2002” presentation for Boston SPIN, Oct 2002. See http://www.boston-spin.org/talks.html#yr2001

Recommended Reading

- *Lean Software Development: an Agile Toolkit* by Mary & Tom Poppendieck
- *Leading Lean Software Development* by Mary & Tom Poppendieck
- *Release It!* By Michael Nygard
- Weyrauch, Kelly, “Safety-Critical. XP Rules.”, *Better Software*, July/August 2004.
- *The Elegant Solution* by Matthew May
- *The Goal* by Eliyahu Goldratt

FDA Safety References

- FDA: Design Control, Medical Devices (March 11, 1997) http://www.fda.gov/cdrh/comp/designgd.html
- FDA: General Principles of Software Validation (Jan 11, 2002) http://www.fda.gov/cdrh/comp/guidance/938.html
- FDA: Premarket Submissions, Software Contained in Medical Devices (May 11, 2005) http://www.fda.gov/cdrh/ode/guidance/337.html
- FDA: Off-The-Shelf Software Use in Medical Devices (Sep 9, 1999) http://www.fda.gov/cdrh/ode/guidance/585.html
- FDA: Cybersecurity for Networked Medical Devices, OTS Software (Jan 14, 2005) http://www.fda.gov/cdrh/comp/guidance/1553.html
- FDA draft guidance: Radio-Frequency Wireless Technology in Medical Devices (Jan 3, 2007) http://www.fda.gov/cdrh/osel/guidance/1618.html

Other Safety / Quality References

- AAMI TIR45:2012, Guidance on the use of AGILE practices in the development of medical device software
- IEC 60812:2006 (2nd ed) Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA)
- IEC 60601-1: 2005 (3rd ed) Medical electrical equipment – Part 1: General requirements for basic safety and essential performance
  (60601-1-4 "Programmable Electrical Medical Systems" is available standalone, but will not be in the future)
- IEC 62304:2006 Medical Device Software – Software Life Cycle Processes
- IEC TIR80002-1:2009, Medical device software - Part 1: Guidance on the application of ISO 14971 to medical device software
- ISO 13485:2003 (2nd ed) Medical devices – Quality management systems – Requirements for regulatory purposes
- ISO 14971:2007 (2nd ed) Medical devices – Application of risk management to medical devices

Contact Information

Nancy Van Schooenderwoert
Lean-Agile Partners, Inc.
162 Marrett Rd., Lexington, MA 02421
781-860-0212
NancyV@leanagilepartners.com
http://www.leanagilepartners.com

Brian Shoemaker, Ph.D.
Principal Consultant, ShoeBar Associates
199 Needham St, Dedham MA   02026
781-929-5927
bshoemaker@shoebarassoc.com
http://www.shoebarassoc.com

ShoeBar
Associates

Lean-Agile
Partners