Winning With Scrum

PMI Metrolina

January 2009

Who is Joe Little?

- Agile Practitioner, Coach, and Trainer
- Conducts Lean-Agile courses with Jeff Sutherland, the Poppendiecks, Mike Vizdos, Mark Pushinsky, Jim York and others.
- Active in Agile for 4 years
- Actively used Waterfall for 18 years; recovering Waterfallic
- Background in PM/PMO and management consulting to JP Morgan Chase, Citibank, Deutsche Bank (BT), BNP Paribas, BofA, Capital One, USAA, etc.
- 20+ years in projects
- Often mistaken for a real techie.
- MBA, CSM, CSP, CST

Attributions

- Contents developed by Joe Little; please give attribution if re-used
- Some ideas are borrowed from , Ken Schwaber, Jeff Sutherland, Mary & Tom Poppendieck, Mike Cohn, Kent Beck and others.
- Agile ideas come from many places; see first the Agile Manifesto.
- Lean ideas come from many places; we particularly like Womack & Jones.
- Thanks to many others for ideas and suggestions.

Context

- This is an intro to Scrum
- Scrum is a flavor of Agile
- Scrum is a way of organizing a project
- We believe (and lots of experience shows) that your <u>chances for success</u> and the <u>degree of success</u> can be greatly enhanced with Scrum
- Scrum, Agile, anything is NOT a silver bullet

Goals – What does winning mean?

- People have better lives
- It's more fun
- Each Team is 5x-10x more productive

6 Blind Men and the Elephant
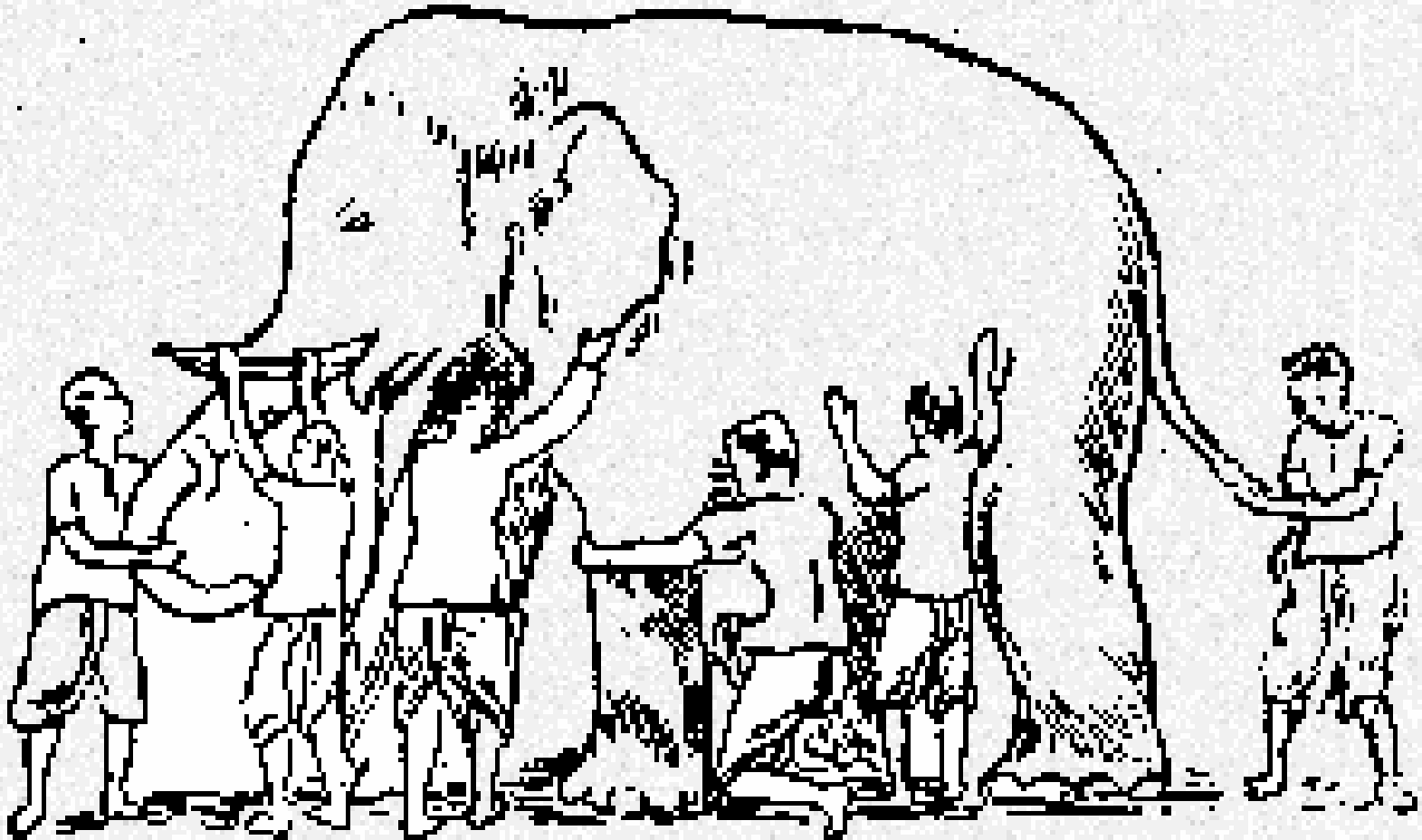
Little's Law (2)

"People are remarkably good at doing what they want to do."

See also the other Little's Law in Wikipedia
(but not a topic for tonight).

The basketball metaphor

©2008 LSU AgCenter

Olympic champion
NBA Champion
NBA Team
European League Team
NCAA Champion
ACC Champion
ACC Team
State HS Champion
County HS Champion
HS Varsity Team
HS JV Team
HS Rec Team
MS Team
7G Rec Team
6G Rec Team
5G Rec Team

Let's make things more visible

Stand if you have:

- Ever learned about Agile (1 hr +)
- Ever done Agile (1/2, 1, 2, 3+ years)
- Ever done a perfect project
- Done Agile but prefer Waterfall
- Done Agile and prefer Agile
- Done something else and prefer it
- Not done Agile but are already optimistic it is better
- Not done Agile but are sceptical it is better
- Heard things about Agile that concerned you

Myths about Agile

It is a myth that...Agile:

- Is a silver bullet
- Will solve my resource issues
- Has no planning / documentation / architecture / <insert pet peeve>
- Doesn't build on my previous experience & expertise
- Is undisciplined (Cowboy Coding)
- Creates quality issues
- Is new and unproven
- Is <u>not</u> being used by industry leaders

Some Firms

- Bank of America
- Wachovia
- Salesforce.com
- State Farm
- Vanguard Group
- BMC Software
- Nokia

- Google
- Yahoo
- Systematics
- PatientKeeper
- GSK
- IBM
- McKinsey

WIIFM (what's in it for me)

- More business value
- More, better, faster, cheaper
- Faster Time To Market (TTM)
- Better work & sustainable pace
- Better adaptation to Change
- More visibility
- More control
- More fun ("I wish I could be on that kinda team again.")
- More confidence
  (but still no silver bullet <sigh>)

Why Scrum?

Scrum was developed because...

- *Scrum is a solution to a difficult problem set.*

Note: Probably no two people will agree completely on the problem set.

Draft Problem Definition for Agile: 0.12 – page 1

- <u>Too much change</u>: from external business environment; from our business leaders; within our own business (processes); from people on the Team; from external technological; from within our technology organization
- Difficulty in accurately identifying the business requirements
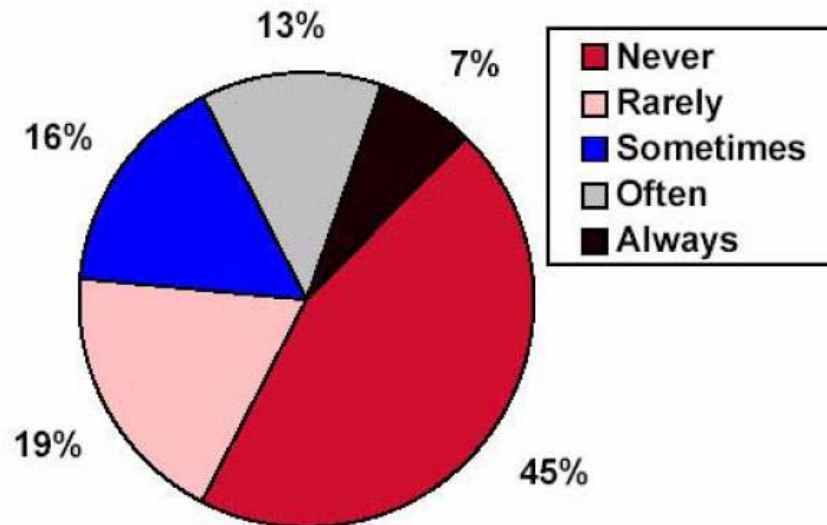- More complex business requirements
- More complex technology

Draft Problem Definition for Agile: 0.12 – page 2

- ❑ Problems of communicating about a highly abstract product (eg, a system)‍
- ❑ Difficulties in balancing the costs and benefits of systems – to deliver good ROI.
- ❑ Difficulties getting people to work together and be productive (motivation, silos, "communication", people issues, other agendas, etc.)‍
- ❑ Engrained opacity and obfuscation
- ❑ "If I were you, I wouldn't start from here." Old Irish saying

Another view of the problem set

Features & Function Usage

THE STANDISH GROUP

- ■ Never — 45%
- □ Rarely — 19%
- ■ Sometimes — 16%
- □ Often — 13%
- ■ Always — 7%

http://www.xp2003.org/xp2002/talksinfo/johnson.pd

Scrum: What is it?

Scrum 101

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.

- It allows us to rapidly and repeatedly inspect actual working software (every one-two-four weeks).

- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.

- Every one-four weeks everyone can see real working software. And decide to release it as is or continue to enhance it during the next Sprint.

Source: Mike Cohn

Scrum Roles

Pigs

- Product Owner
- Team member
- ScrumMaster

Chickens

- Stakeholder

Scrum Roles

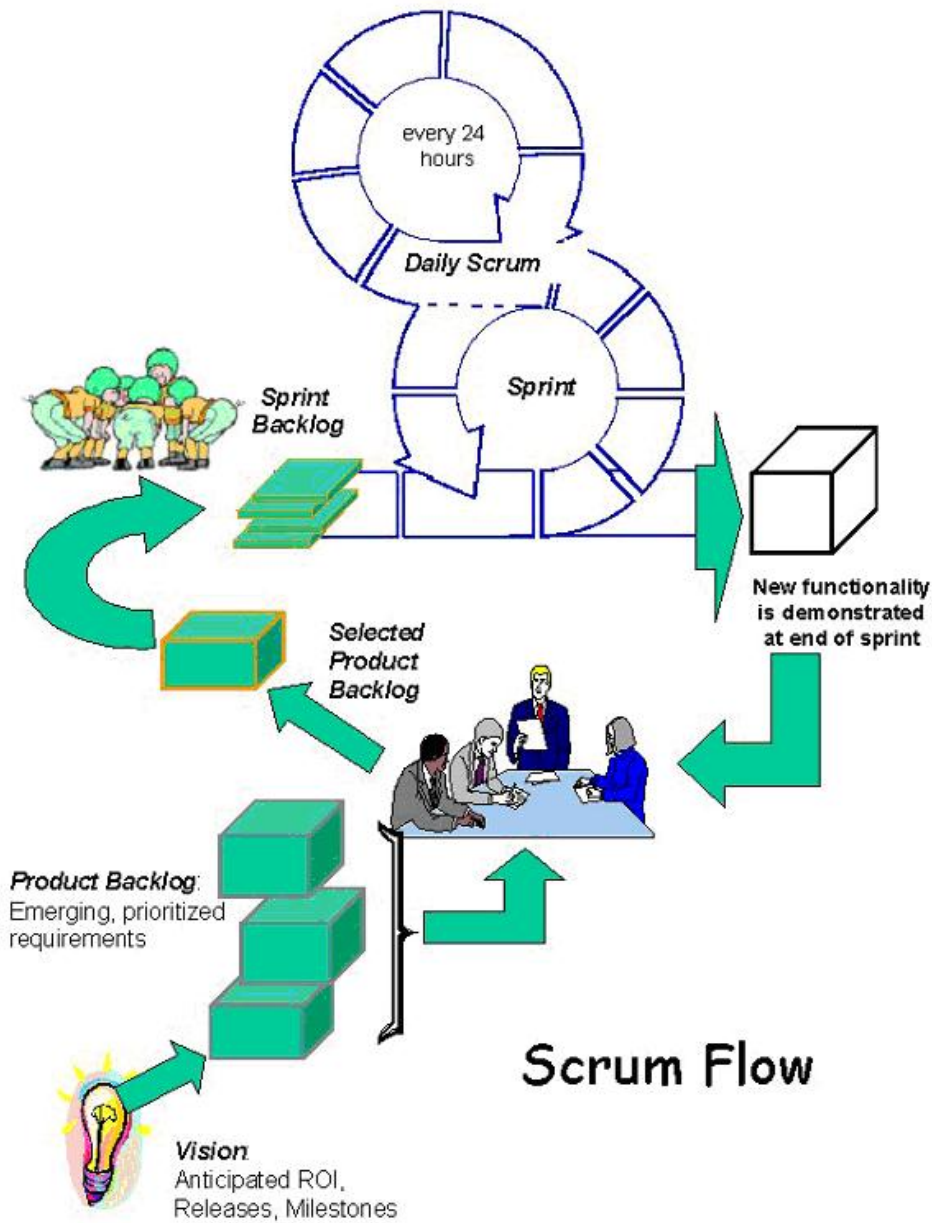|	
Product Owner	■ Owns the Product Backlog; gets it continuously refactored ■ Final prioritizer of the Product Backlog ■ Empowered to make decisions for all customers and users ■ Assures Biz info flows into Team (eg, presents and explains Product Backlog items)
Team member	■ Self-organizing ■ Seven plus or minus two (incl PO & SM) ■ Virtually all skills sets needed to build the product ■ Responsible for estimating and committing to work ■ Full autonomy and authority to make the Sprint a success
ScrumMaster	■ Responsible for the process ■ Responsible for maximizing team productivity (Impediments List) ■ Sets up and conducts meetings ■ Representative to management and team ■ Characteristics of a border collie or sheepdog

Key Scrum Artifacts

- Impediments List
- Product Backlog
- Sprint Backlog
- Sprint Burndown Chart
- The Increment (working software)
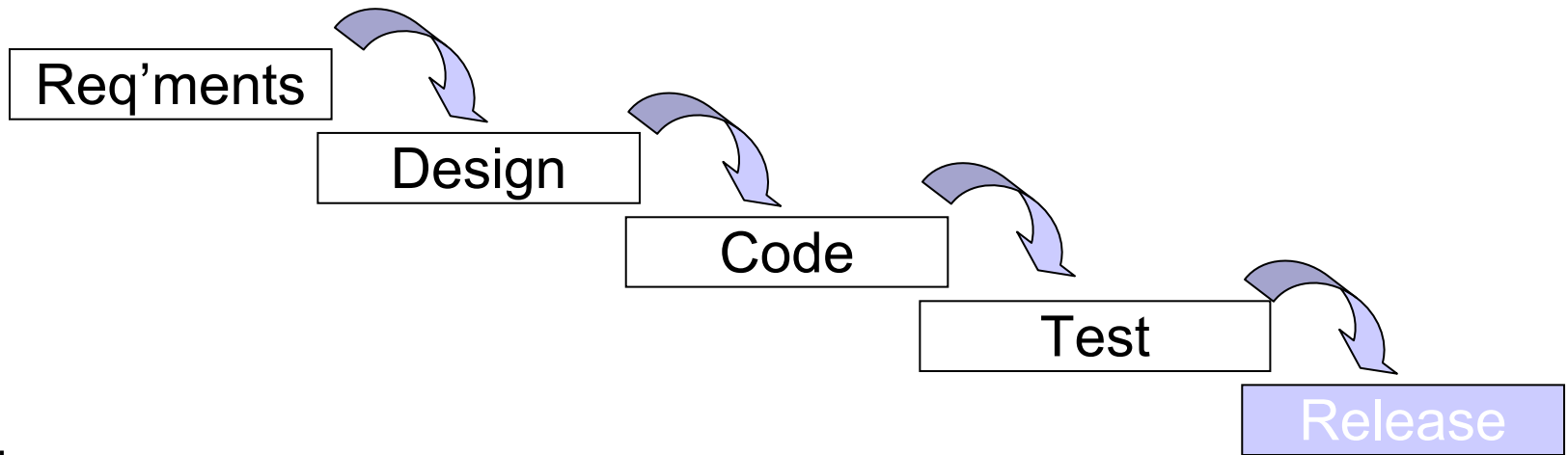- Release Burndown Chart

Prod Bklog list on cards
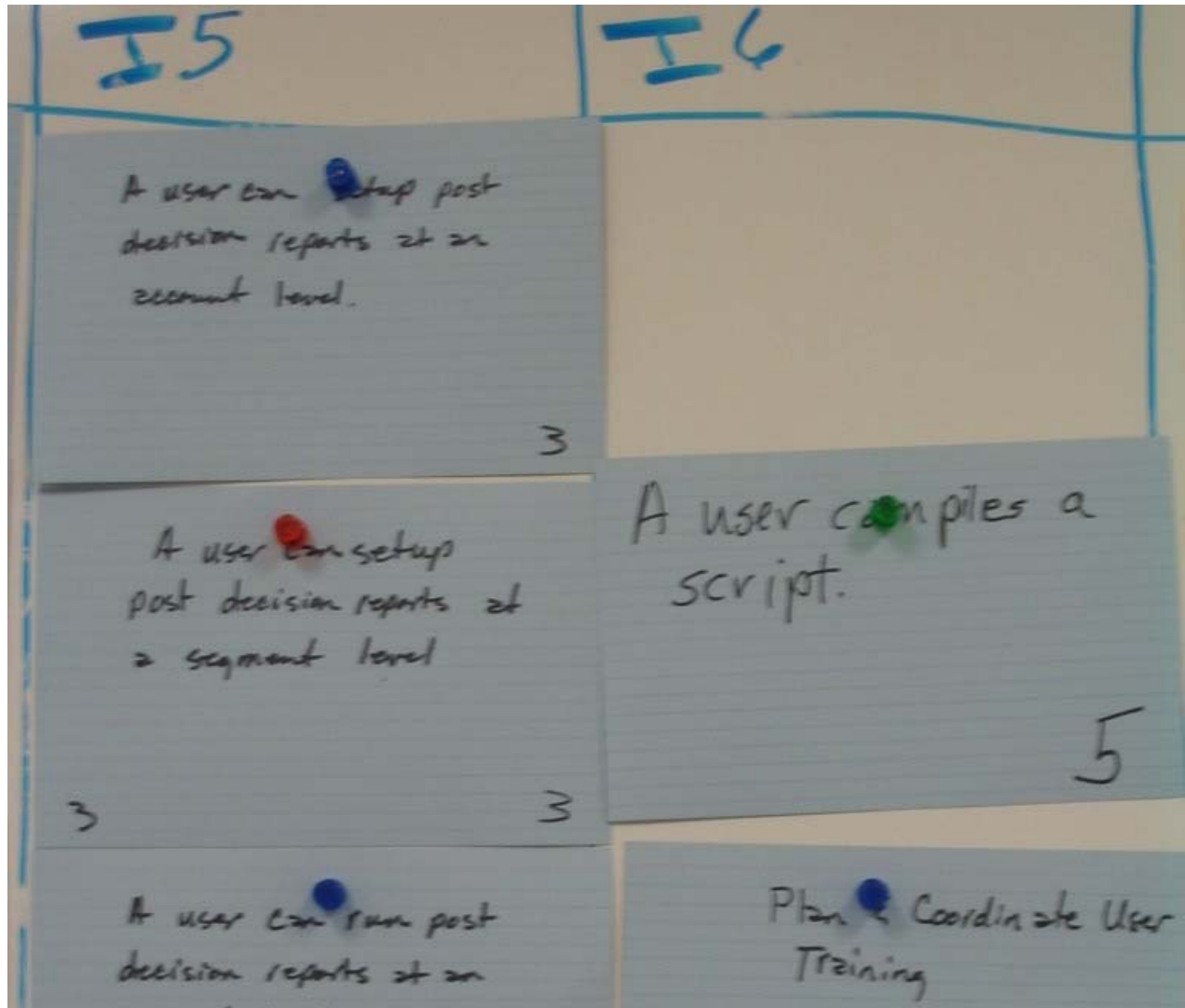
Key Scrum Meetings

- Release Planning
- Sprint Planning
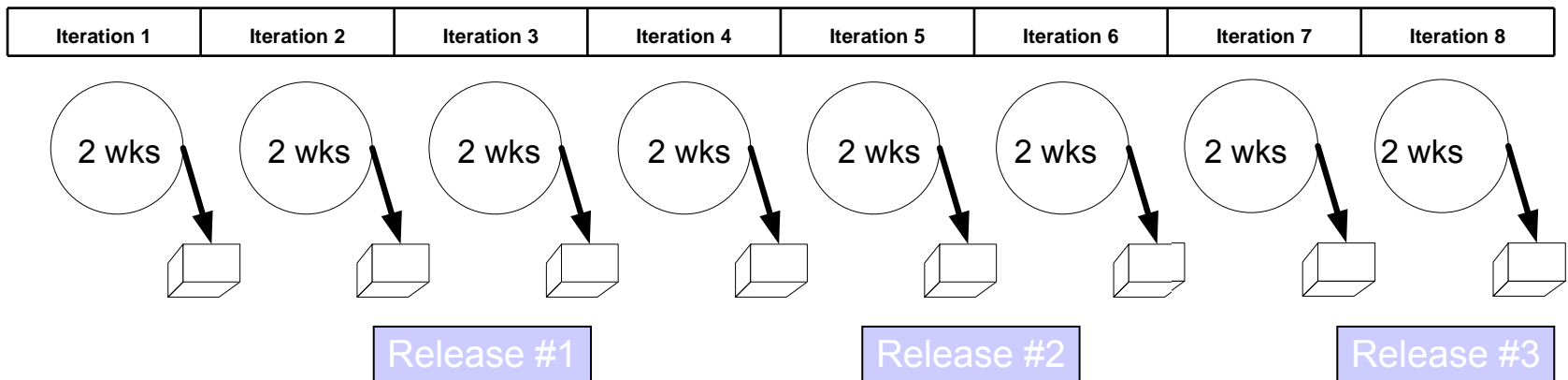- Daily Scrum (Daily Standup)
- Sprint Review (demo)
- Retrospective

every 24 hours

Daily Scrum

Sprint

Sprint Backlog

New functionality is demonstrated at end of sprint

Selected Product Backlog

Product Backlog: Emerging, prioritized requirements

Vision: Anticipated ROI, Releases, Milestones

Scrum Flow

Waterfall vs. Agile – Delivering Incremental Value

Waterfall

Req'ments

Design

Code

Test

Release

Agile

Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7	Iteration 8
2 wks	2 wks	2 wks	2 wks	2 wks	2 wks	2 wks	2 wks

Release #1

Release #2

Release #3

What's going on?

The Agile Manifesto

"We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right,

we value the items on the left more"

Agile Principles - 1

1. Our highest priority is to <u>satisfy the customer</u> through early and <u>continuous delivery</u> of valuable software.

2. <u>Welcome changing</u> requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. <u>Deliver working software frequently</u>, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must <u>work together daily</u> throughout the project.

5. Build projects around <u>motivated individuals</u>. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of <u>conveying information</u> to and within a development team is face-to-face conversation.

Agile Principles - 2

7. Working software is the primary <u>measure of progress</u>.

8. Agile processes promote <u>sustainable</u> development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to <u>technical excellence and good design</u> enhances agility.

10. <u>Simplicity</u>--the art of maximizing the amount of <u>work not done</u>-- is essential.

11. The best architectures, requirements, and designs <u>emerge</u> from <u>self-organizing</u> teams.

12. At regular intervals, <u>the team reflects</u> on how to become more effective, then tunes and adjusts its behavior accordingly.

PDCA

- The Deming Cycle: a scientific approach

- Plan, Do, Check (Study), Act
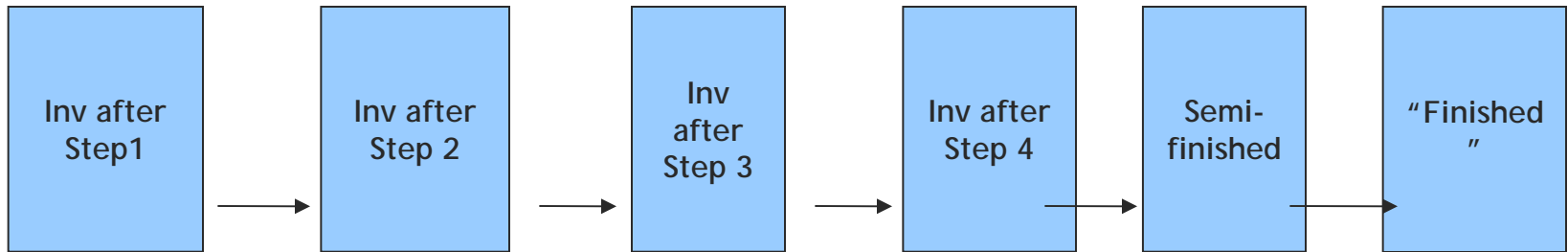
- Nested PDCA's. Repeated PDCA's. Iterative PDCA's.
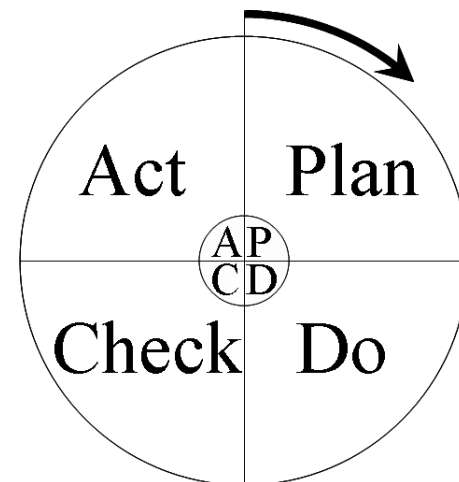
What is Scrum?

"Scrum is a way of harnessing creativity,
   the joy of work,
   [and] the pleasure of teamwork
   into extraordinary productivity
   in building complex products."

Ken Schwaber

Just-In-Time

Inv after Step1	→	Inv after Step 2	→	Inv after Step 3	→	Inv after Step 4	→	Semi-finished	→	"Finished"

…becomes…

Step 1	→	Step 2	→	Step 3	→	Step 4	→	Semi-finished	→	"Finished"

Defined and Empirical Processes

"It is typical to adopt the <u>defined</u> (theoretical) modeling approach when the underlying mechanisms by which a process operates are reasonably well understood.

When the process is too complicated for the defined approach, the <u>empirical</u> approach is the appropriate choice."

<u>Process Dynamics, Modeling, and Control</u>, Ogunnaike and Ray, Oxford University Press, 1992

What makes Agile work?

- Better collaboration with business
- More adapted to change/learning
- Communication (f-t-f)
- Motivation
- Doing Less
- Dispersed ownership
- Time boxes

- Inspect & adapt
- Focus on the real thing
- Three heads are better than one
- Collocation
- Information radiators
- Short feedback loops
- Team autonomy
- Accepted Responsibility

These vary (eye of beholder, project, team).

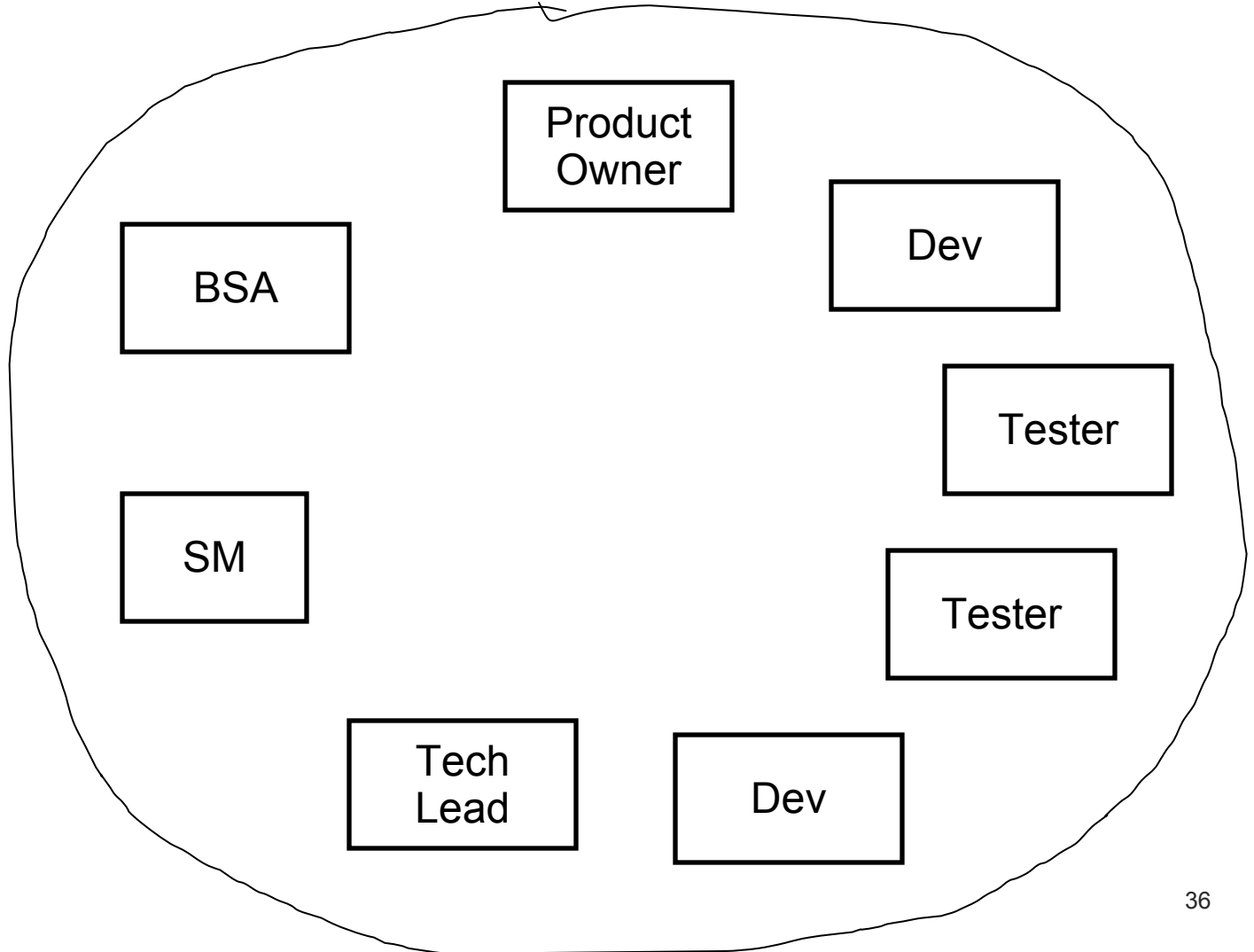# What is he most important thing for you to do in Agile?

Think for yourself.

Some ideas, process, principles and practices can help, but the Team <u>together</u> has to make the project succeed.

Typical Core Team

The
Committed

(Pigs)

With all
their
human
attributes
and rough
edges

Product
Owner

Dev

BSA

Tester

SM

Tester

Tech
Lead

Dev

No silver bullet (ugh!)

"Scrum is sometimes thought of as a silver bullet; implement it and your development problems will melt away.

This is not the case.

Implement Scrum and all of the reasons that an organization has trouble delivering quality software on schedule are thrown up in your face, day after day, month after month - made obvious and critical by Scrum. Scrum is just an engine for development, not a solution to problems.

Scrum helps you identify, triage, and solve the problems, but the difficult work of solving the problems remains yours."

Ken Schwaber

Short Bibliography

- *The New New Product Development Game. Harvard Business Review, 1986 (January-February).* Takeuchi & Nonaka
- *Agile Project Management with Scrum*, Schwaber
- *Agile Software Development with Scrum*, Schwaber and Beedle
- *Lean Software Development,* Mary and Tom Poppendieck
- *Agile Software Development,* Alistair Cockburn
- *Lean Thinking*, Womack and Jones
- *Extreme Programming Explained*, 2nd Ed., Kent Beck
- *Agile Project Management*, Jim Highsmith
- "Want Better Software?" article by Mike Cohn
- *www.agilealliance.com*
- *www.controlchaos.com*
- *www.martinfowler.com/articles/newMethodology.html*
- *www.poppendieck.com*
- *Google "Agile Methods" : 1,700,000 hits*

Lean-Agile has serious intellectual, experiential and empirical underpinnings. And Agile is CMM-I compliant.

What's wrong with Scrum?

- Must deal with the tyranny of waterfall
- Often counter-intuitive
- Too much truth (we have to admit we are imperfect)
- Old roles change, esp for middle managers
- Too much change - disruptive

When would I not use Scrum?

- If the people in the team or around the team were very resistant to Scrum.

# When to choose Scrum initially?

- More change
- High priority effort (1 reason: remove impediments)
- Willing & a good team
- Willing & engaged business
- A decent ScrumMaster and Product Owner

Win with Scrum

- Good teams can become 5x – 10x more productive

- Bring courage, willingness to learn from fast failures; and build trust

- Relentlessly pursue perfection

Contact Info

Joe Little
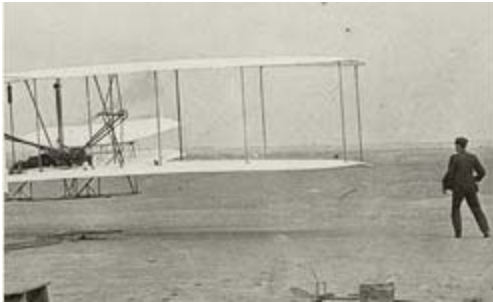
Lean & Agile coaching & training

Kitty Hawk Consulting, Inc.

Charlotte, NC

O: 704-376-8881

M: 917-887-1669

jhlittle@kittyhawkconsulting.com

Agile & Business blog:

http://agileconsortium.blogspot.com

http://kittyhawkconsulting.com

http://leanagiletraining.com

"TURNING IDEAS INTO REALITY"