ADAPTing to Agile
for
Continued Success

Agile 2010 – Orlando
August 13, 2010

Mike Cohn

mike@mountaingoatsoftware.com

Expectations

The OO-Agile connection

- Agile is the second wave of objects
  - This is why so many Manifesto authors were early OOD / OOP advocates:
    - Kent Beck
    - Alistair Cockburn
    - Ward Cunningham
    - Bob Martin
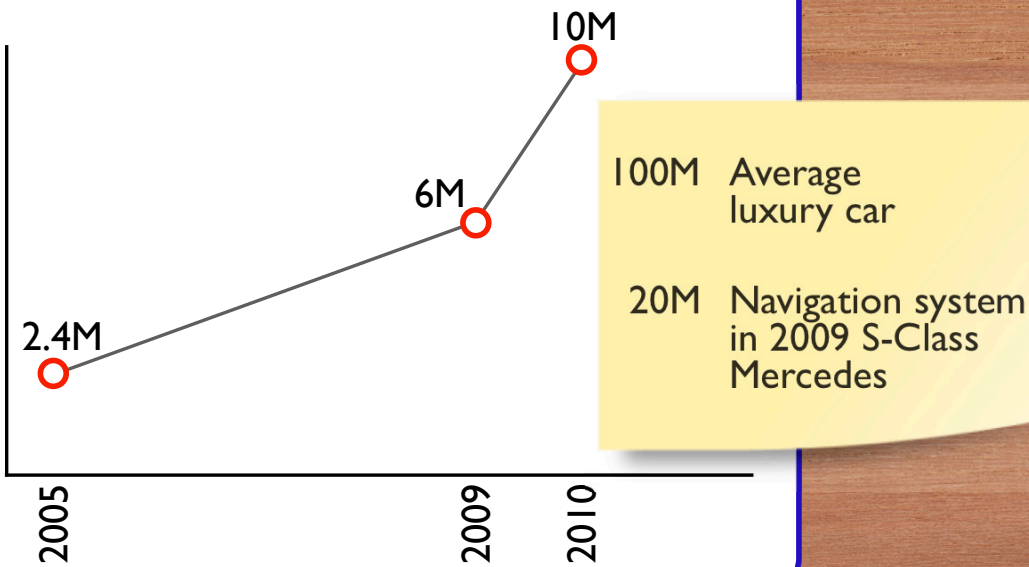   - Dave Thomas
    - etc.

Looking back at OO

- Some said the OO revolution failed because:
  - Objects didn't achieve their full purpose
  - Being enamored with objects prevented us from looking for even better ways to build software
  - Businesses were led to expect miracles
- Yet: Could today's applications be written without objects?

Lines of Code in Ford Vehicles

10M

6M

2.4M

2005   2009   2010

100M   Average luxury car

20M   Navigation system in 2009 S-Class Mercedes

What expectations do we have of agile today that could lead us to look back on it and call it a failure?

1 Agile teams are 5–10x better.

2 There is a certain set of things you need to do or you aren't agile.

3 Becoming agile is easy.

Expectation #1: Agile is 5–10x better

- Very few teams have achieved the gains that some teams have shown are possible

- We can point to 5x and 10x success stories
  - These are the companies that did everything right
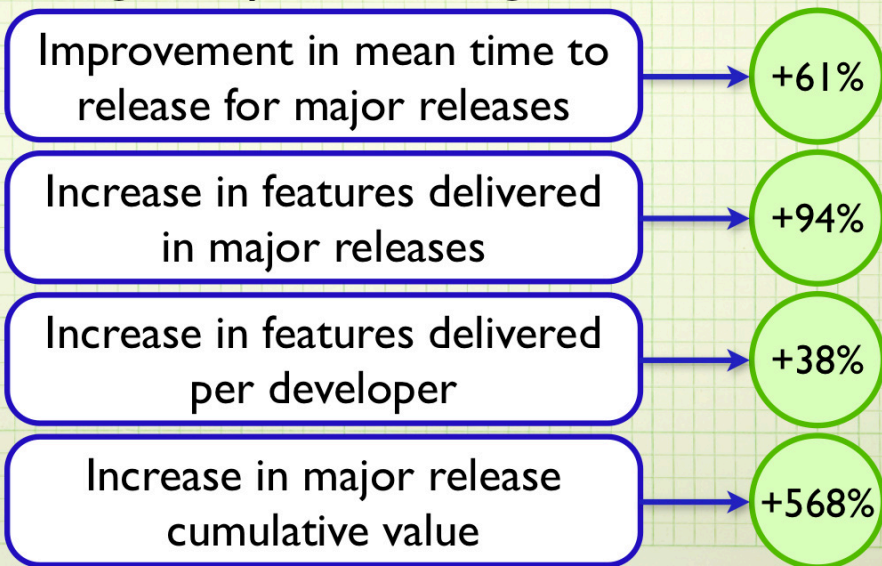- But these aren't the norm

Salesforce.com

- During first year of using Scrum

Improvement in mean time to release for major releases	+61%
Increase in features delivered in major releases	+94%
Increase in features delivered per developer	+38%
Increase in major release cumulative value	+568%

Source: Scrum Gathering, April 16, 2008.

ePlan Services

Productivity (NCSS / month)

US average	270
Three years prior to introducing agile	389
First nine months after starting agile	1206

Defects per KNCSS

Three years prior to introducing agile	10
First nine months after starting agile	2.9
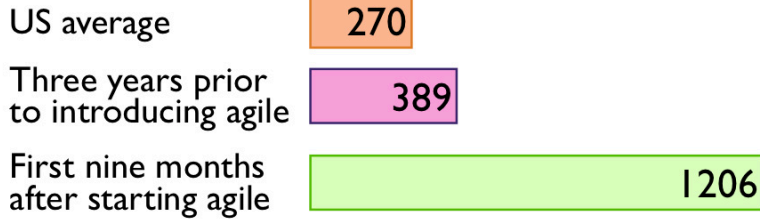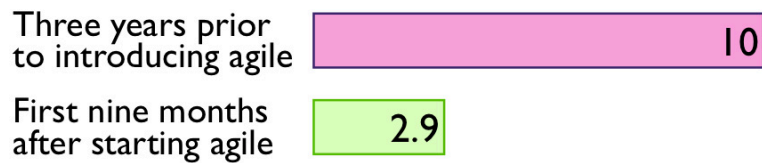
Cosmodemonic Biotech

	Waterfall	Agile
Use Case pages	3,000	
User Stories		1,400
Calendar months	9	12
Person months	540	54
Lines of Java code	58,000	51,000
Lines of Java code per person-month	120	840

Improvements using Extreme Programming

	Pre-XP	XP	Improvement
Cost	$2.8 million	$1.1 million	61%
Schedule	18 months	13.5 months	24%
Defects	2,270	381	83%
Staffing	18	11	39%

Source: "10 Tips for Successful Agile Transitions," Joshua Kerievsky, QCon 2007.
At http://www.infoq.com/presentations/10-tips-for-agile-transitions

Expectation #2: Agile is "this" set of things

- Many teams miss the mark
  - Either don't do "all" of agile or don't do it well
- "We do Scrum, but…"
  - "we do 3-month sprints."
  - "we don't have anything shippable at the end of each sprint."
  - "we do testing in a separate sprint."

It's not about strict adherence

Inspect and Adapt

- Meet daily
- Test first
- Integrate continuously

© Copyright Mountain Goat Software®

Expectation #3: Doing agile is easy

- "Mastery" in two days

- Most claims are realistic

  - "Although agile processes promise greater productivity once in place, productivity may decrease during the transition while the team learns new techniques." [1]

  - "Maybe we need to say, 'Agile is hard, and you can't master it by sitting through a two-day course.'" [2]

- But there are some promises of miracles:

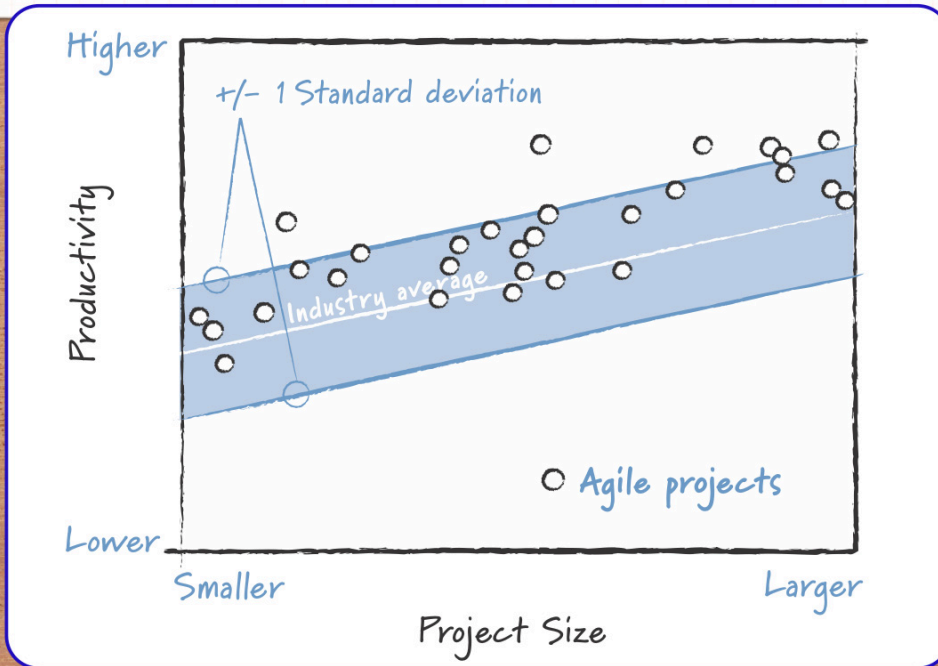 - "Current research shows that any team can achieve hyper-productivity in a few sprints, even in a dysfunctional company." [3]

[1]Cohn and Ford, "Introducing an Agile Process to an Organization," *IEEE Computer*, June 2003.
[2]James Shore, http://jamesshore.com/Blog/The-Decline-and-Fall-of-Agile.html
[3]Jeff Sutherland, http://scrumjeffsutherland.blogspot.com/2009/12/jeff-sutherland-google-dec-14-2009.html.

© Copyright Mountain Goat Software®

So let's see what the truth is

Higher — Lower / Productivity

+/− 1 Standard deviation

Industry average

○ Agile projects

Smaller — Larger / Project Size

Higher — Lower / Time-to-Market

○ Agile projects

+/− 1 Standard Deviation

Industry average
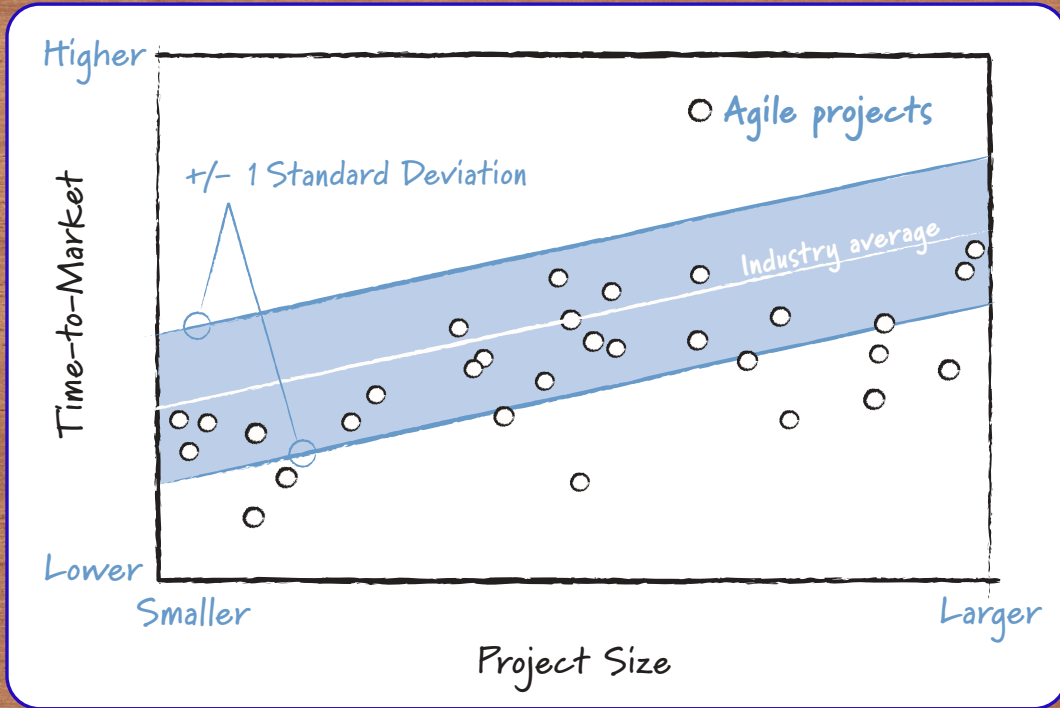
Smaller — Larger / Project Size

Will agile continue to "fail" against these expectations?

- Of course, if that means
  - Not all companies become 5–10x better
  - Being done half-heartedly in some places
  - It will continue to be hard to do
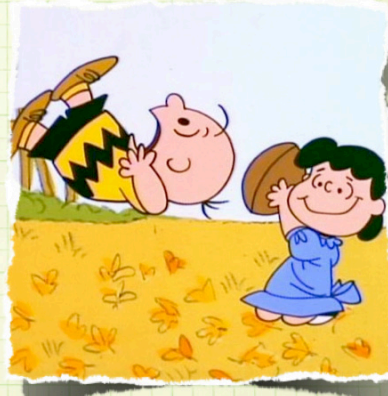
But these are inappropriate measures of success

Agile is not something you become, it's something you become more of.

Becoming more than you are
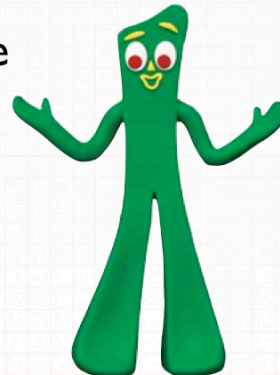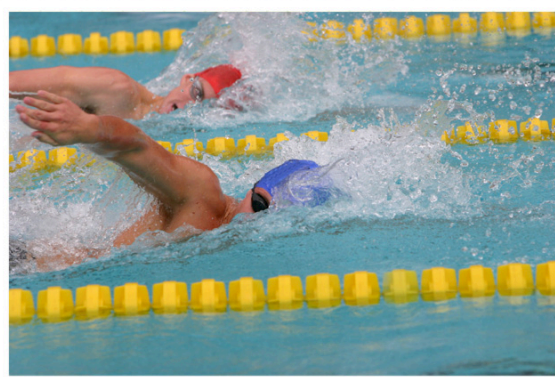
- I'm **more** flexible than I used to be
- You're **more** flexible than I am
- She **is** flexible

© Copyright Mountain Goat Software®

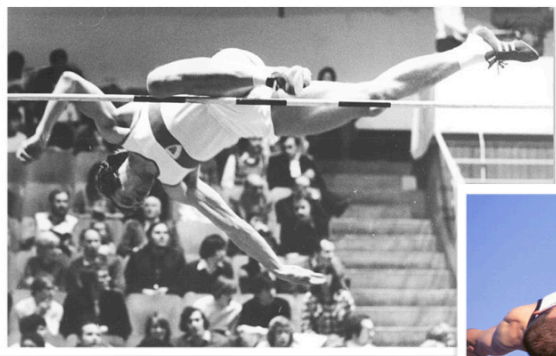# Succeeding with agile

- Success in agile is about becoming better than we were
  - Stop obsessing over being perfect

© Copyright Mountain Goat Software®

How do we get there?

- We raise the bar on one another
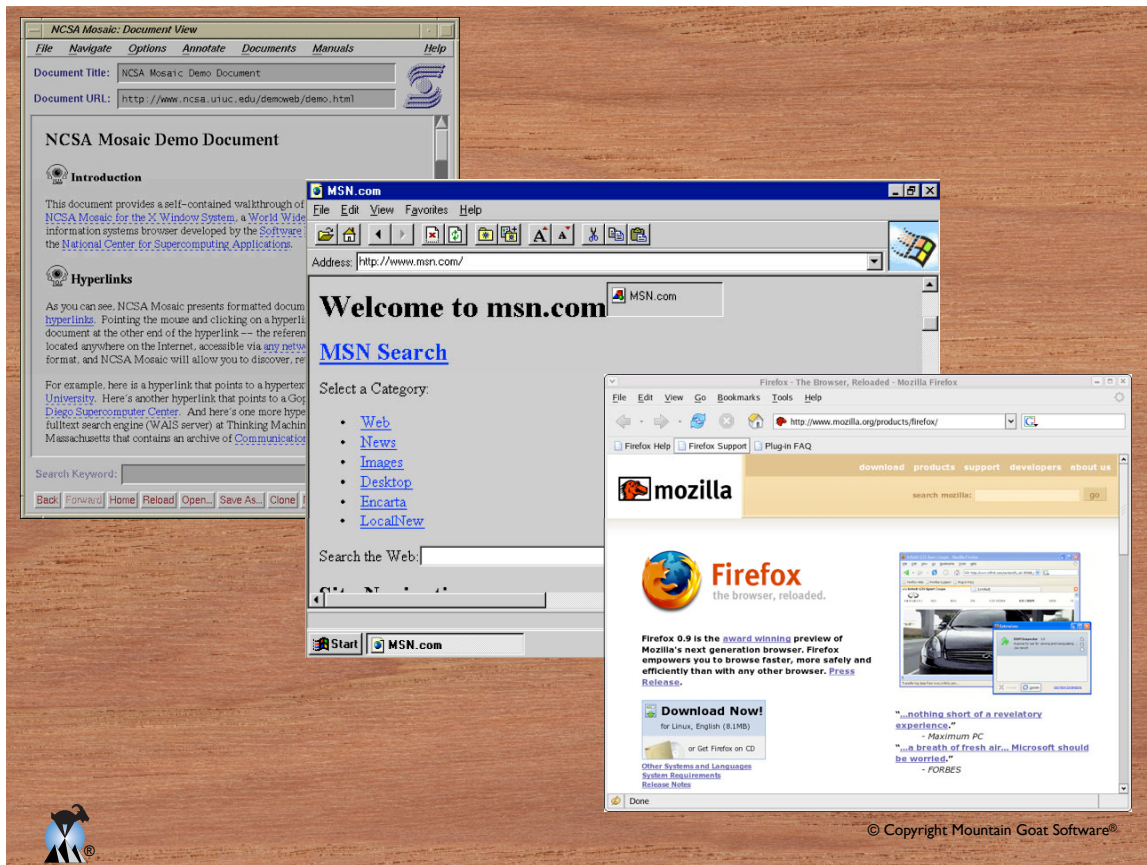
Raising the bar—slowly but steadily

© Copyright Mountain Goat Software®

Continuing to raise the bar on one another

© Copyright Mountain Goat Software®

This idea used to bother me

It's still about continuous improvement

A Awareness that there is room for improvement

D Desire to change

A Ability to work in an agile manner

P Promote early successes to build momentum and get others to follow

T Transfer the impact of agile throughout the organization so that it sticks

---

"I'm the ScrumMaster and ...

...the developers are not meeting expectations for code quality.

One of our challenges is that we're still hacking our way through lots of legacy code that isn't unit-testable or automated yet. This code is mission-critical and the person who has been working mostly on that area of code consistently leaves holes in the design and implementation of new pieces of that code.

How can I get her to write high-quality code with automated tests?"

Individual and group change

- All individuals will need to move through the Awareness, Desire, and Ability stage
  - But will do so at different rates
- Early adapters and leaders:
  - Use the Promote stage to build Awareness and Desire in later adopters
  - Need to Transfer the impact of agile to groups like Human Resources or the transition will fail

Tools for building...

Awareness

- Communicate that there's a problem
- Use metrics
- Provide exposure to new people and experiences
- Focus attention on the most important reason or two for changing

Tools for building...

Desire

- Communicate that there's a better way
- Create a sense of urgency
- Build momentum
- Get the team to take agile for a test drive
- Align incentives (or, at least, remove disincentives)
- Focus on addressing any fears
- Help people let go
- Don't discredit the past
- Engage everyone in the transition

Waterfallacies

- Mistaken beliefs or ideas about agile created from working too long on waterfall projects.

Examples

- Agile ignores architecture, which would be disastrous for the type of system we build.
- Agile is OK for simple websites, but our system is too complicated.
- Our team is spread around the world, and agile requires face-to-face communication.

Ability

- Provide coaching and training
- Hold individuals accountable
- Share information
- Set reasonable targets
- Just do it

Promote

- Publicize success stories
- Host an agile safari
- Attract attention

Transfer

- Transfer the effects of agile beyond the current group
  - A team transfers to its department
  - A department transfers to its division
  - etc.
- If you don't transfer, the transition will eventually and inevitably fail
  - Too much organizational gravity pulling us back toward the status quo
- Example:
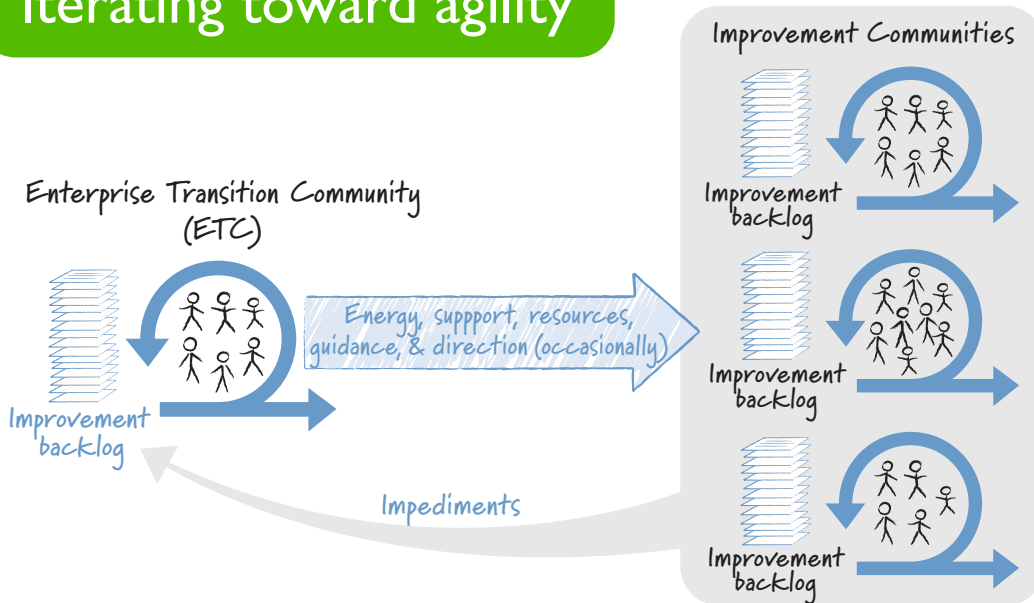 - If you don't align promotions, raises, annual reviews, those will work against you

Iterating toward agility

Iterating toward agility

Improvement Communities

Enterprise Transition Community (ETC)

Improvement backlog

Energy, suppport, resources, guidance, & direction (occasionally)

Improvement backlog

Improvement backlog

Improvement backlog

Impediments

Enterprise Transition Community (ETC)

- Creates a culture in which passion and desire to improve can thrive
- Does not direct the transition effort
    - Provides energy, resources, support and guidance
    - Removes organizational impediments to agility
- Encourages Improvement Communities to form

ETC members

- Sponsor
    - From highest level at which change is supported
    - Not a checkbook-only commitment
- Others
    - From any level but driven by desire to improve
- Disbands when the "transition" part of adopting agile is over

ETC responsibilities

- Articulate the reasons for adopting agile
- Stimulate conversation
- Provide resources
- Engage everyone
- Set appropriate aspirations
- Anticipate and address people issues and other impediments
- Encourage simultaneous focus on practices and principles

An ETC's improvement backlog

Item	Who	Note
Create an "Agile Office" where teams can get help.		Jim (CTO) to talk this up at monthly development meeting. Let's see if there's any interest.
Establish an internal program for developing ScrumMasters.		How do we identify good internal candidates? How do we develop them?
Collect and disseminate Scrum success stories in our company.	SC	Savannah has expressed interest in this.
Resolve dispute with facilities over rearranging second floor cubicles.	JS	Jim to talk to Ursula in facilities
Get more teams to do continuous integration.	AR	Arie will summarize metrics from his project and see how many teams he can motivate.

Improvement communities (ICs)

- Form around the passion of a small number of people
  - Expand from there
- Do the real work of improving how the organization implements agile
- Focus on goals with practical relevance
- Examples:
  - ScrumMaster, Testing, Product Owner, Continuous Integration

Working with an IC

- An IC works with a project team
- Work is not done in an ivory tower
- Most ICs work in 2–4-week iterations
- Disband or refocus when goal has been achieved

ETC Improvement Backlog

...

Establish an internal program for developing ScrumMasters.

...

Not everything on an IC's improvement backlog needs to tie back to the ETC's backlog

An IC Improvement Backlog

Figure out how to identify good candidates to become ScrumMasters (in addition to those who ask to participate in this program).

Establish an internal mentoring program.

Develop some internal classroom training. Which courses? Who can teach them? Can we license courses?

Get budget for next year for external coaching. How many days? At what expected daily rate?

---

1 Go back to the office and create an improvement backlog

2 Create communities or not but find passionate people to attack improvement backlog items
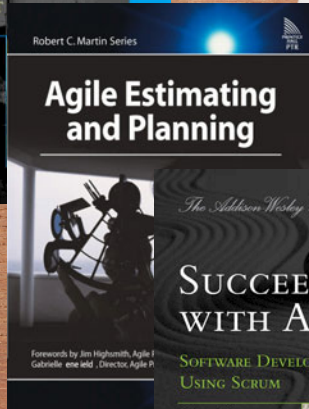
3 Return here next year more agile than you are today

4 Let's see how soon we can make it "software development" instead of "Agile Software Development"

Mike Cohn

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

twitter: mikewcohn

(888) 61-AGILE

MOUNTAIN GOAT
S O F T W A R E