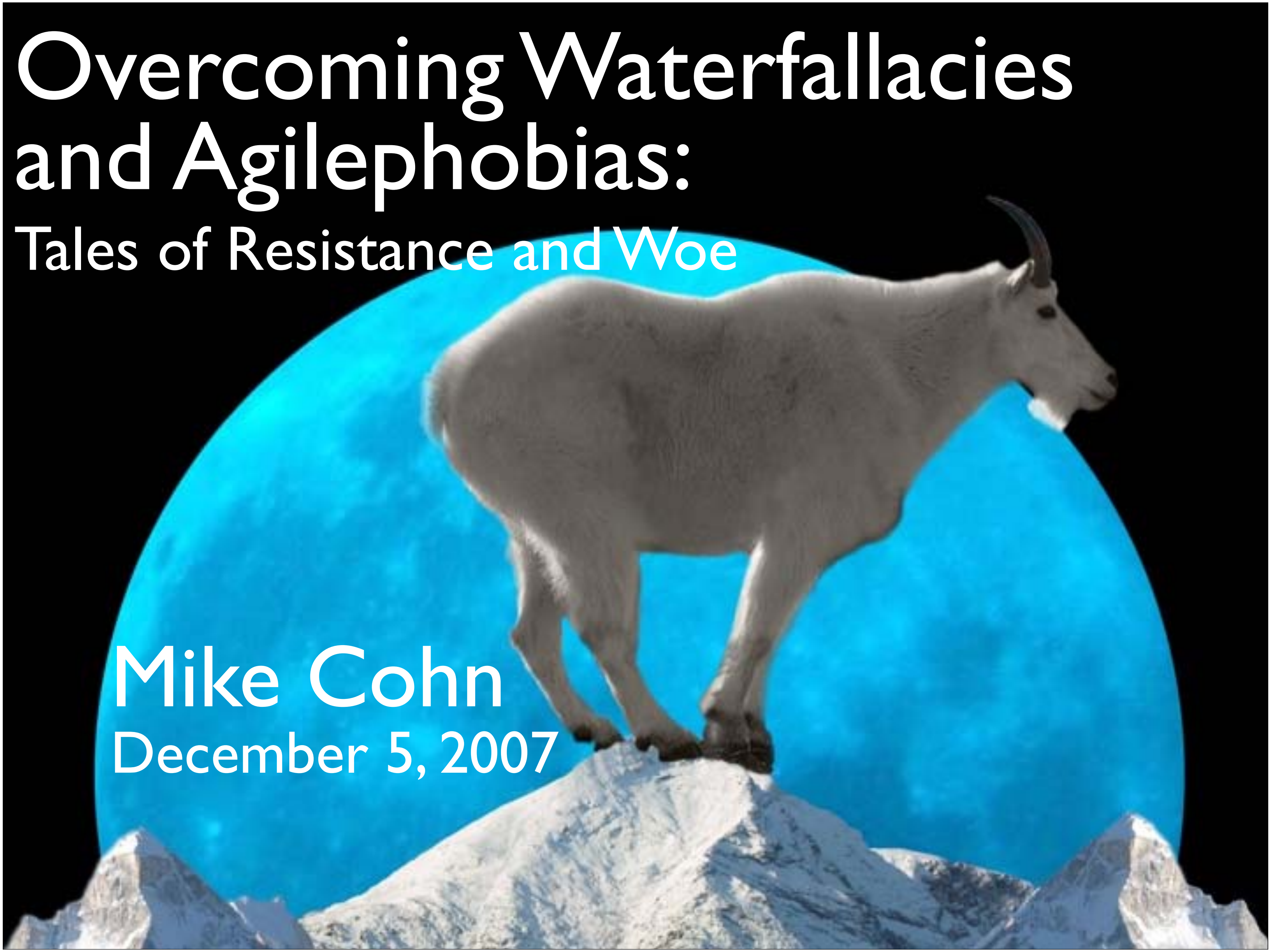
Overcoming Waterfallacies and Agilephobias:

Tales of Resistance and Woe



Mike Cohn

December 5, 2007

Topics today...

1. Waterfallacies
2. AgilePhobias
3. Overcoming resistance

Waterfallacies

Waterfallacy

noun.

mistaken belief or idea about agile that stems from prolonged exposure to waterfall projects

1

Agile can’t give me a firm date and a commitment to what will be in the product.

Is the answer just to under-commit and then deliver more than expected?

Knowing the scope and date

- It’s always been impossible to know both the scope and the date of a project
  - The vast majority of waterfall projects get this wrong, too
    - And then end up delivering too little of the wrong functionality
- Outsourcing companies that promise to fix both do so by either padding or lying

If we...

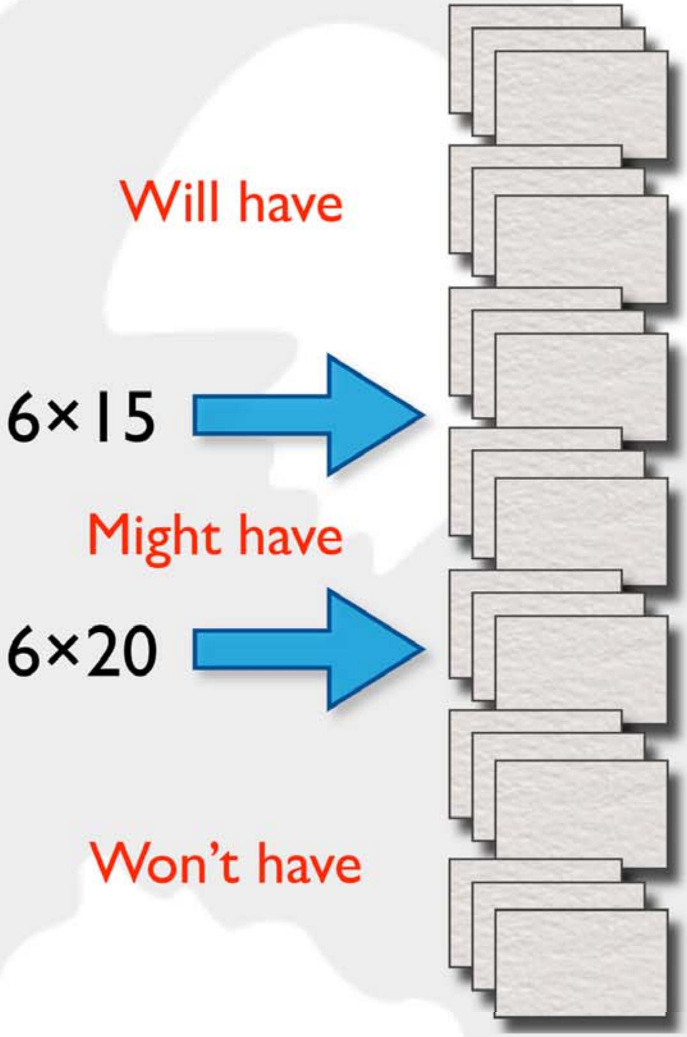
1. acknowledge the impossibility of guaranteeing functionality and date
2. and view fixed scope or fixed date projects as risk management

Then we...

- Can write fixed price or fixed contracts
- And use the same principles on non-contracted projects that have a similar need for a low risk of dropped scope or schedule overrun

Fixed-date planning: an example

Desired release date	30 June
Today's Date	1 January
Number of sprints	6 (monthly)
Low velocity	15
High velocity	20



Will have

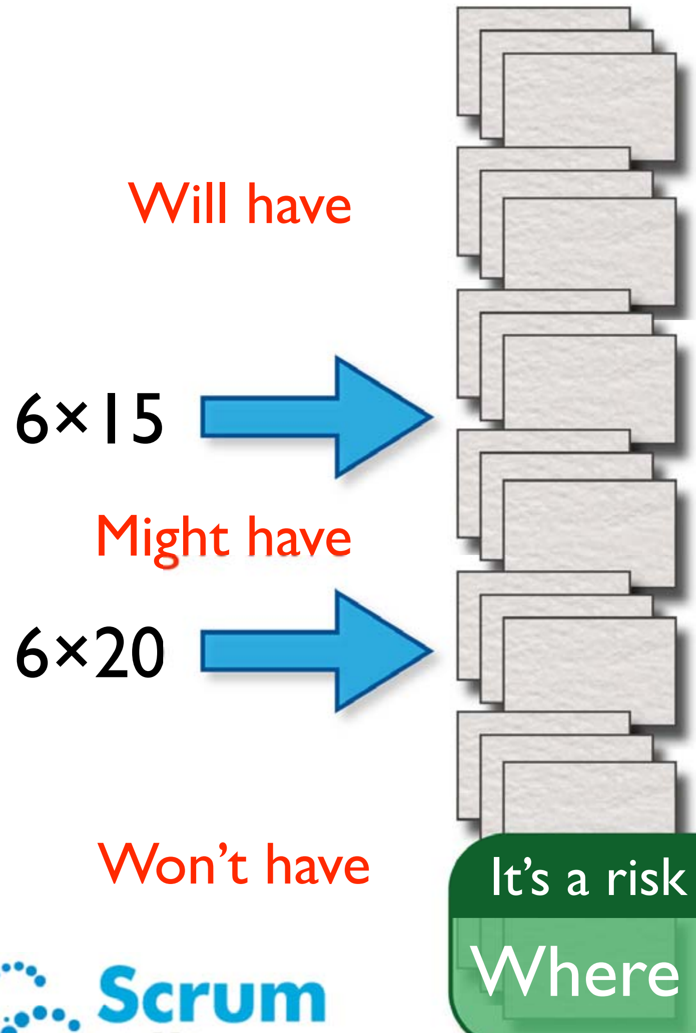
6×15

Might have

6×20

Won't have

Fixed-date contracting



If you write a contract for just the *will haves*:

- You won’t likely win the contract
- But you’ll probably make money if you do

If you write a contract that includes the *might haves*:

- You will likely win the contract
- But probably not make money on it

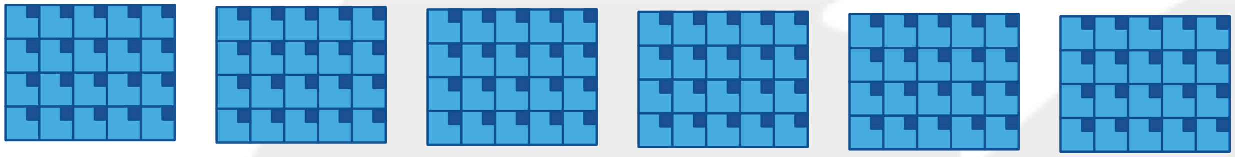
It’s a risk issue

Where do you want to be?

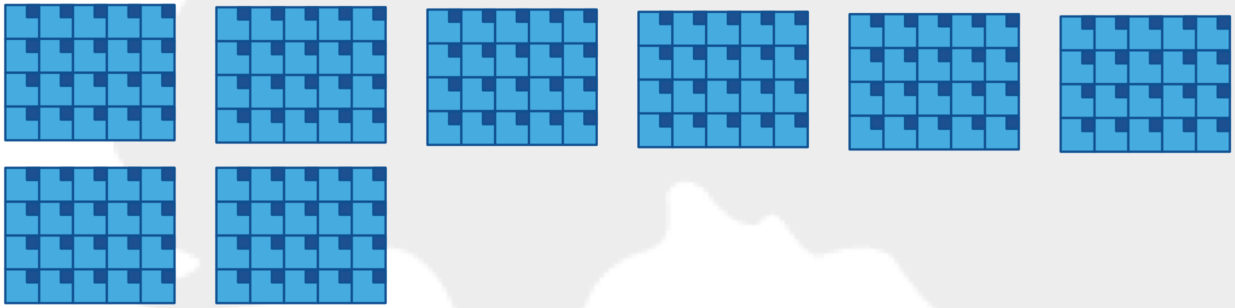
Fixed-scope planning: an example

Total story points desired	120
Low velocity	15
High velocity	20

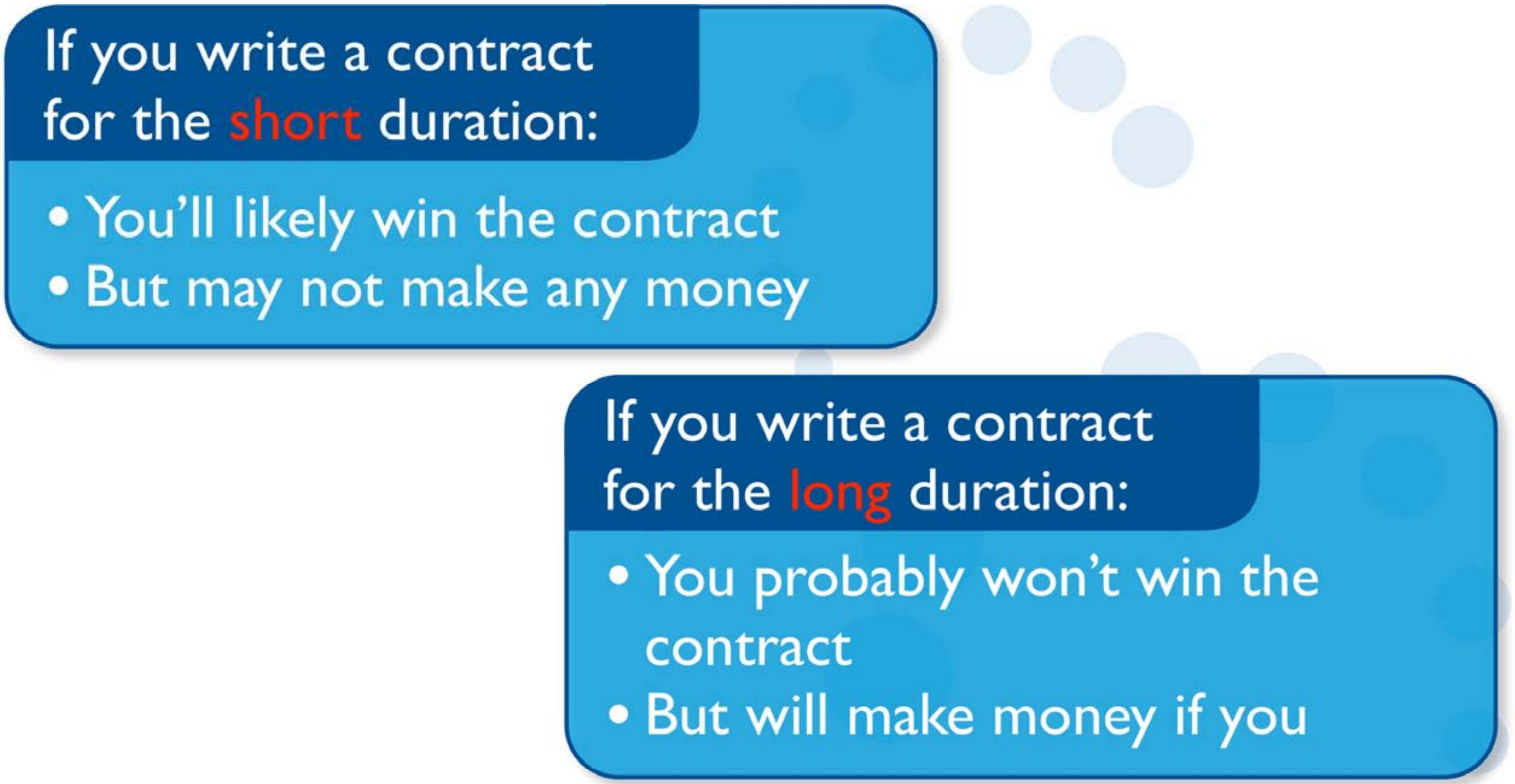
120÷20=



120÷15=



Fixed-scope contracting



If you write a contract for the short duration:

- You'll likely win the contract
- But may not make any money

If you write a contract for the long duration:

- You probably won't win the contract
- But will make money if you

It's a risk issue

Where do you want to be?

The value of customer feedback

- Agile does make fixed-price and fixed-scope projects harder
  - Because it acknowledges
    - the importance of customer feedback
    - and the existence of emergent requirements
- Sequential processes treat these as changes of scope

2

Agile requires our testers to be involved and testing right from the start. But they're tied up on other projects.

We can't move to agile because we can't move testers from these other projects.

Agile is not a silver bullet

- Doesn’t solve a single problem you have, only exposes the problems you do have
- Hopefully exposes these problems to those who can solve them
- By repeatedly solving the problems you identify, things improve
- Agile is hard work
  - There will be days you wonder why you’re doing this

Digging out of this problem

- If others on the first projects can help test, have them do so
    - Perhaps just enough to move some small number of testers (1?) to the new project
- Have the new project run cross-functionally
    - Whole team does whatever it takes to pay off the testing debt
- Rule from day one: On new projects don't allow any new debt to build up

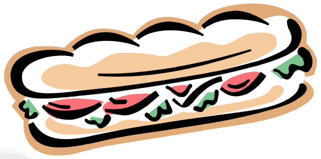
3

Agile requires everyone to be a generalist.

That isn’t cost-effective and doesn’t make sense for my project. Some of what we do is highly specialized.

Must everyone be a generalist?

- Every deli in the world has figured specialization out
  - Why can't we?



- Not everyone needs to be a generalist
  - But it's very beneficial to have some cross-skilled people; "generalizing specialists"
  - This happens naturally on most agile teams because of their cross-functional composition
- Some domains require specialists
  - bioinformatics, video game development, etc.

4

Agile puts a lot of emphasis on unit testing. I'm not working on a legacy application (yet!) but we are nine-months in and haven't made automated testing a priority.

Can we still be agile?

Agile without a lot of tests in place

- Two most important principles of agile:

1. Inspect and Adapt

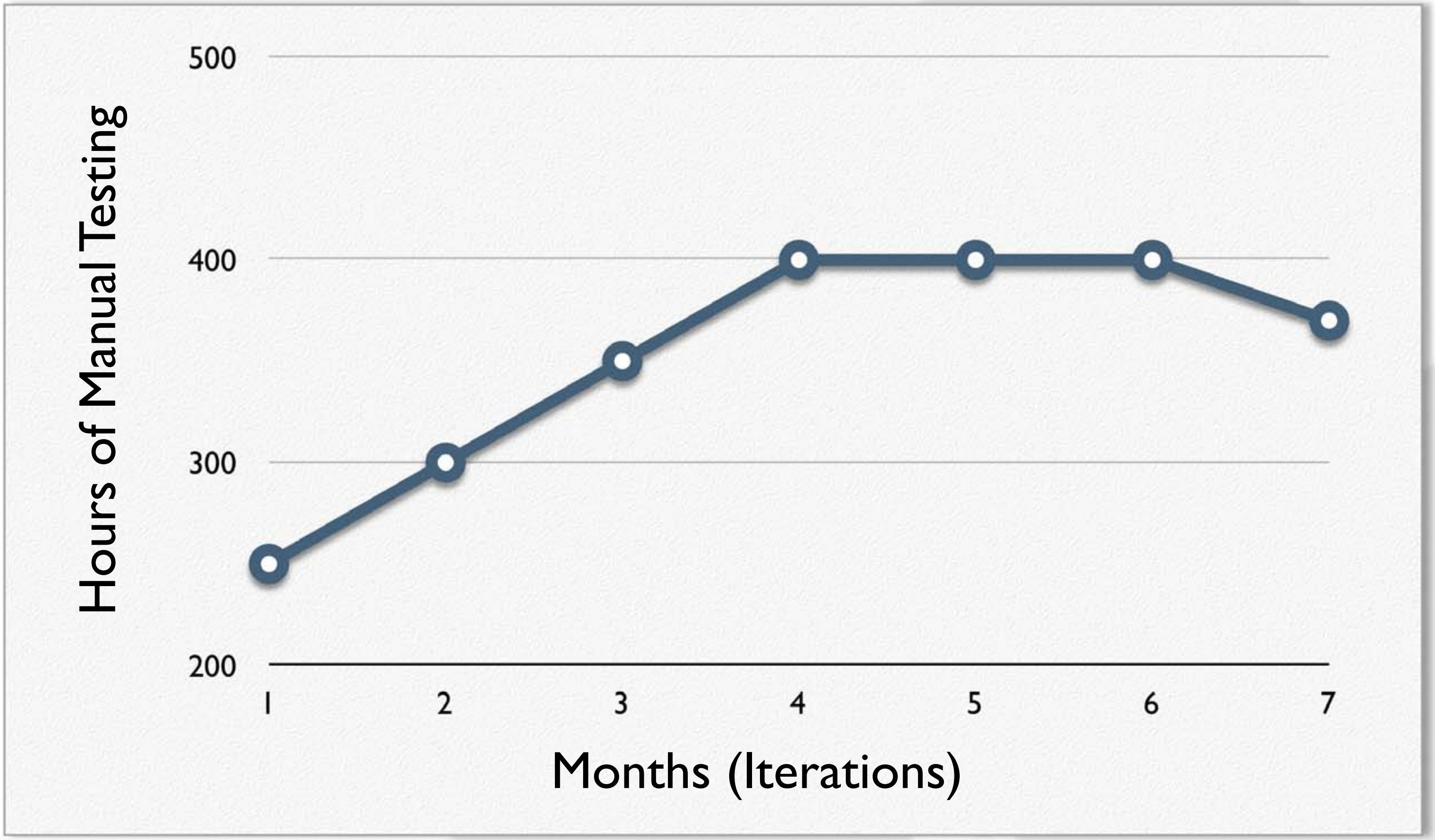
  - Start where you are, do something, see how you did

2. Iterate

  - And then do it again until you get to where you want to be

A three-step process

1. We end this iteration with less manual testing than we started.
   Go after the "low-hanging fruit" by automating some easy-to-automate tests of great value.
2. All new features come with automated tests.
3. Everything (maybe, we hope) has an automated test.


Hours of Manual Testing
500
400
300
200
1
2
3
4
5
6
7
Months (Iterations)

5

We work in distributed environment, frequently with some team members in India, China, or the Ukraine. Self-organization seems to clash with some cultures.

How will agile development work in these environments?

Self-organization around the world

- I suspect it's true that that self-organizing teams in the different countries will organize differently
  - Cultural differences will influence but not prevent self-organization
- Self-organization relies on
  - **C**ontainer (in which to organize)
  - **D**ifferences (among people)
  - Transforming **E**xchanges
- Self-organization leads to more creativity, ownership, productivity, engagement

Agilephobia

noun.
strong fear or dislike of agile, usually due to the uncertainty of change

Some AgilePhobias

- I’m afraid I’ll have nothing to do
- I’m afraid I’ll lose my job
- I’m afraid people will see how little I actually do
- I’m afraid I won't be able to keep up
- I’m afraid I won’t be able to learn the new software
- I’m afraid this will mean hard work
- I’m afraid I’ll be fired if the decisions we make don’t work out
- I’m afraid of conflict and trying to reach consensus

More agilephobias

- I’m afraid I won’t get raises or promotions anymore
- Drat! There go my 3-hour lunches
- Drat! This means I can’t mosey in at 10:30 anymore
- Drat! This means I’ll have to actually think now
- Drat! This means I’ll actually have to talk to people now
- It’s just so much easier and safer when someone else tells me exactly what to do
- It’s just so much easier and safer when I can tell them exactly what I want them to do

Overcoming AgilePhobias

- Being aware of what people fear is half the challenge
- Acknowledge these agilephobias and over time work to have solutions for them
- Over time people will lose their fear of losing their job
  - But will likely still need revised job descriptions in an agile enterprise

Overcoming Resistance

Overcoming resistance

- Sell the problem, not the solution
  - No one wants a solution to a problem they don't (think they) have
  - Be open to hearing better solutions than you have
- Communicate why the change and why now
- Put team members in touch with customers
  - Let them hear the problems you are hearing
- Emphasize benefits of the change
- Help resisters find new roles

Dealing with a sense of loss

Actions you can take...

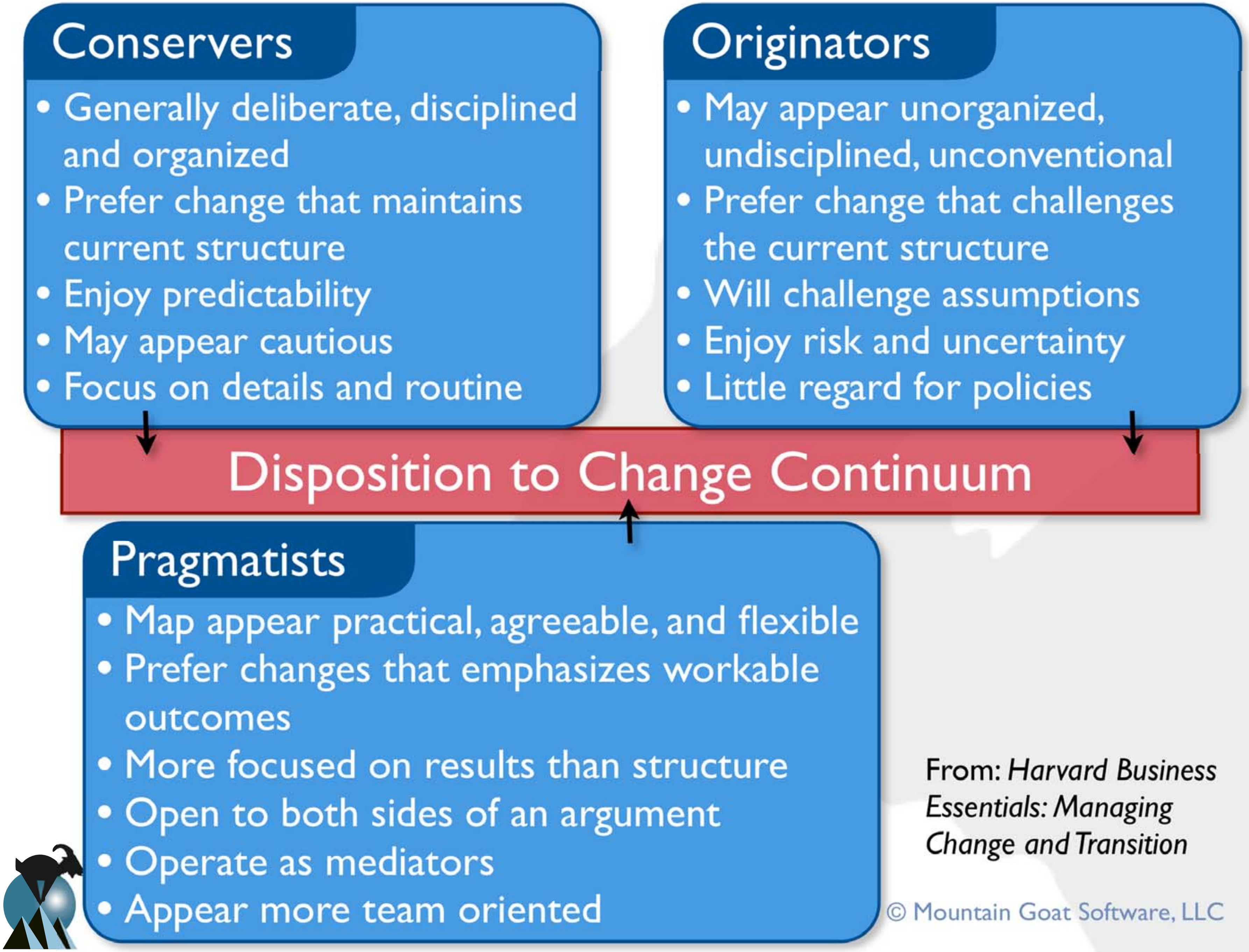
- Identify who's losing what
- Accept the reality and importance of subjective, personal loss
  - Don't argue with people about what they're losing or that they aren't really losing it
- Acknowledge losses openly and sympathetically
- Don't be surprised by overreaction
- Expect and accept the signs of grieving

Source: *Managing Transitions* by William Bridges

More actions for dealing with loss

- Define what's over what's not
- Mark the endings
- Treat the past with respect
- Remind people that the past they wish to hold on to was once new
- Overcommunicate

Source: *Managing Transitions* by William Bridges



Conservers

- Generally deliberate, disciplined and organized
- Prefer change that maintains current structure
- Enjoy predictability
- May appear cautious
- Focus on details and routine

Originators

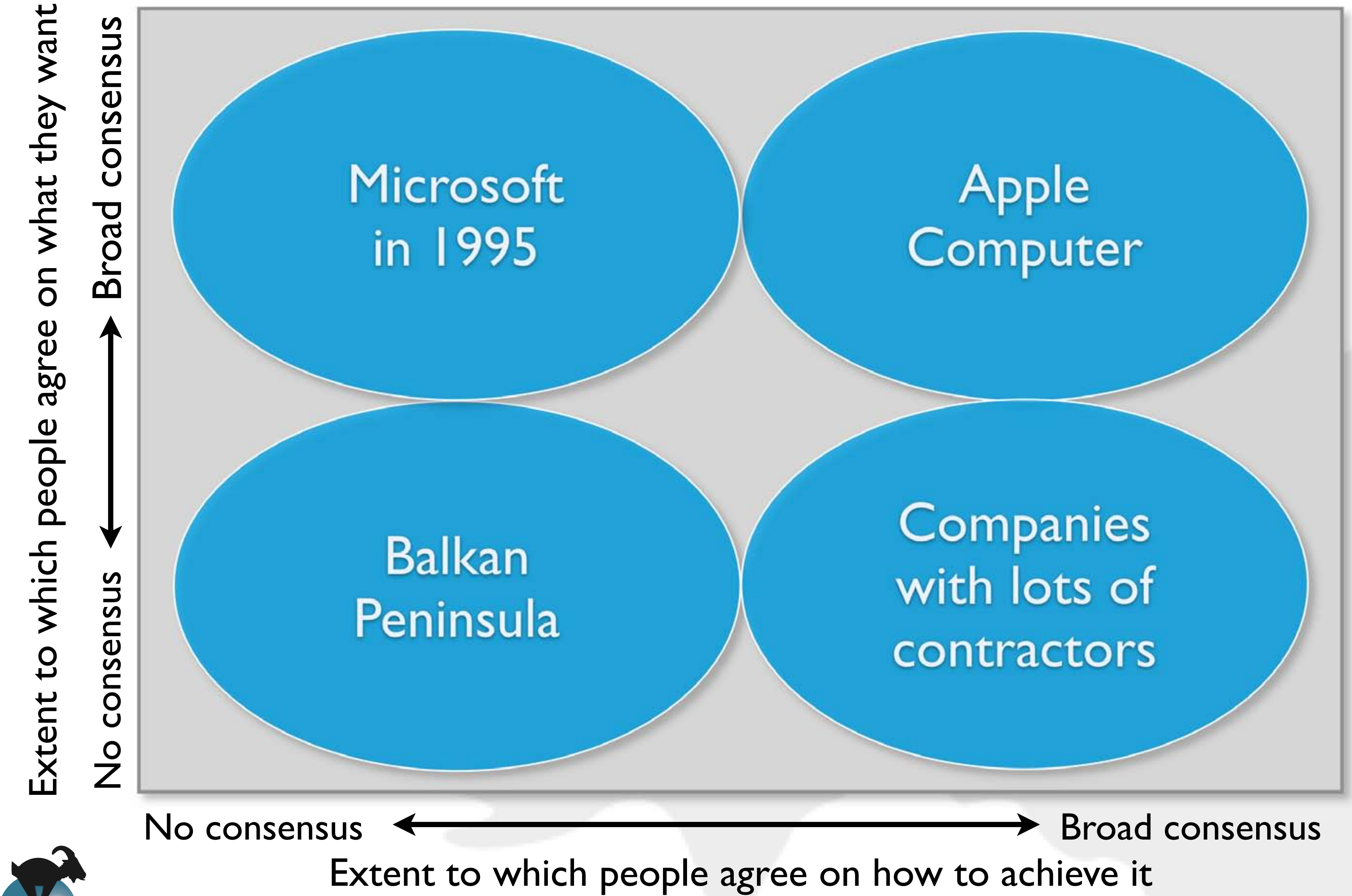
- May appear unorganized, undisciplined, unconventional
- Prefer change that challenges the current structure
- Will challenge assumptions
- Enjoy risk and uncertainty
- Little regard for policies

Disposition to Change Continuum

Pragmatists

- Map appear practical, agreeable, and flexible
- Prefer changes that emphasizes workable outcomes
- More focused on results than structure
- Open to both sides of an argument
- Operate as mediators
- Appear more team oriented

From: *Harvard Business Essentials: Managing Change and Transition*



Extent to which people agree on what they want

No consensus ↔ Broad consensus

Microsoft in 1995

Apple Computer

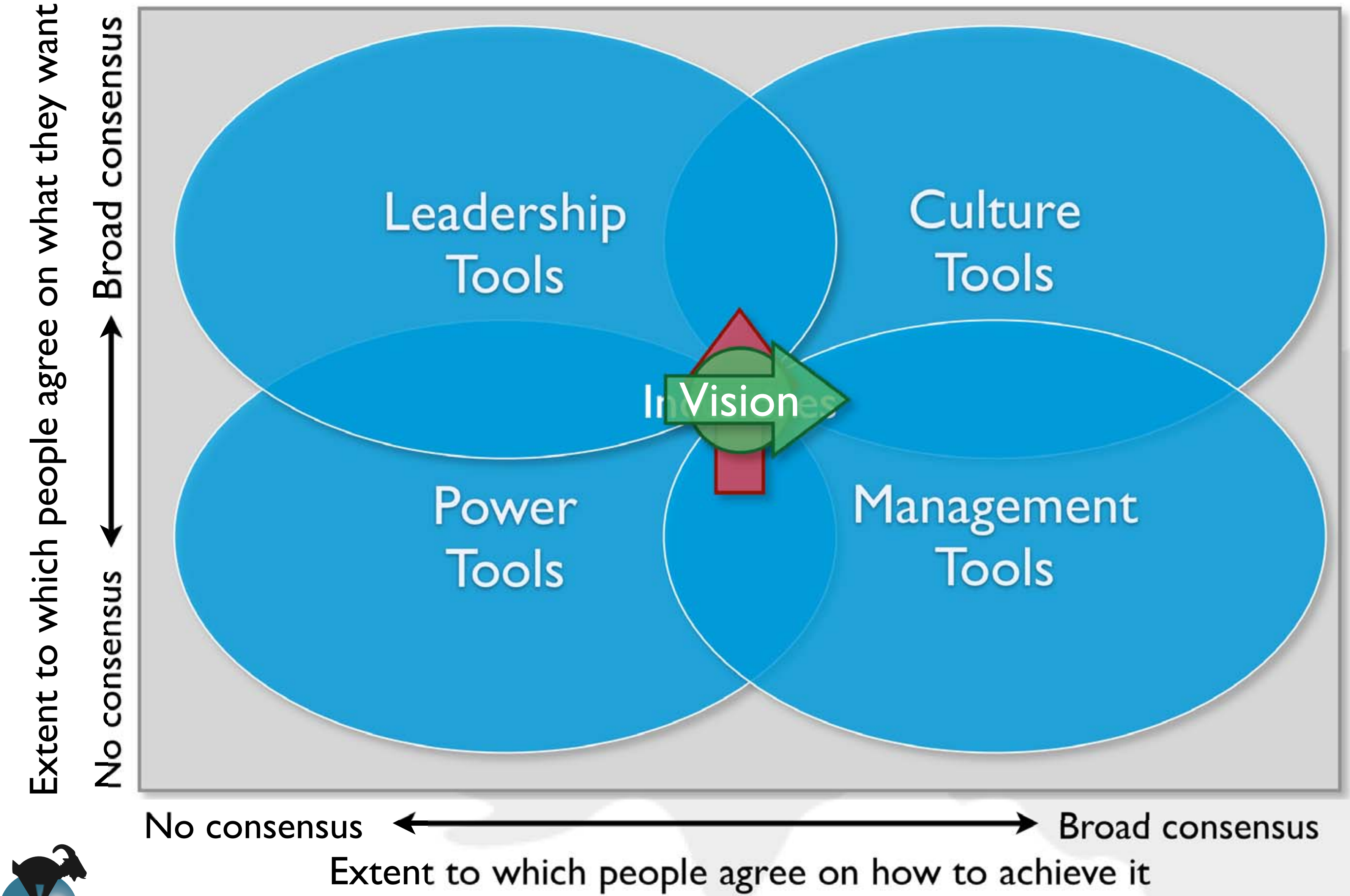
Balkan Peninsula

Companies with lots of contractors

No consensus ↔ Broad consensus

Extent to which people agree on how to achieve it

“Tools of Cooperation and Change,” Christensen et al., Oct. 2006.



Leadership Tools

Culture Tools

Power Tools

Management Tools

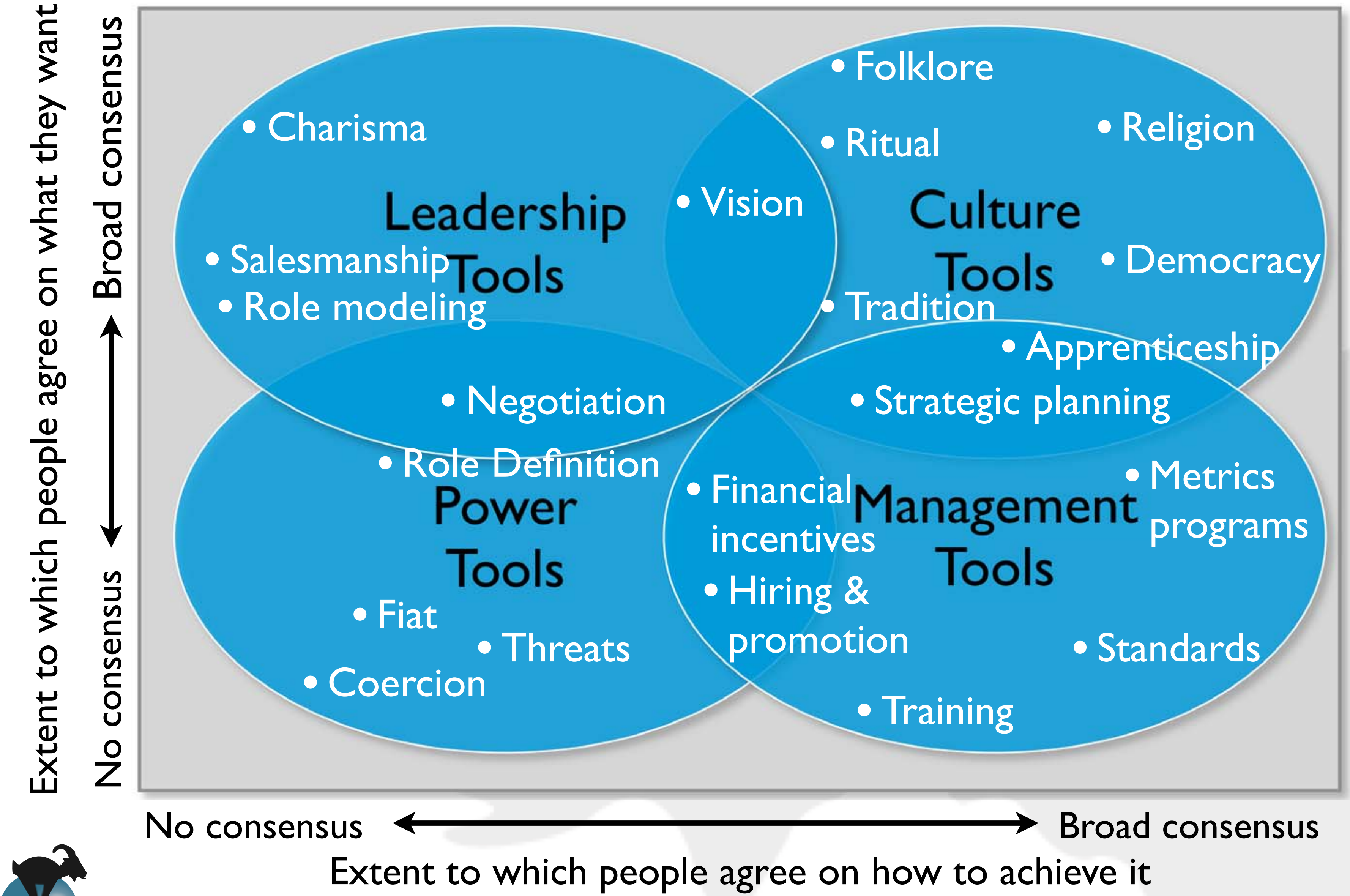
Vision

Extent to which people agree on what they want

No consensus ↔ Broad consensus

Extent to which people agree on how to achieve it

No consensus ↔ Broad consensus



Leadership Tools

- Charisma
- Salesmanship
- Role modeling
- Vision

Culture Tools

- Folklore
- Ritual
- Religion
- Democracy
- Tradition
- Apprenticeship

Power Tools

- Negotiation
- Role Definition
- Fiat
- Threats
- Coercion

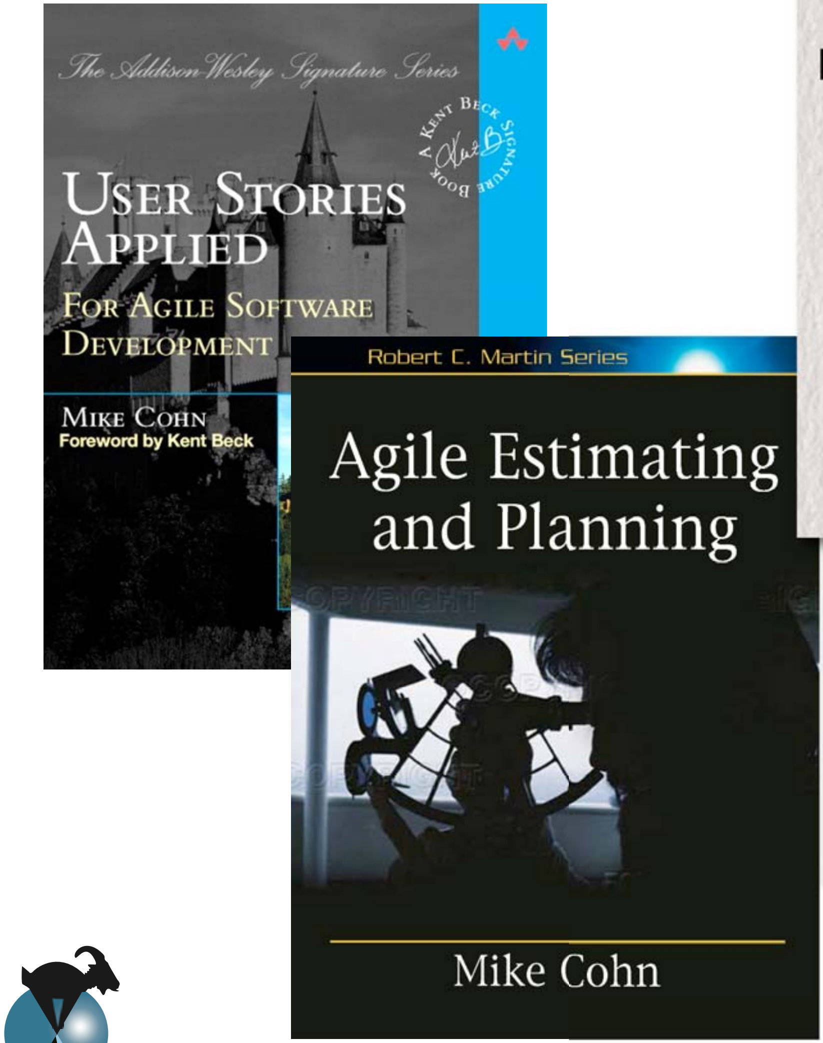
Management Tools

- Strategic planning
- Financial incentives
- Metrics programs
- Hiring & promotion
- Standards
- Training

Extent to which people agree on what they want: No consensus ↔ Broad consensus

Extent to which people agree on how to achieve it: No consensus ↔ Broad consensus

Mike Cohn contact info


The Addison-Wesley Signature Series
A Kent Beck Signature Book
User Stories Applied
For Agile Software Development
Mike Cohn
Foreword by Kent Beck
Robert C. Martin Series
Agile Estimating and Planning
Mike Cohn

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)

MOUNTAIN GOAT SOFTWARE