Transitioning to Agile

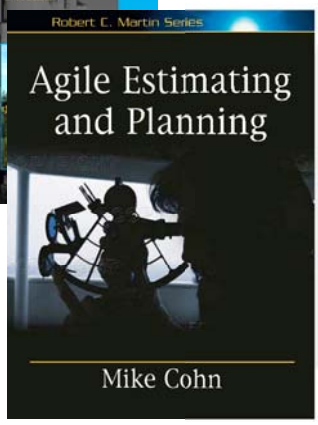
A Guide to Good Practices in Context

Mike Cohn
President
Mountain Goat Software
Lafayette, Colorado
mike@mountaingoatsoftware.com

SD WEST 2007

MARCH 19-23, 2007, SANTA CLARA, CA

Mike Cohn - background



Consultant, author, and speaker

- Founding member and director of Agile Alliance, Scrum Alliance, and Agile Project Leadership Network
- Founder of Mountain Goat Software

The twelve things you must do

1. Dispense with predictability
2. Treat the transition as a project
3. Use a congruent approach
4. Pick the right project
5. Get the right people on (and off) the project
6. Start development with a beachhead team
7. Overcome resistance
8. Have a customer
9. Engage the change agents
10. Reflect
11. Don't try it all at once
12. Follow a guide

1 Dispense with predictability

- How we traditionally view our organizations
  - Behavior is highly predictable
  - Once set in motion, will continue in motion
  - Predictable
- An organization change strategy can be mapped out:
  - Do this first, then that, then such and so
  - And we'll end up right where I predict

"This machine imagery [Newtonian view] leads to the belief that studying the parts is the key to understanding the whole. Things are taken apart, dissected literally or figuratively...and then put back together without any significant loss. The assumption is that the more we know about the workings of each piece, the more we will learn about the whole."

~Margaret Wheatley
in *Leadership and the New Science*

Is it top down or bottom up?

- Two simplistic views of transitioning to agile:
  - Top down
    - Powerful leader shares a vision
  - Bottom-up
    - A team starts and everyone else sees the benefits of the new approach
- But, transitioning to agile is neither top-down nor bottom-up
  - It's everywhere, all together, all-at-once

What we do on projects

- On projects we learn we cannot precisely anticipate:
  - our users' requirements
  - how long it will take to develop a feature or entire system
  - which design will be best
  - the set of tasks necessary to develop a feature
- So we devise alternative approaches
  - Rather than ask for upfront specs, we deliver partial solutions, solicit feedback, and repeat
  - Rather than design the whole system, we design incrementally and adjust based on what we learn
- We need to do the same for the transition effort

2 Treat the transition as a project

- Establish an "Agile Transition Team" ("Agile Adoption Team," etc.)
  - Who?
    - Sponsor—senior person responsible for success
    - Area managers or leads who can make it happen
  - Meet weekly
  - Run monthly iterations managing work from a Transition Backlog

Stocking the transition backlog

- Brainstorm needed activities within these broad categories
  1. Establish a sense of urgency
  2. Create a vision
  3. Communicate the vision
  4. Empower others to act
  5. Plan for, support and create short-term wins
  6. Consolidate improvements
  7. Inspect and adapt

Ref: John Kotter, *Leading Change*, 1996.

An example transition backlog

Decide how pervasive to go with Scrum—software development only or full company	All
Identify which issues Scrum can solve or help with.	DF
Set expectations that it will hurt.	MC
Understand how testing will fit within Scrum.	TC
Identify other groups/functions that will be affected.	MC, DF
Identify needed organizational changes—reporting, reviews, compensation, career paths, etc.	
Determine how we'll communicate progress	
Are there agile project management tools we should use?	

3 Use a congruent approach

Part of the move to agile is a move to self-organizing teams → Moving to self-organization requires self-organization

"You will self-organize!"

Pre-requisites of self-organization

Container
- A boundary within which self-organization occurs
  - Company, project, team, city, role, nationality

Differences
- There must be differences among the "agents" acting in our system
  - Technical knowledge, domain knowledge, education, experience, power, gender

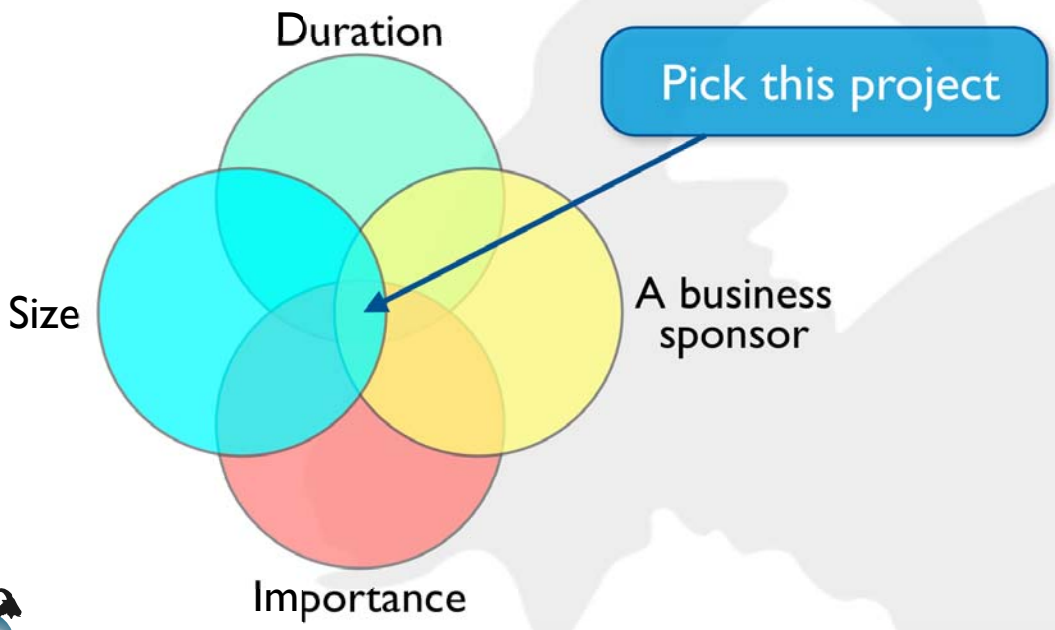
Transforming Exchanges
- Agents in the system interact and exchange resources
  - Information, money, energy (vision)

Glenda Eoyang: *Conditions for Self-Organizing in Human Systems*

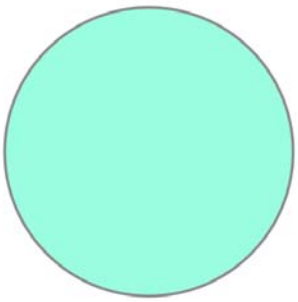
Using the CDE model

- When stuck thinking about how to nudge the organization think of the:
  - Containers
    - formal teams, informal teams, clarify (or not) expectations
  - Differences
    - Dampen or amplify them within or between containers
  - Exchanges
    - Insert new exchanges, new people, new techniques or tools

4 Pick the right project

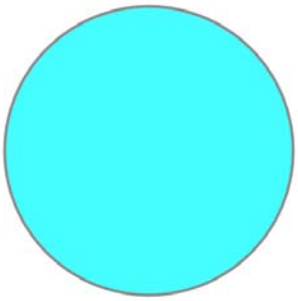


Size



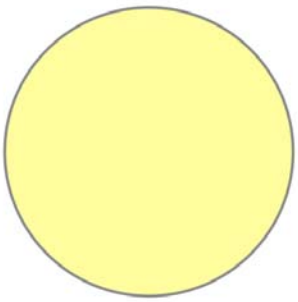
- Something that can be started with one team
  - Scale up the team using the principles shown later
- Probably no more than 1-5 teams on initial project
  - Probably can't scale any higher during a medium length project anyway

Duration



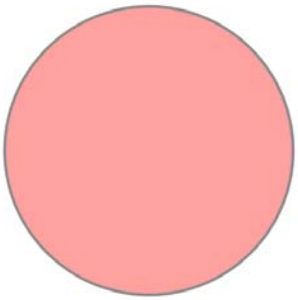
- Pick a project with a medium duration
  - Probably 2-4 months
  - Medium for your organization will be different from mine
- Don't want too short
  - Won't prove as much
    - "It only works on small projects"
- Don't want too long
  - Want to see value quickly

A business sponsor



- Pick a project that has a business sponsor who will engage because:
  - You'll need a business-side ally
  - A satisfied sponsor is your best chance of doing it again
  - The best way to have a satisfied business sponsor is to start with one willing to be involved
- Pick one who embraces agile

Importance



- Pick an important project because:
  - People don't change behavior unless they need to
  - Some parts of agile are difficult
    - It will be hard to stick with it unless the project is important
- Don't pick one that is too risky
- But if you pick a trivial project the results will be dismissed

You **can** start in the middle of a project

"Don't start with an initial 'learning project' that is of marginal importance. Start on a project that is absolutely critical to your company; otherwise it will be too difficult to implement all the hard things [agile] will ask of you."

~Jim Highsmith in *Adaptive Software Development Ecosysmtems*

The ideal project

- Has an active, engaged business sponsor
- Is critical to your organization
- Would likely or possibly fail if done in the status quo manner
- Can start with 1 team
  - Will grow to no more than 5 teams
- Lasts "a handful of months"

Get the right people on (and off) the project

- Be aware that there are typically two teams

The project team

The transition team

The project team

- The project team should
  - Be cross-functional / multi-disciplinary
    - everyone needed to finish all work of an iteration
  - Include opinion leaders
    - Both favorably disposed toward and opposed to agile
  - Include evangelists
  - Include those who've done it before (elsewhere)
  - Include a range of perspectives

The transition team

- Responsible for guiding a transition to agile
  - Sometimes this is implicitly the same team as the project team
    - "if you do well, we'll all switch"
- Think about
  - Who has the power to make or break the transition to agile?
  - Who controls critical resources or expertise?
  - How will each be affected?
  - How will each react?

Additional transition team considerations

- Who will gain or lose something by the transition to agile?
- Are there blocs likely to mobilize against or in support of the transition?
- Do team members have sufficient credibility that the teams' opinions and results are taken seriously?
- Can team members put their personal interests aside in favor of the organizational goal?

Who should not be on these teams

- People with big egos
  - Big egos fill the room; leave little space for others
  - Don't understand their own limitations
- Snakes
  - Someone who poisons relationships among team members
- Reluctant participants
  - Lack time or enthusiasm
  - But may have needed expertise or political clout

6 Start development with a beachhead team

- Cannot start effectively if focus is spread too thin
- Give them the early infrastructural tasks
- Their goal is to build enough of and the right parts of the system so that other teams can be added

Two approaches to expanding

1 Split and seed

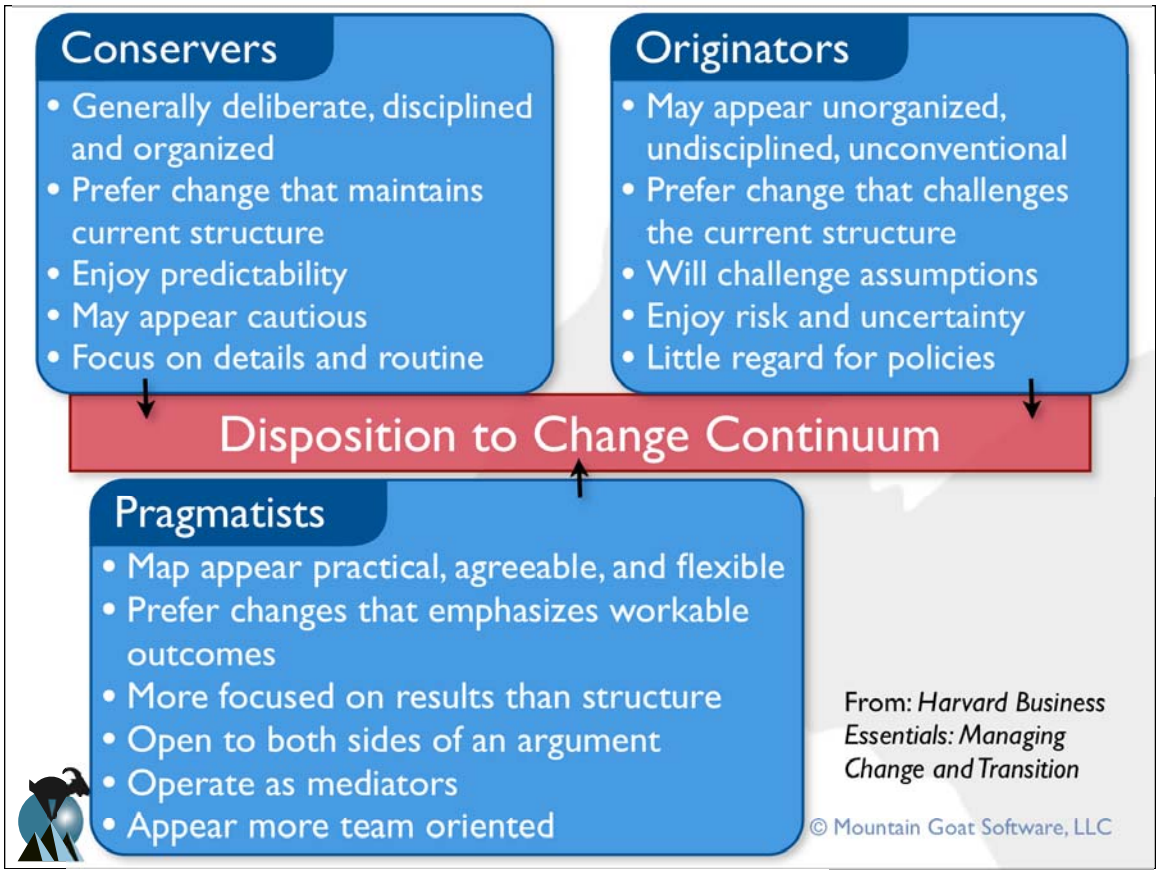
- Start with 1-3 teams and get them successful
- Split those teams up and use members to seed new teams
- Repeat

2 Internal coaching

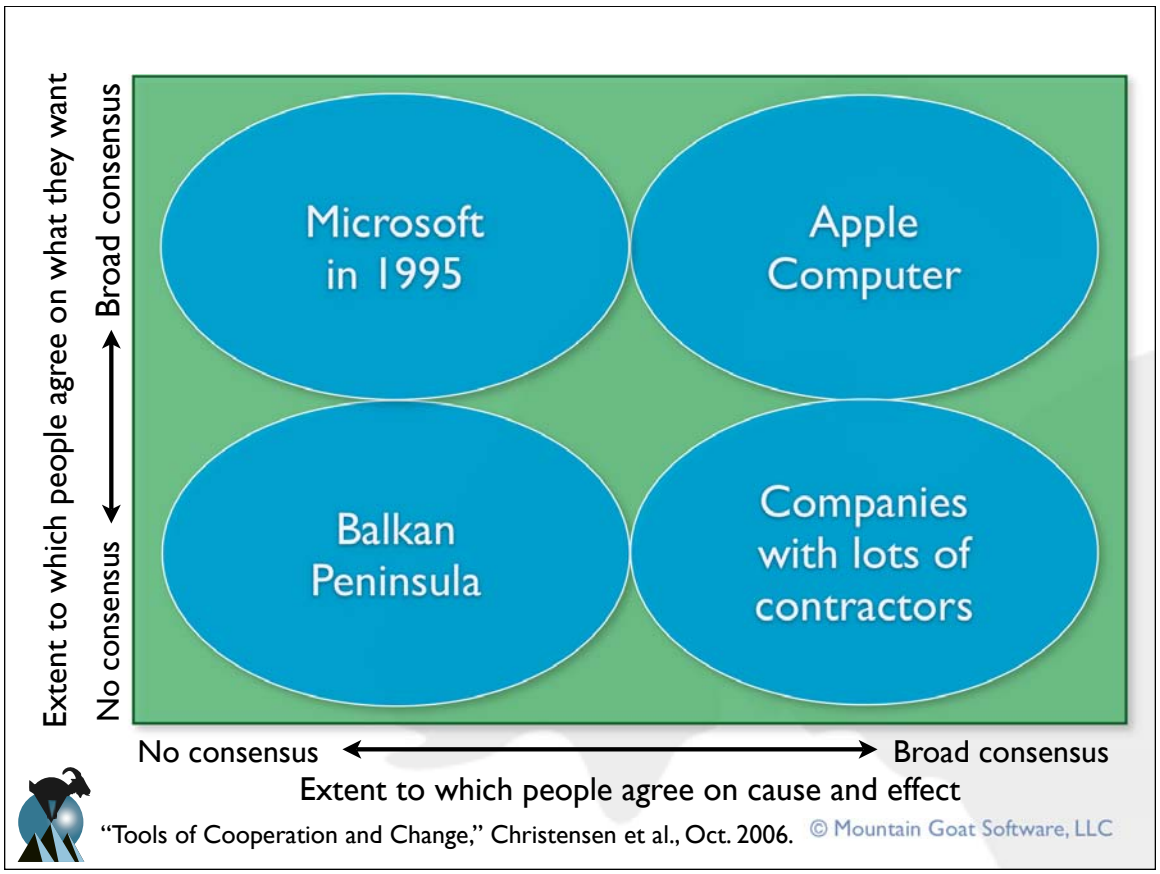
- Start with 1-3 teams and get them successful
- Keep teams together
- Select 1-2 coaches from those teams
  - Use them to advise new teams
  - Assign specific responsibilities
  - Attend planning meeting
  - Attend 2 daily scrums per week
  - Spend 4 hours with the team per sprint

7 Overcome resistance

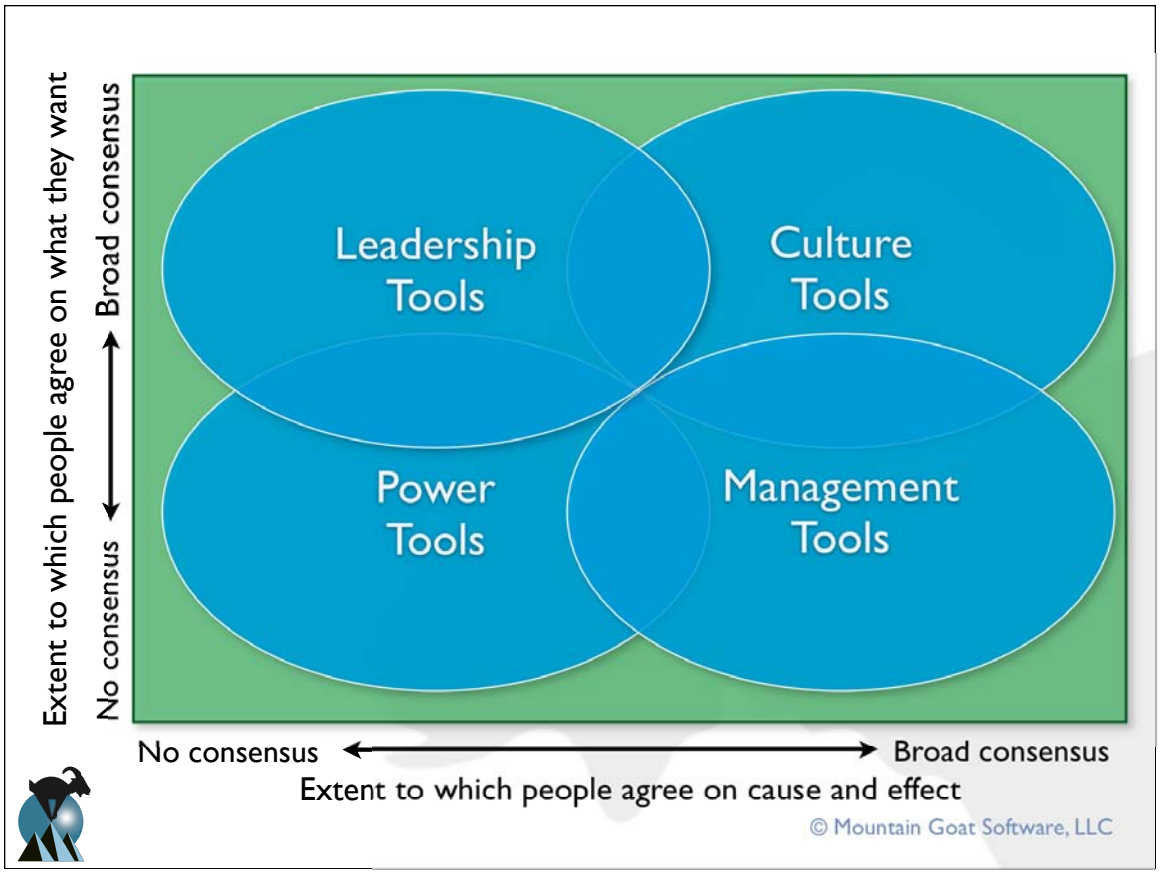
- Sell the problem, not the solution
  - No one wants a solution to a problem they don't (think they) have
  - Be open to hearing better solutions than you have
- Communicate why the change and why now
- Put team members in touch with customers
  - Let them hear the problems you are hearing
- Emphasize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


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
© Mountain Goat Software, LLC

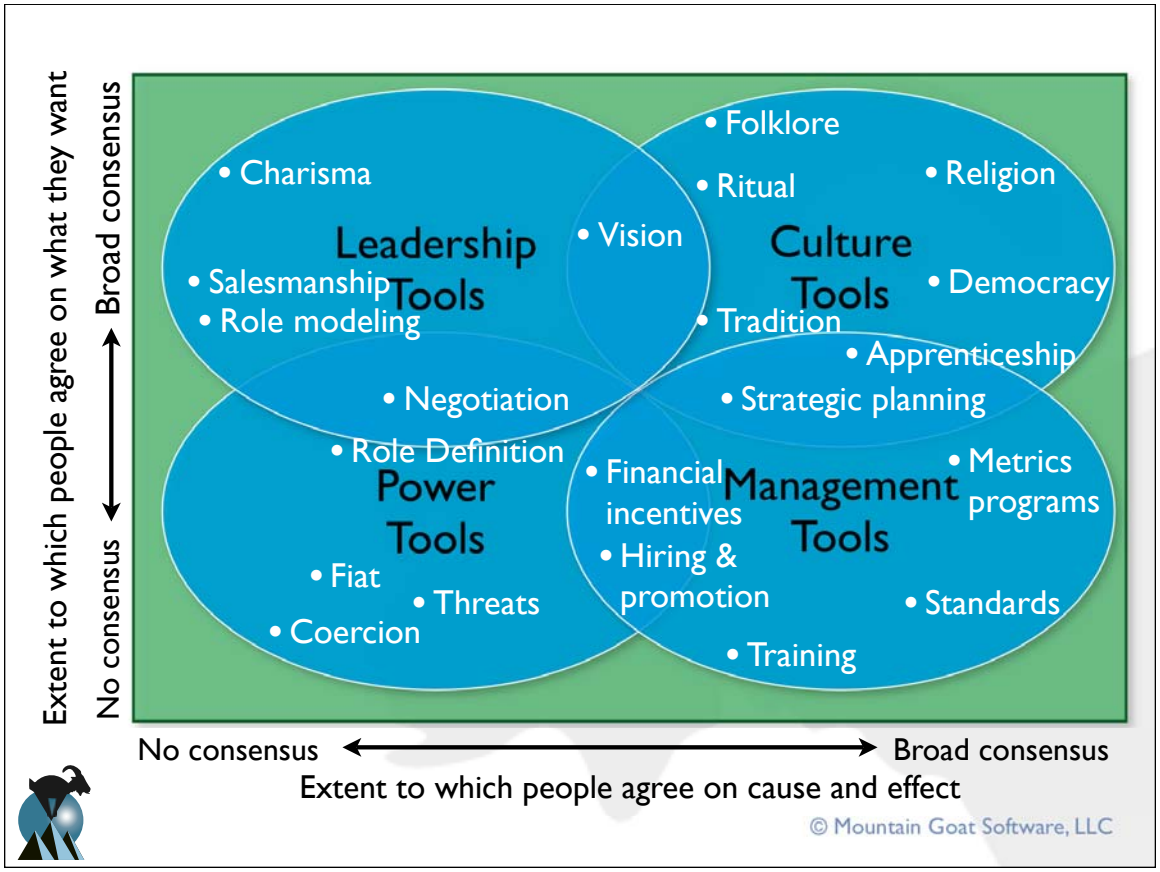
29


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 cause and effect
"Tools of Cooperation and Change," Christensen et al., Oct. 2006.
© Mountain Goat Software, LLC

30


Extent to which people agree on what they want
Broad consensus
No consensus
Leadership Tools
Culture Tools
Power Tools
Management Tools
No consensus
Broad consensus
Extent to which people agree on cause and effect
© Mountain Goat Software, LLC

31


Extent to which people agree on what they want
Broad consensus
No 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
• Financial incentives
• Hiring & promotion
• Metrics programs
• Standards
• Training
No consensus
Broad consensus
Extent to which people agree on cause and effect
© Mountain Goat Software, LLC

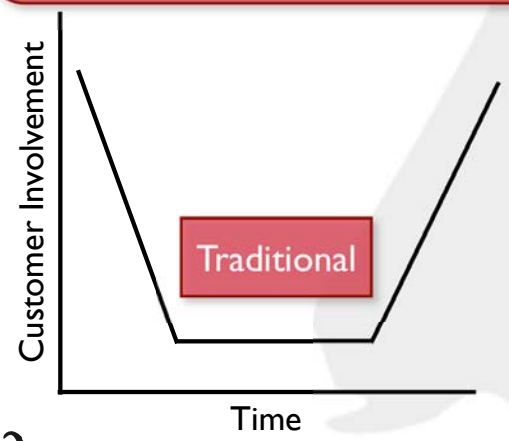
32

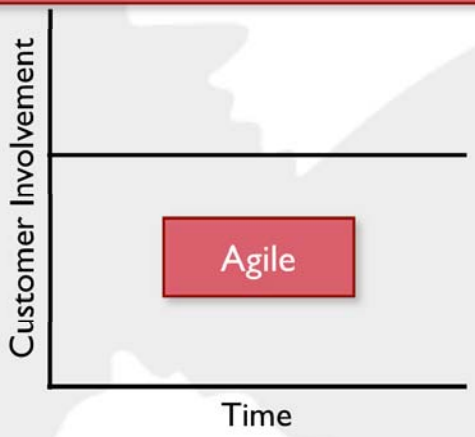
8 Have a customer

- To succeed you need a
  - Customer
  - Product Owner
  - Customer Team
- Whatever you want to call it, you need one
- Doesn't have to "sit with the team"
  - But set your targets around the level of involvement you'll get

Customer involvement

Agile processes don't require more customer involvement than a successful project with another process
BUT
the involvement is spread throughout the project.





Engage the change agents

Change agents...

- help others see problems and address them
- articulate the need for a change
- are accepted as trustworthy and competent
- can see and diagnose problems
- motivate people to change
- work through others to translate intent into action

Identifying change agents

- Find out who people listen to
  - These may not be people with formal authority
- Look for people who think differently
  - Change agents aren't satisfied with the status quo
- Consider new employees or others who may not be infected with a common mindset yet
- Consider people with different backgrounds
  - The programmer with the art history degree

10 Reflect

- The three most important words in agile:

Inspect and Adapt

- At the end of each iteration, reflect
- Whole team gathers and discusses what they'd like to:

Start doing | Stop doing | Continue doing

A start, stop, continue list

Start

- Showing the software to customers early
- Specifying acceptance tests early and with customers
- Doing code inspections
- Getting FitNesse into the nightly builds
- Trying to finish one story before moving to the next

Stop

- Being disrespectful of QA

Continue

- Making progress with the canonical database
- Emphasizing test automation

11 Don't try it all at once

- Don't try to do everything all at once
- Start with the agile practices that seem most valuable in your context
- Put other practices on your Transition Backlog
  - Prioritize them in as you go
  - Use Start/Stop/Continue meetings as a guide

My preferred sequence

Requirements as user stories

Estimate and create release plan

Nightly or continuous build with tests

Foundational skills (testing and OOD/P)

Follow a guide

- Follow the advice of someone who has been there, done that
  - An employee
  - A contractor
  - Outside mentor / coach
- Train
  - Better design skills, unit testing and test automation, agile project management, estimating and planning
- Can save you from many mistakes

Common sense?

"Integrating is painful."

So do it more often until you become so good at it that's it not painful.

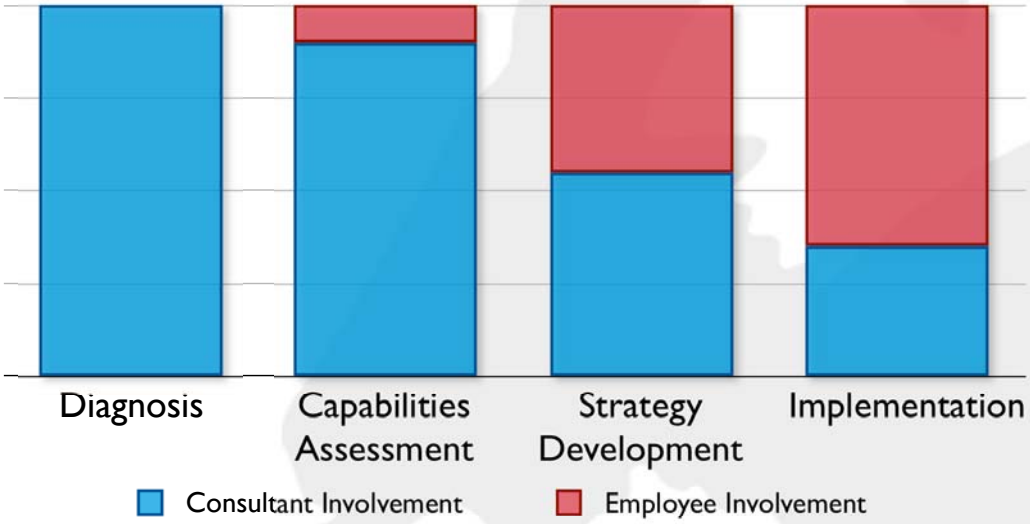
"We're having trouble delivering finished software in two weeks."

So try one-week iterations until you've mastered that. Two weeks will then be a breeze.

"There's not enough room to write requirements on cards."

So use smaller cards.

The role of consultants and employees in change programs



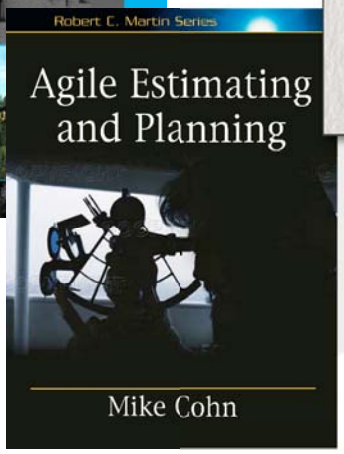
From: *Harvard Business Essentials: Managing Change and Transition*

Upcoming public classes

Date	What	Where
April 10–11 April 12	Certified ScrumMaster Agile Estimating and Planning	Santa Clara, CA
May 30-31	Certified Scrum Product Owner (with Ken Schwaber)	Boston, MA
June 11-12 June 13	Certified ScrumMaster Agile Estimating and Planning	Dallas
Jul 31-Aug 1 August 2	Certified ScrumMaster Agile Estimating and Planning	Denver, CO
Sept 11-12 Sept 13	Certified ScrumMaster Agile Estimating and Planning	Orlando, FL
Other classes in London, Stockholm, and Oslo if you're up for a longer trip.		

Register at
www.mountaingoatsoftware.com

Mike Cohn contact info



mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)

MOUNTAIN GOAT SOFTWARE