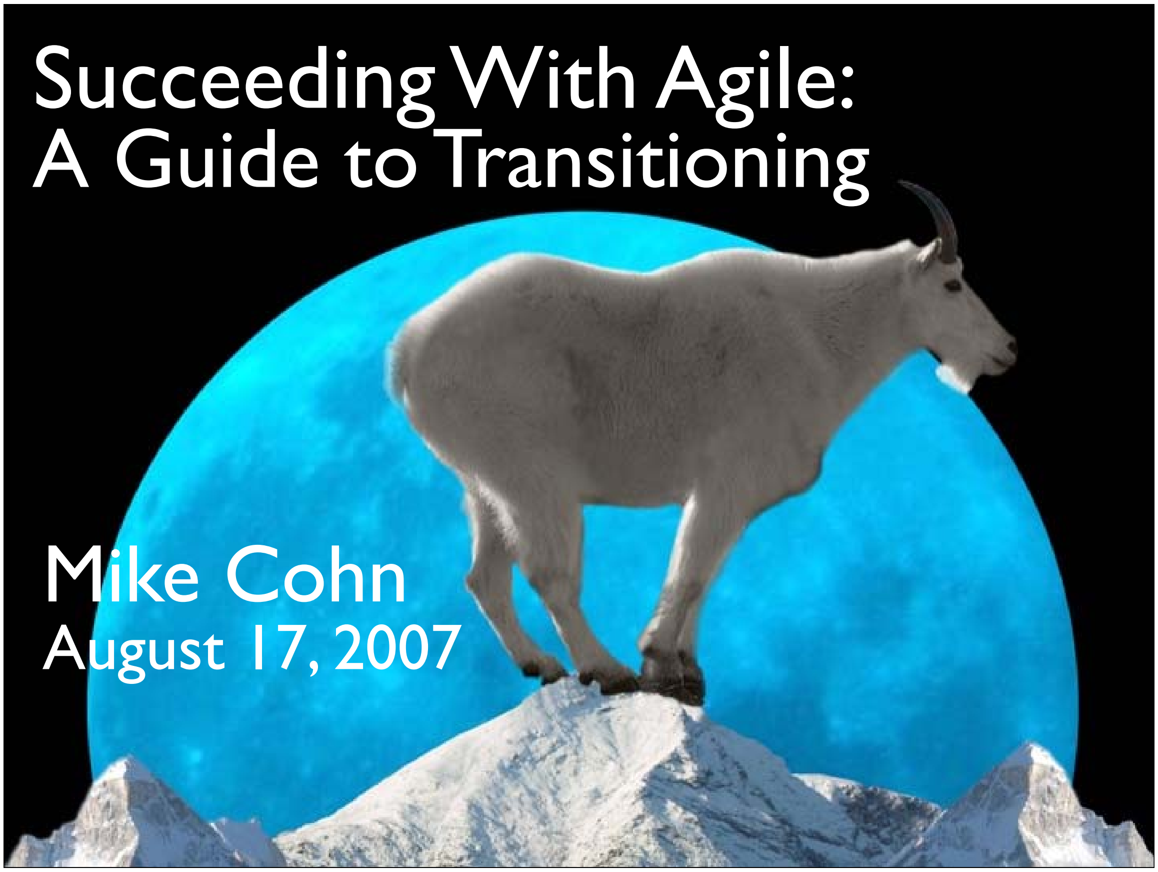
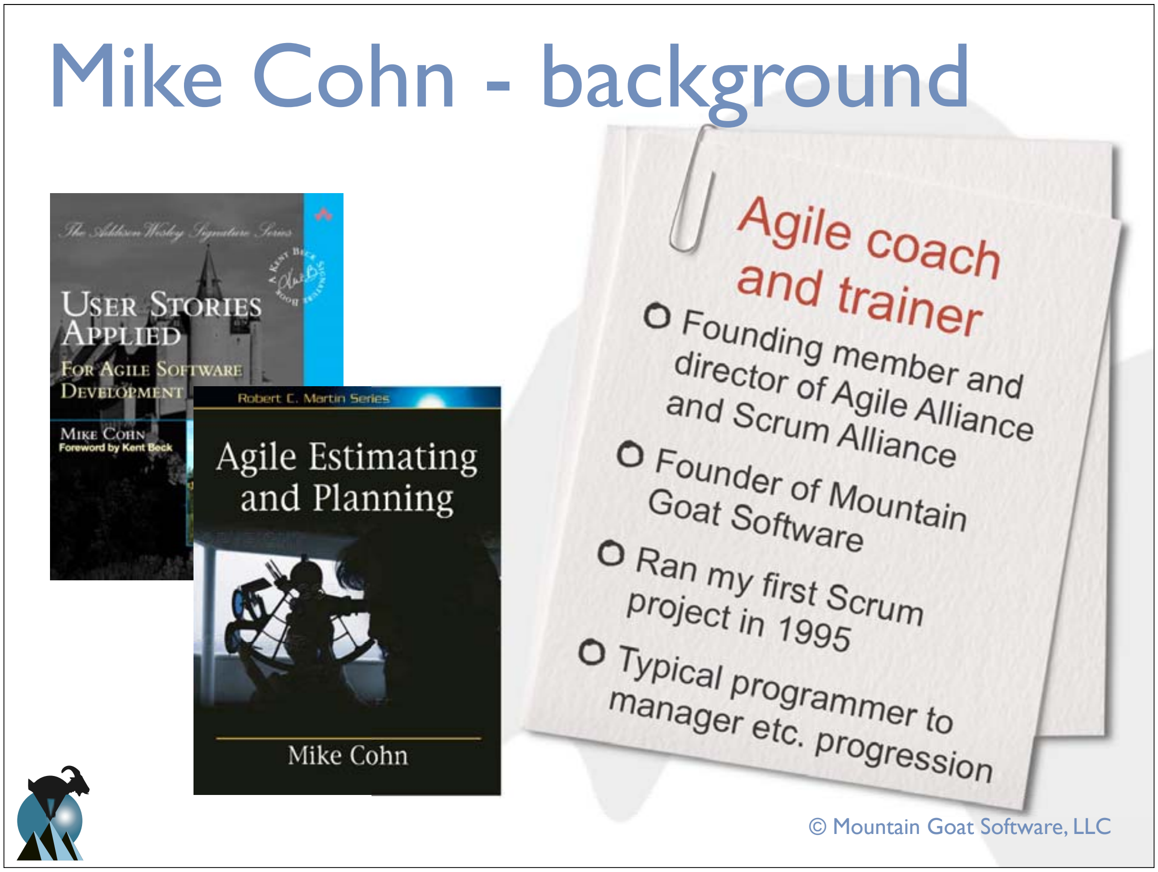


Succeeding With Agile: A Guide to Transitioning

Mike Cohn

August 17, 2007

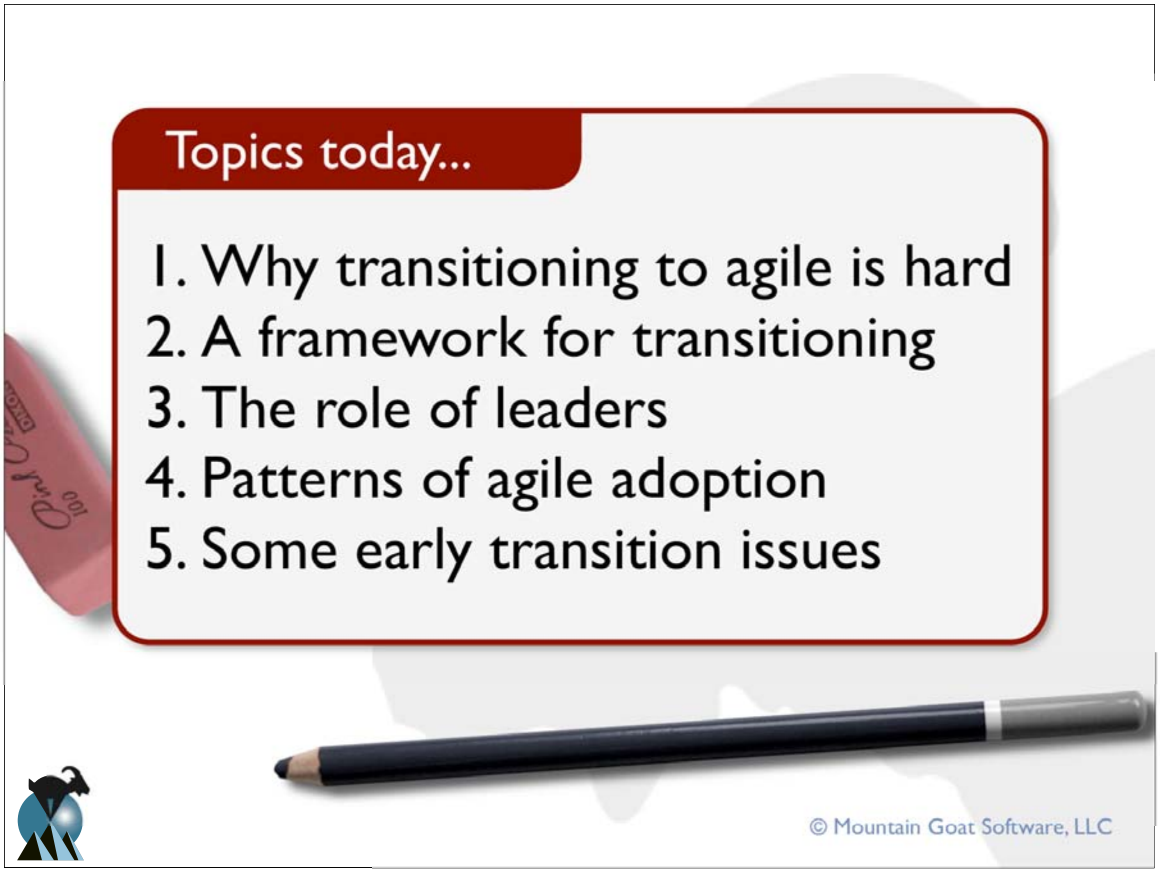


Mike Cohn - background

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

Agile coach and trainer

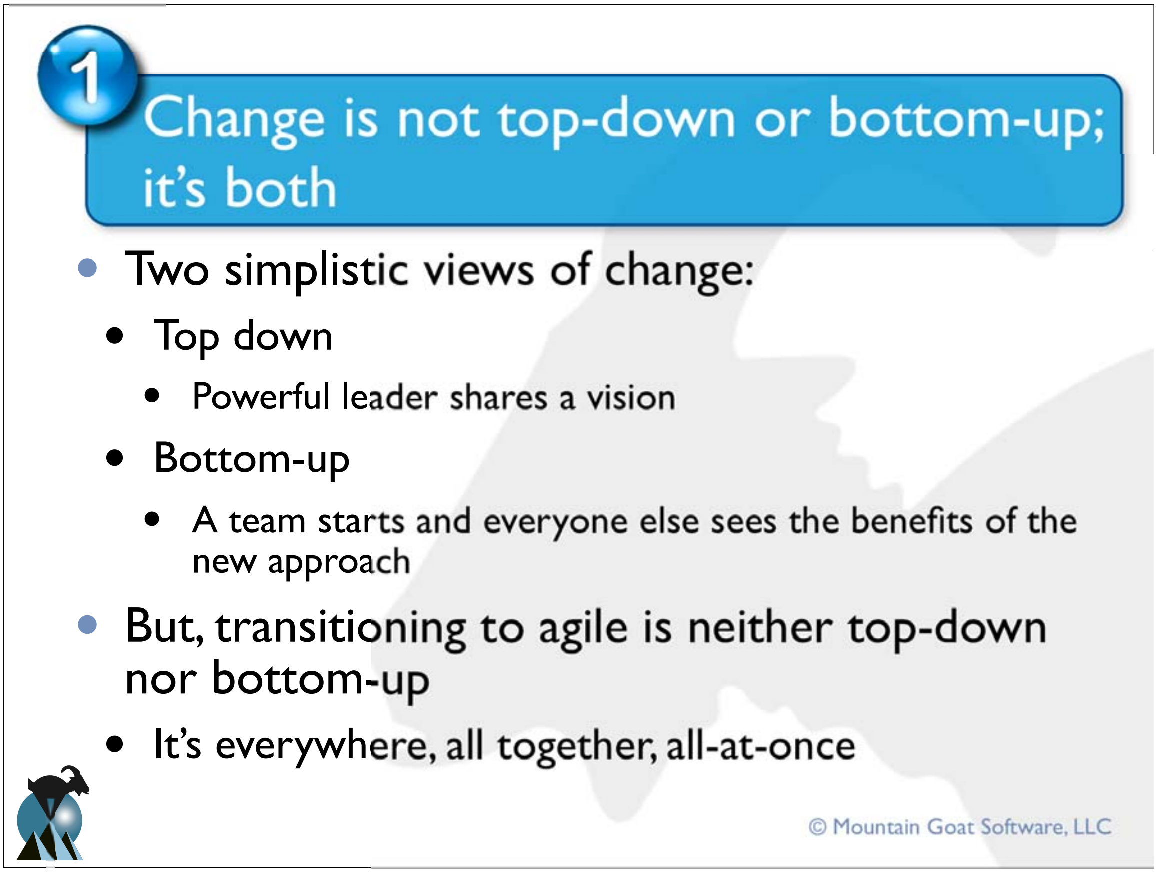
- Founding member and director of Agile Alliance and Scrum Alliance
- Founder of Mountain Goat Software
- Ran my first Scrum project in 1995
- Typical programmer to manager etc. progression



Topics today...

1. Why transitioning to agile is hard
2. A framework for transitioning
3. The role of leaders
4. Patterns of agile adoption
5. Some early transition issues

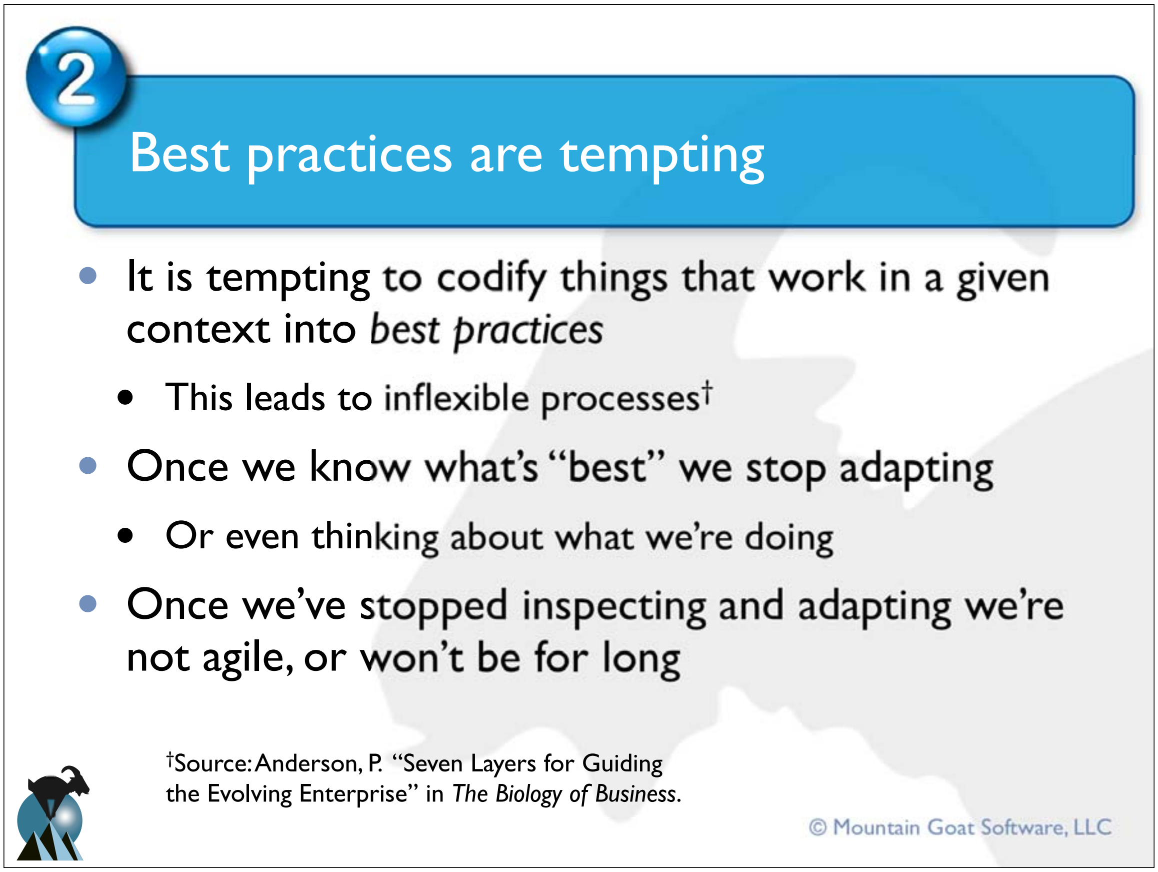
Why Transitioning to Agile Is Hard



1 Change is not top-down or bottom-up; it's both

- Two simplistic views of change:
  - 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

© Mountain Goat Software, LLC

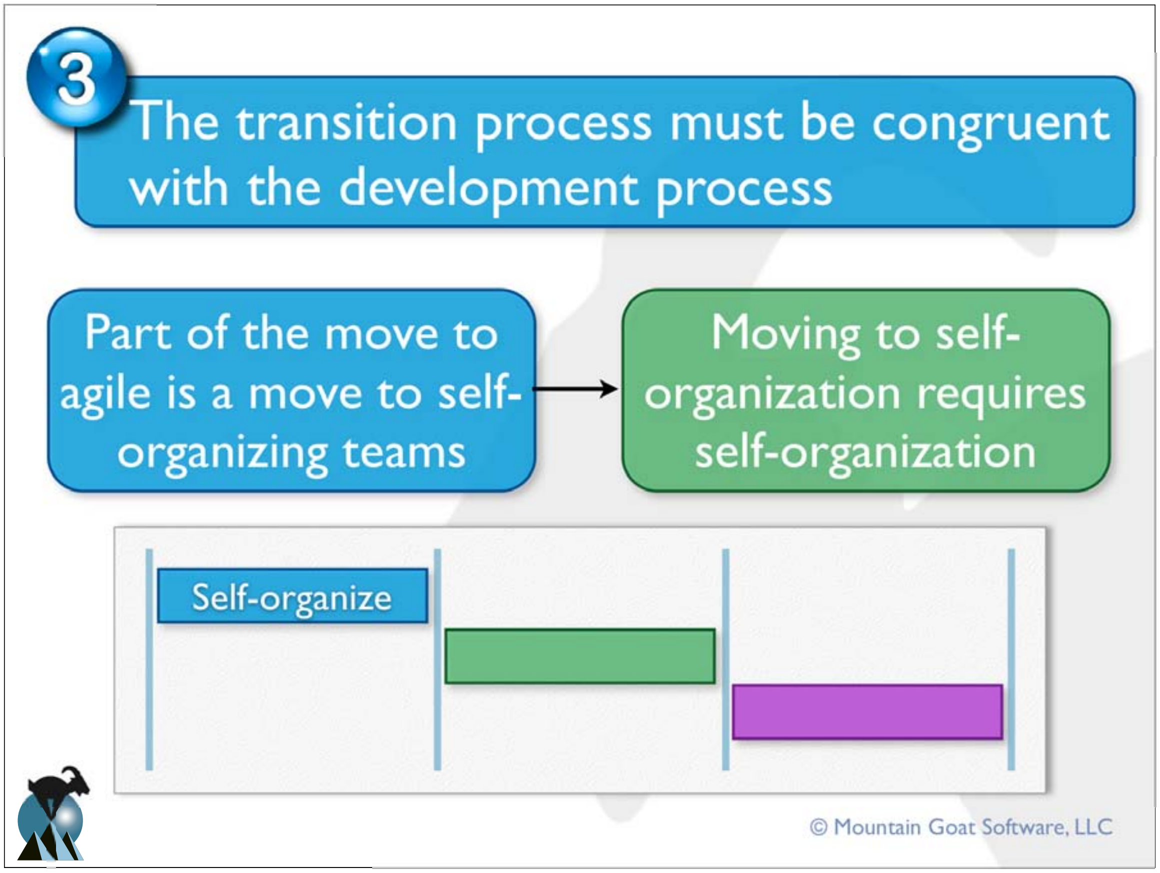


2 Best practices are tempting

- It is tempting to codify things that work in a given context into *best practices*
  - This leads to inflexible processes†
- Once we know what's "best" we stop adapting
  - Or even thinking about what we're doing
- Once we've stopped inspecting and adapting we're not agile, or won't be for long

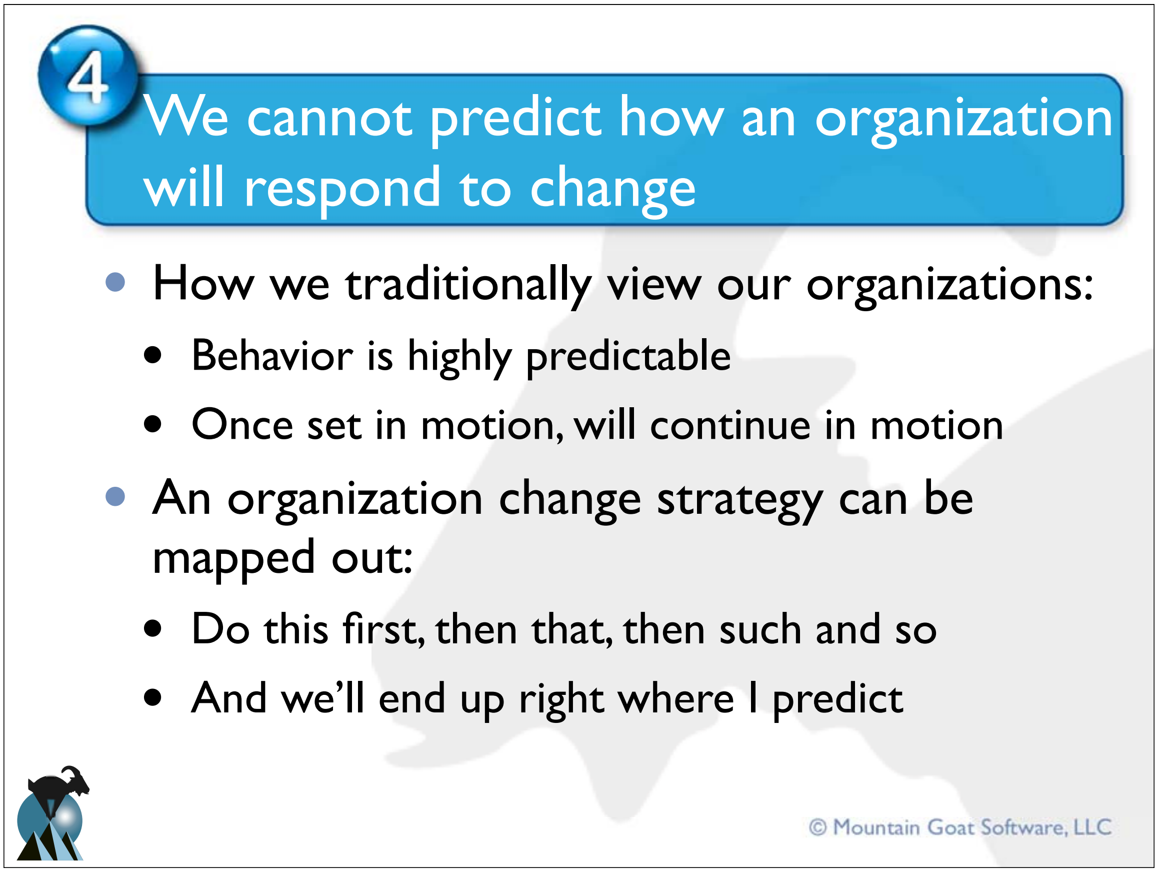
†Source: Anderson, P. "Seven Layers for Guiding the Evolving Enterprise" in *The Biology of Business*.

© Mountain Goat Software, LLC



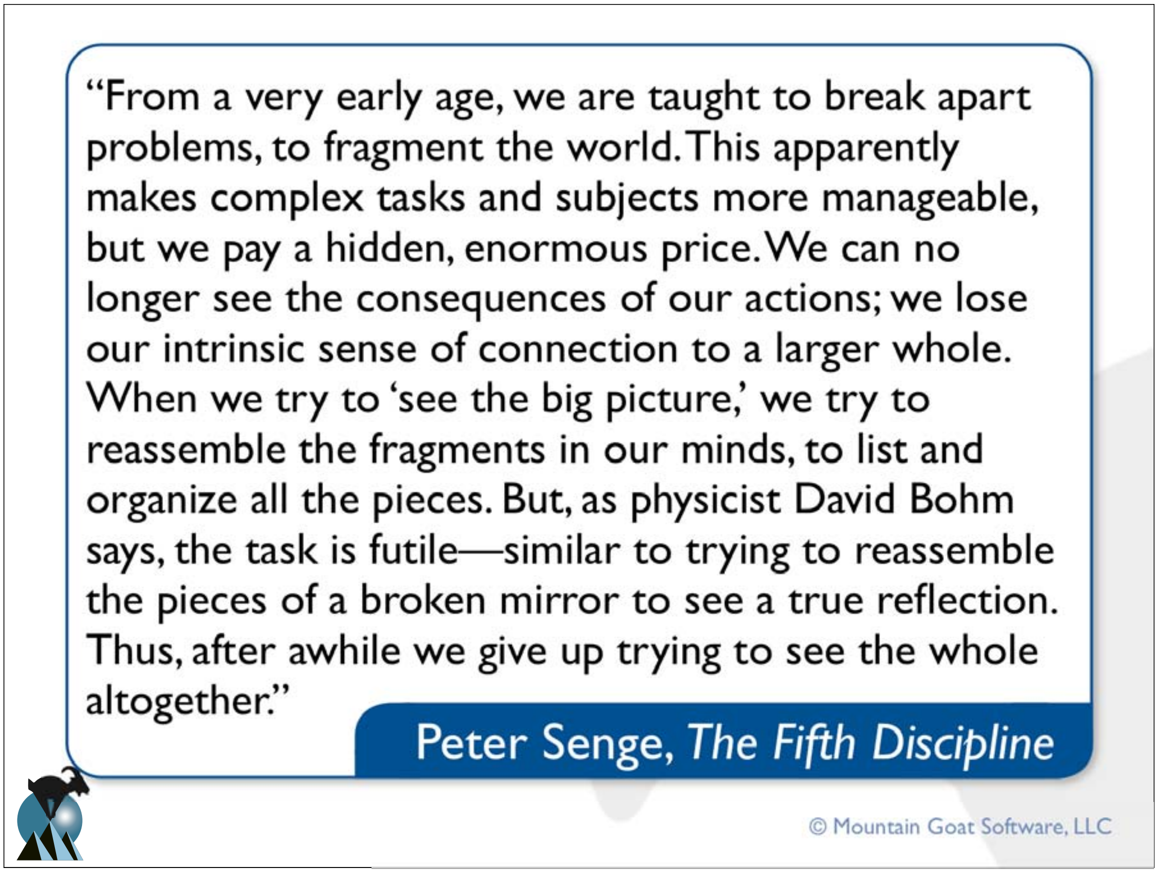
3 The transition process must be congruent with the development process

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



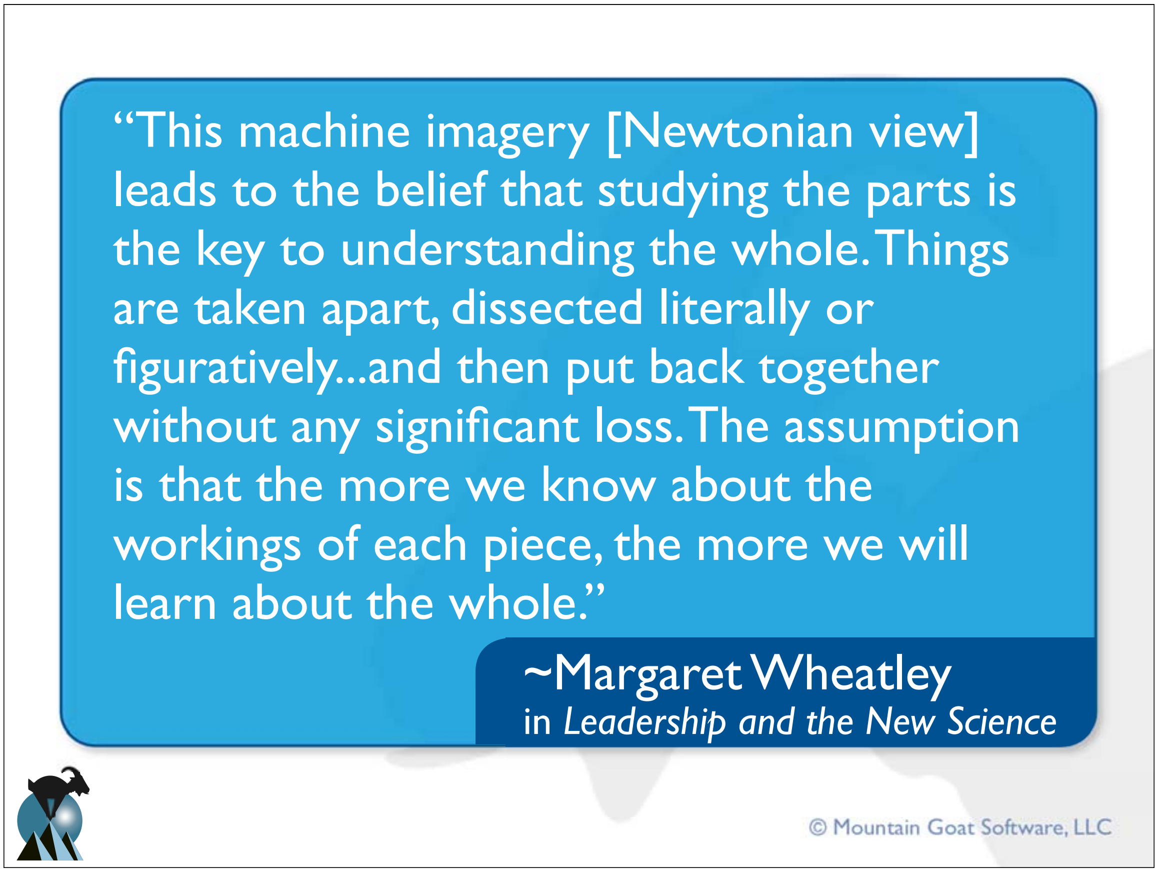
4 We cannot predict how an organization will respond to change

- How we traditionally view our organizations:
  - Behavior is highly predictable
  - Once set in motion, will continue in motion
- 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



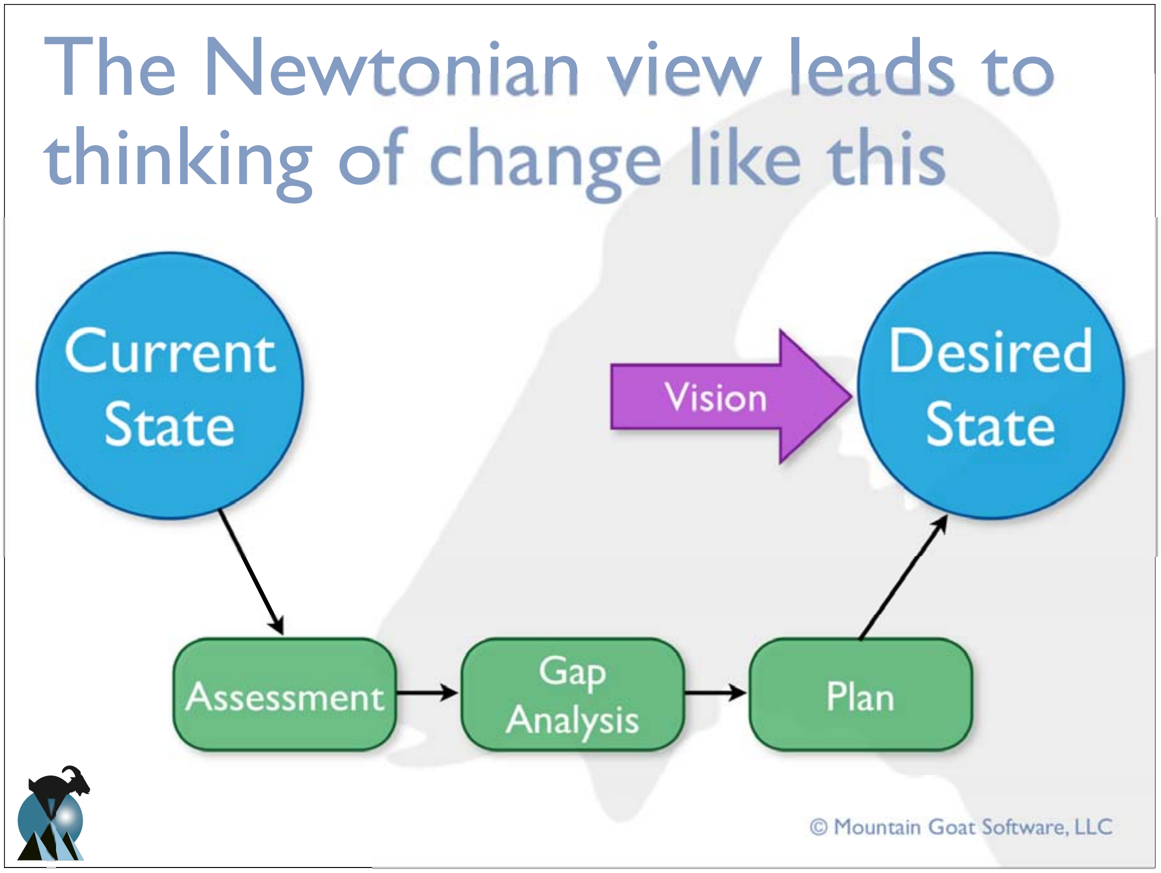
"From a very early age, we are taught to break apart problems, to fragment the world. This apparently makes complex tasks and subjects more manageable, but we pay a hidden, enormous price. We can no longer see the consequences of our actions; we lose our intrinsic sense of connection to a larger whole. When we try to 'see the big picture,' we try to reassemble the fragments in our minds, to list and organize all the pieces. But, as physicist David Bohm says, the task is futile—similar to trying to reassemble the pieces of a broken mirror to see a true reflection. Thus, after awhile we give up trying to see the whole altogether."

Peter Senge, *The Fifth Discipline*



"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*



The Newtonian view leads to thinking of change like this

Current State

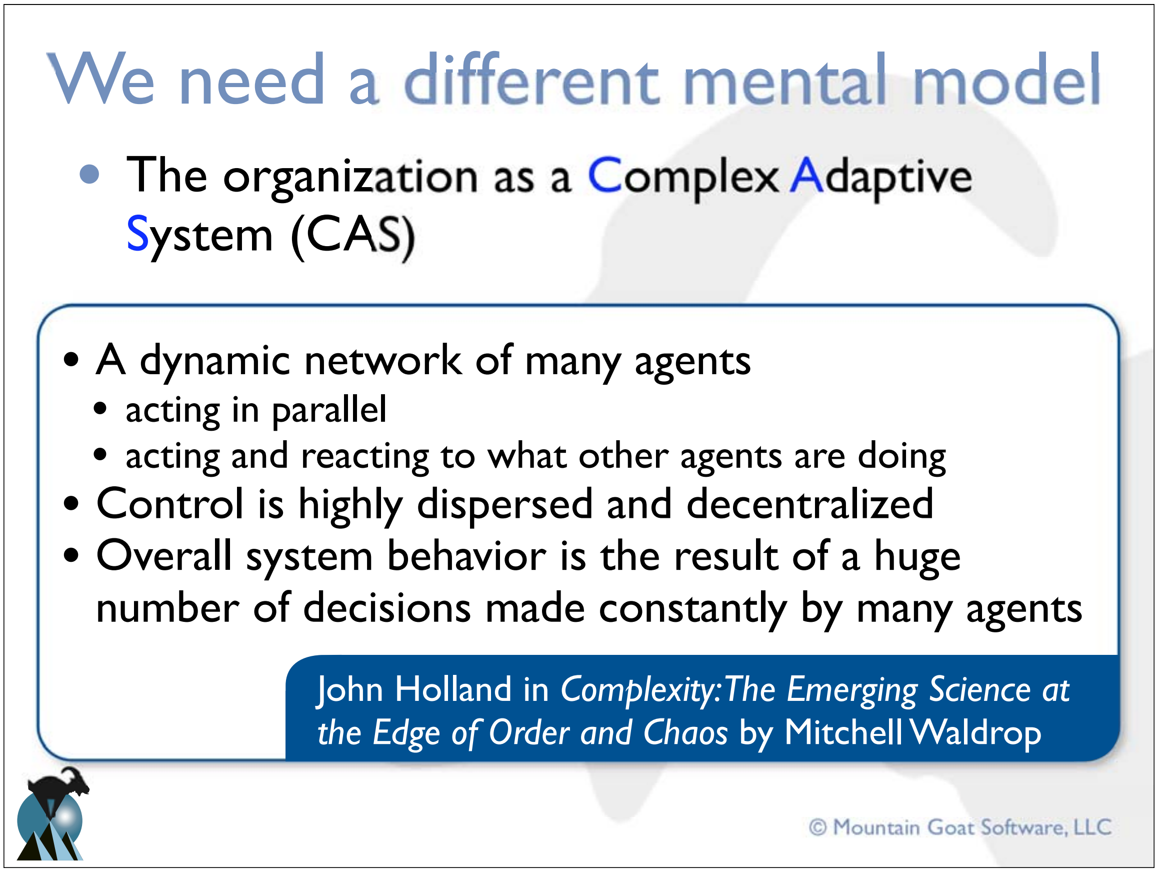
Vision

Desired State

Assessment

Gap Analysis

Plan



We need a different mental model

- The organization as a Complex Adaptive System (CAS)

- A dynamic network of many agents
  - acting in parallel
  - acting and reacting to what other agents are doing
- Control is highly dispersed and decentralized
- Overall system behavior is the result of a huge number of decisions made constantly by many agents

John Holland in *Complexity: The Emerging Science at the Edge of Order and Chaos* by Mitchell Waldrop

Differing views of success

Newtonian view

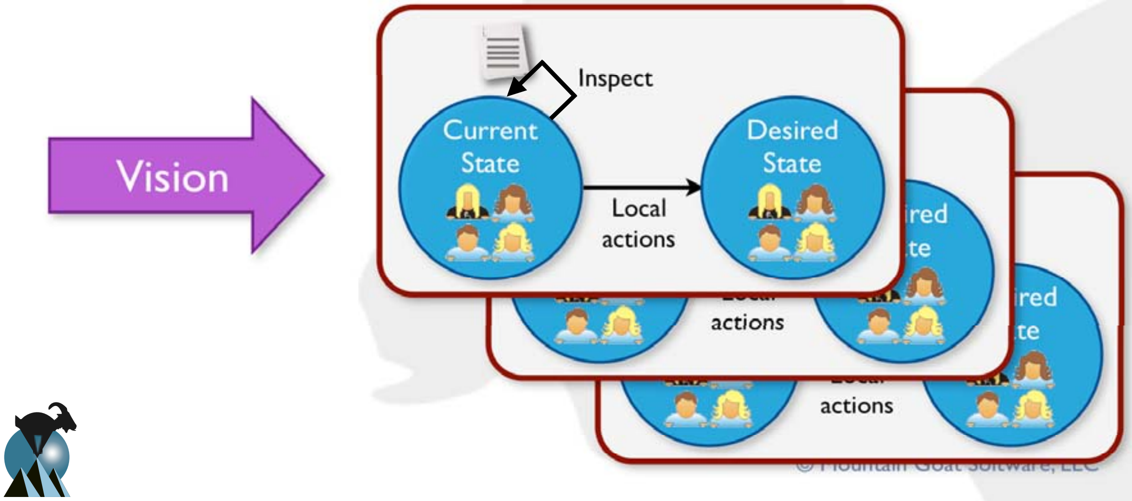
Success =
closing the gap with the desired state

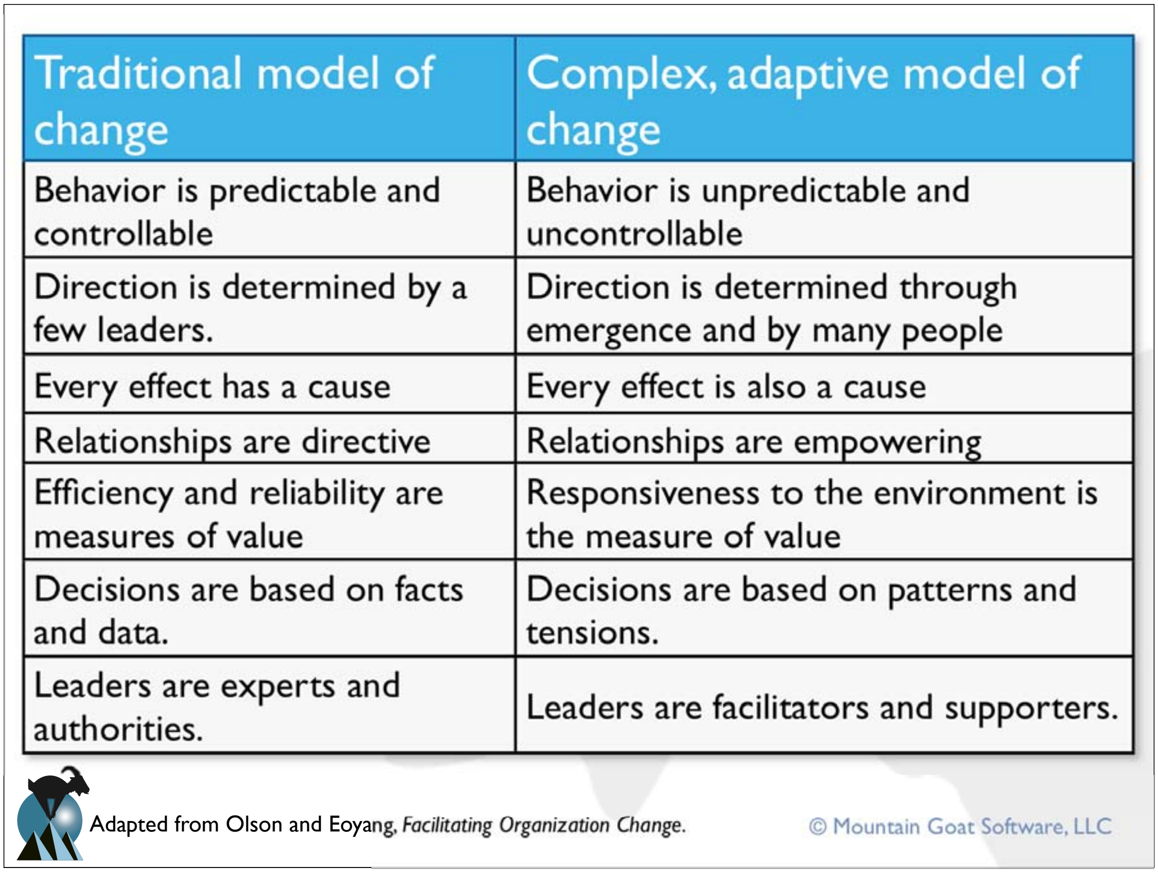
CAS view

Success =
achieving a good fit with the environment

Local goals and gaps

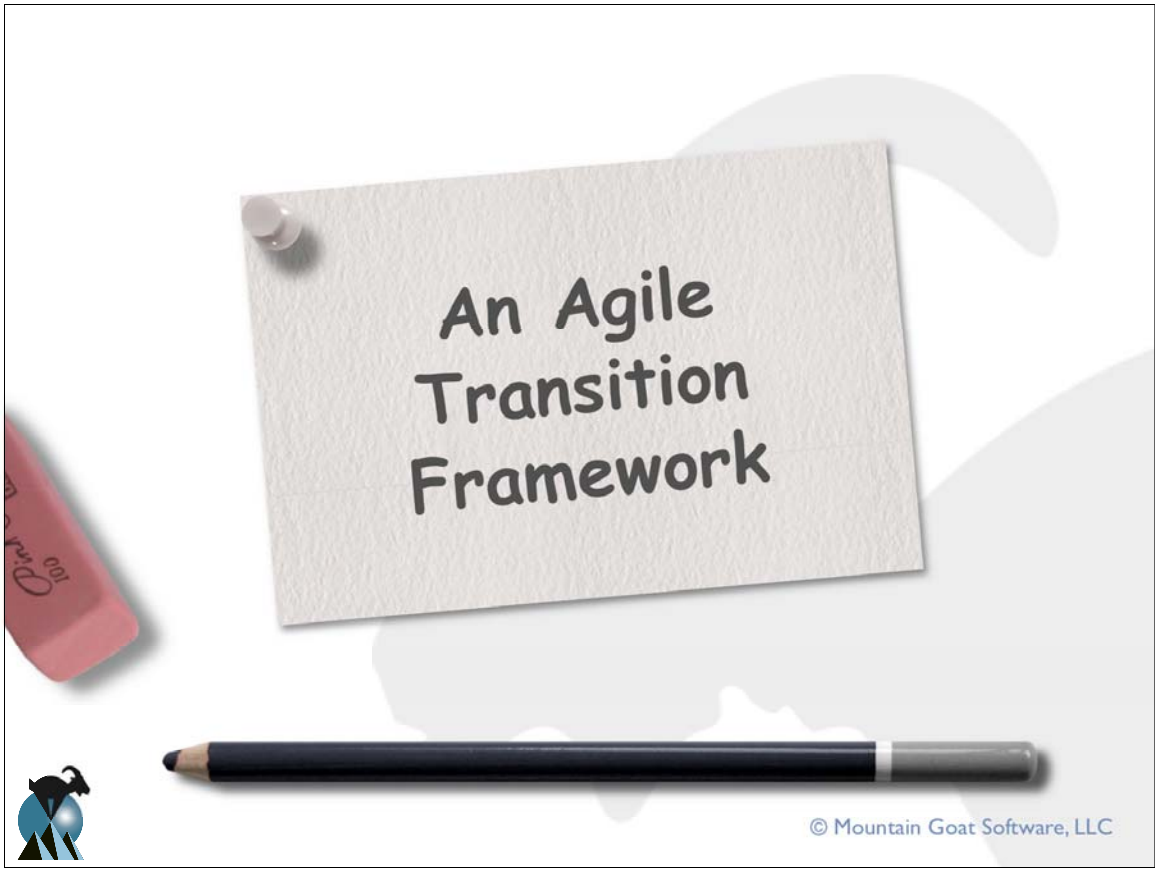
- Local agents (individuals, project teams, discipline coworkers) identify local gaps based on their local goals





Traditional model of change	Complex, adaptive model of change
Behavior is predictable and controllable	Behavior is unpredictable and uncontrollable
Direction is determined by a few leaders.	Direction is determined through emergence and by many people
Every effect has a cause	Every effect is also a cause
Relationships are directive	Relationships are empowering
Efficiency and reliability are measures of value	Responsiveness to the environment is the measure of value
Decisions are based on facts and data.	Decisions are based on patterns and tensions.
Leaders are experts and authorities.	Leaders are facilitators and supporters.

Adapted from Olson and Eoyang, *Facilitating Organization Change.*



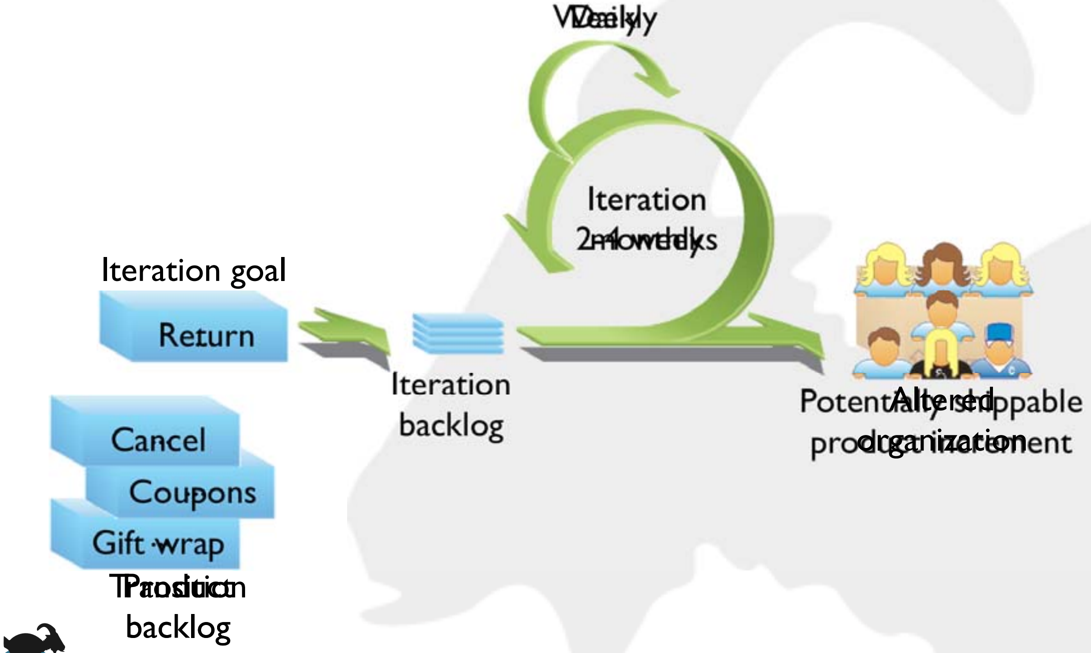
An Agile Transition Framework

- 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

An agile transition process

An agile process



Weekly

Daily

Iteration

2-4 weeks

1 month

Iteration goal

Return

Iteration backlog

Cancel

Coupons

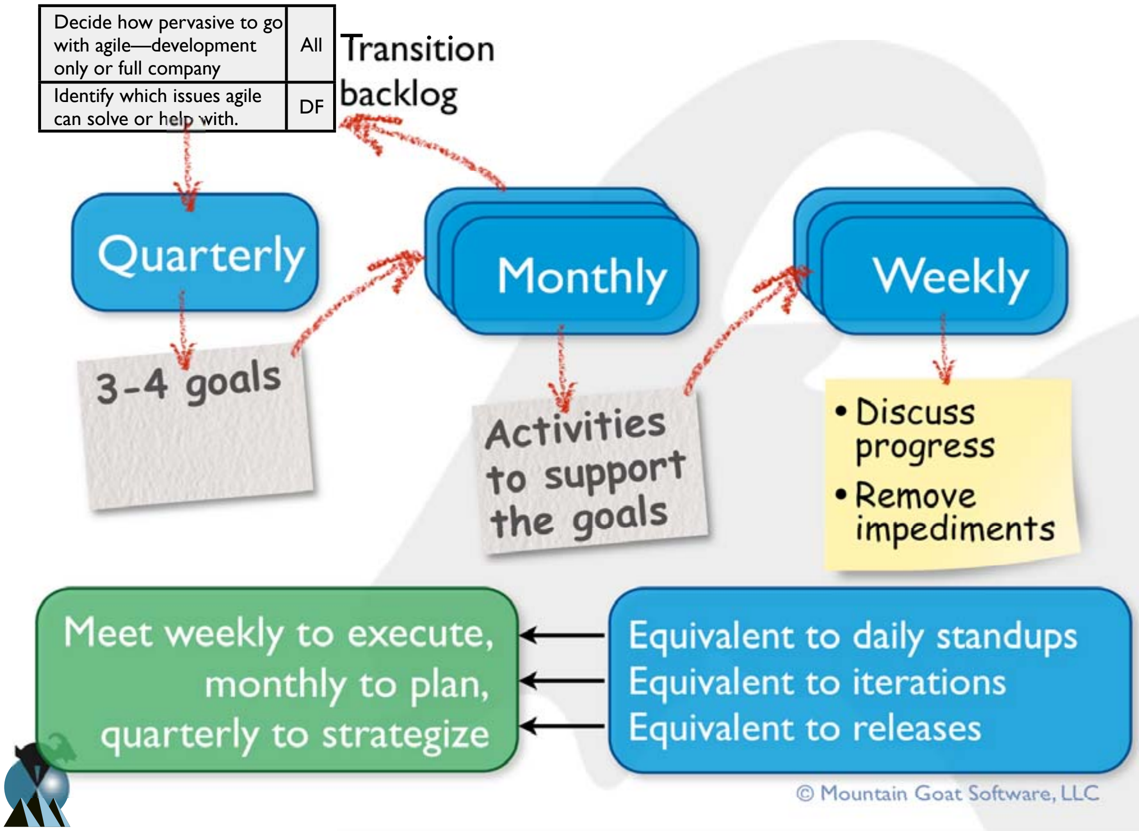
Gift wrap

Transition backlog

Product backlog

Potentially shippable product increment

Altered organization



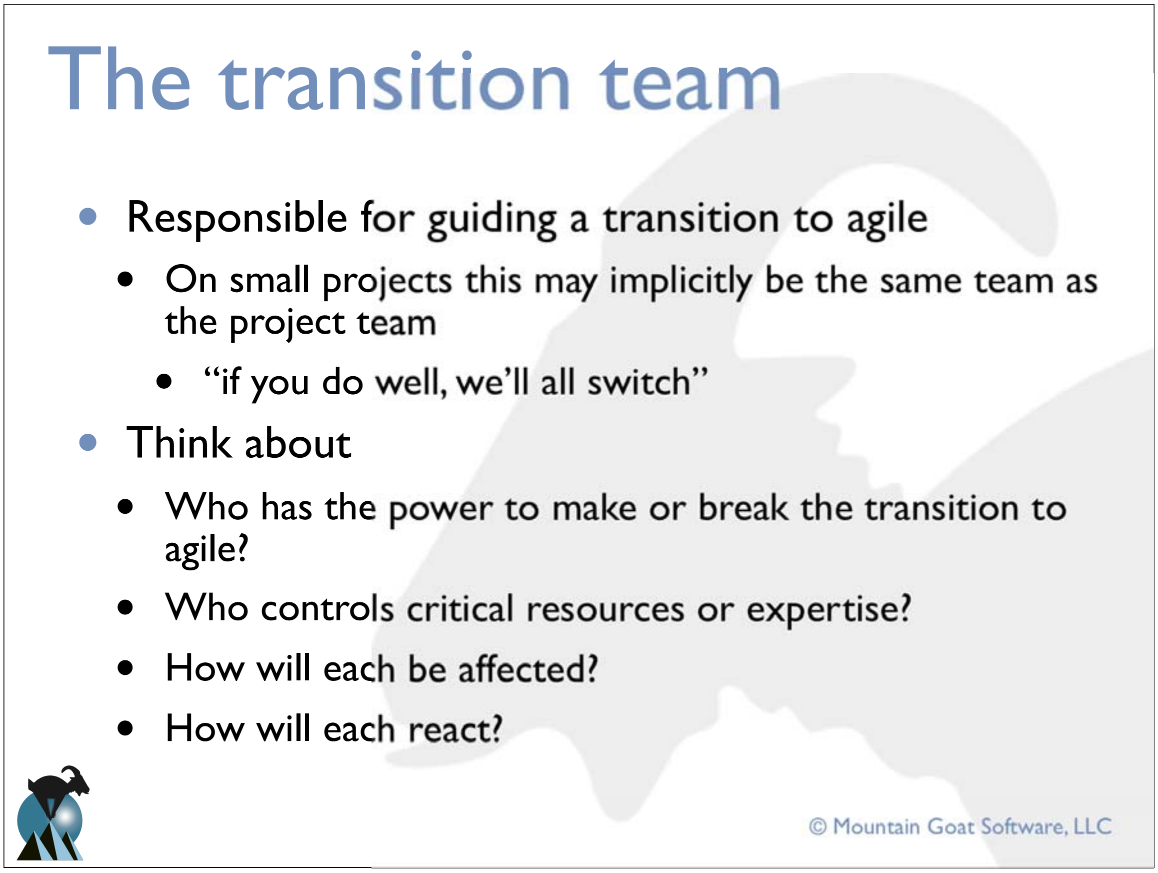
Transition backlog	
Decide how pervasive to go with agile—development only or full company	All
Identify which issues agile can solve or help with.	DF

Meet weekly to execute, monthly to plan, quarterly to strategize

Equivalent to daily standups
Equivalent to iterations
Equivalent to releases

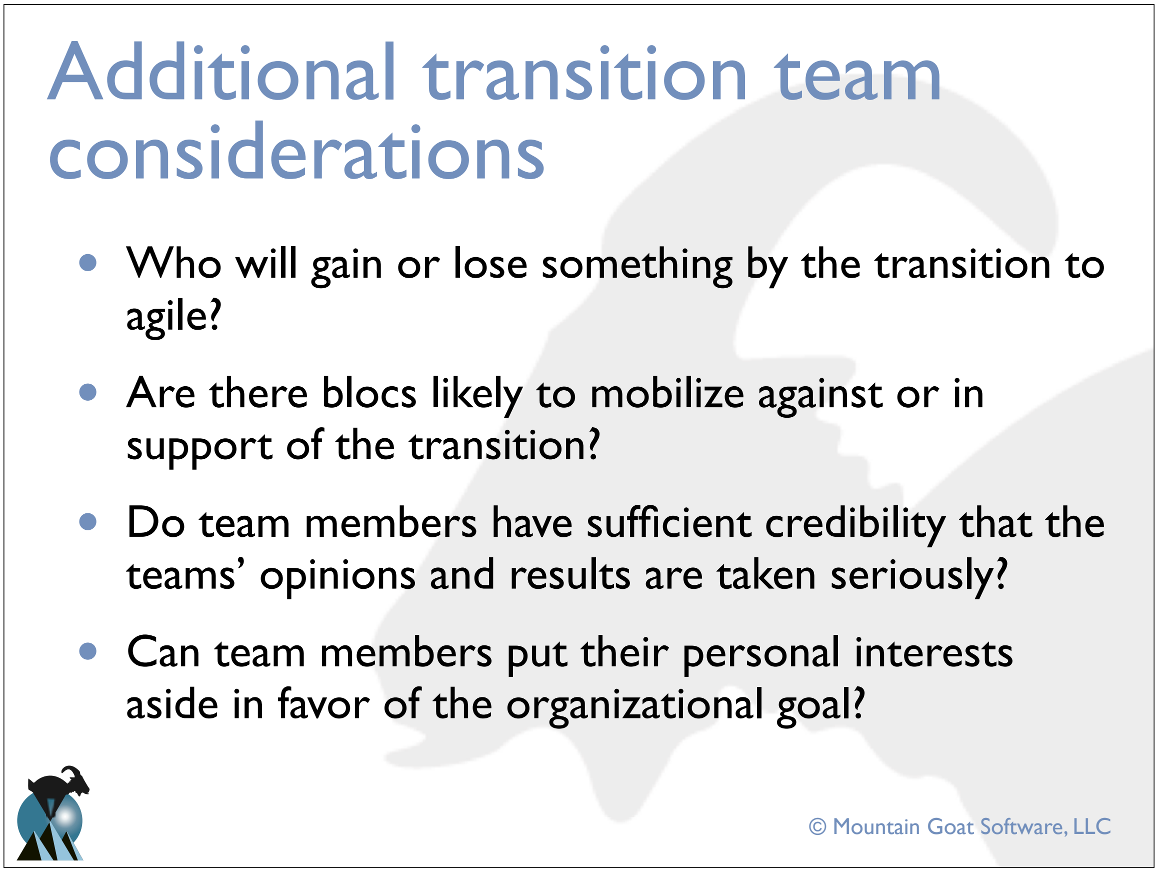
Establish a “guiding coalition”

- Who?
  - Sponsor—senior person responsible for success
  - Area managers or leads who can make it happen



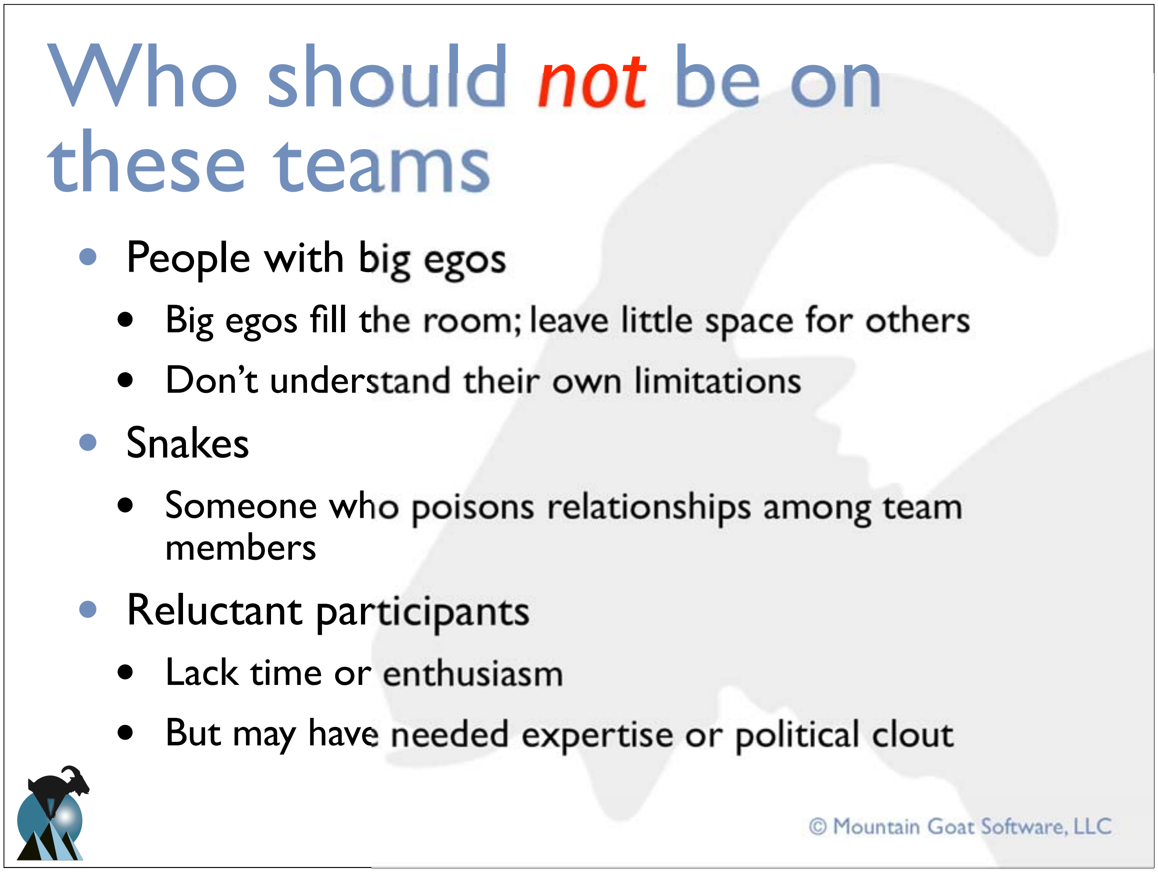
The transition team

- Responsible for guiding a transition to agile
  - On small projects this may implicitly be 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

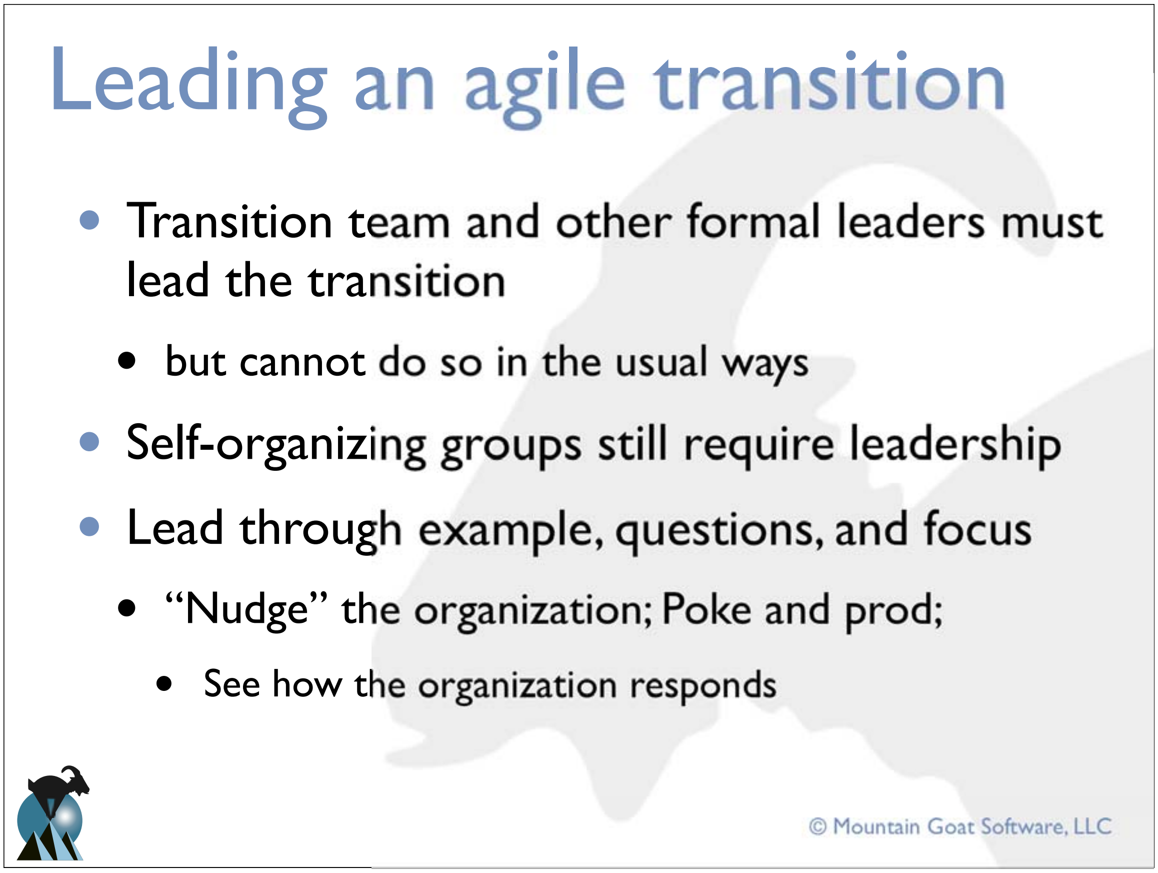
- Who will gain or lose something by the transition to agile?
- Are there blocs likely to mobilize against or in support of the transition?
- Do team members have sufficient credibility that the teams’ opinions and results are taken seriously?
- Can team members put their personal interests aside in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



Who should *not* be on these teams

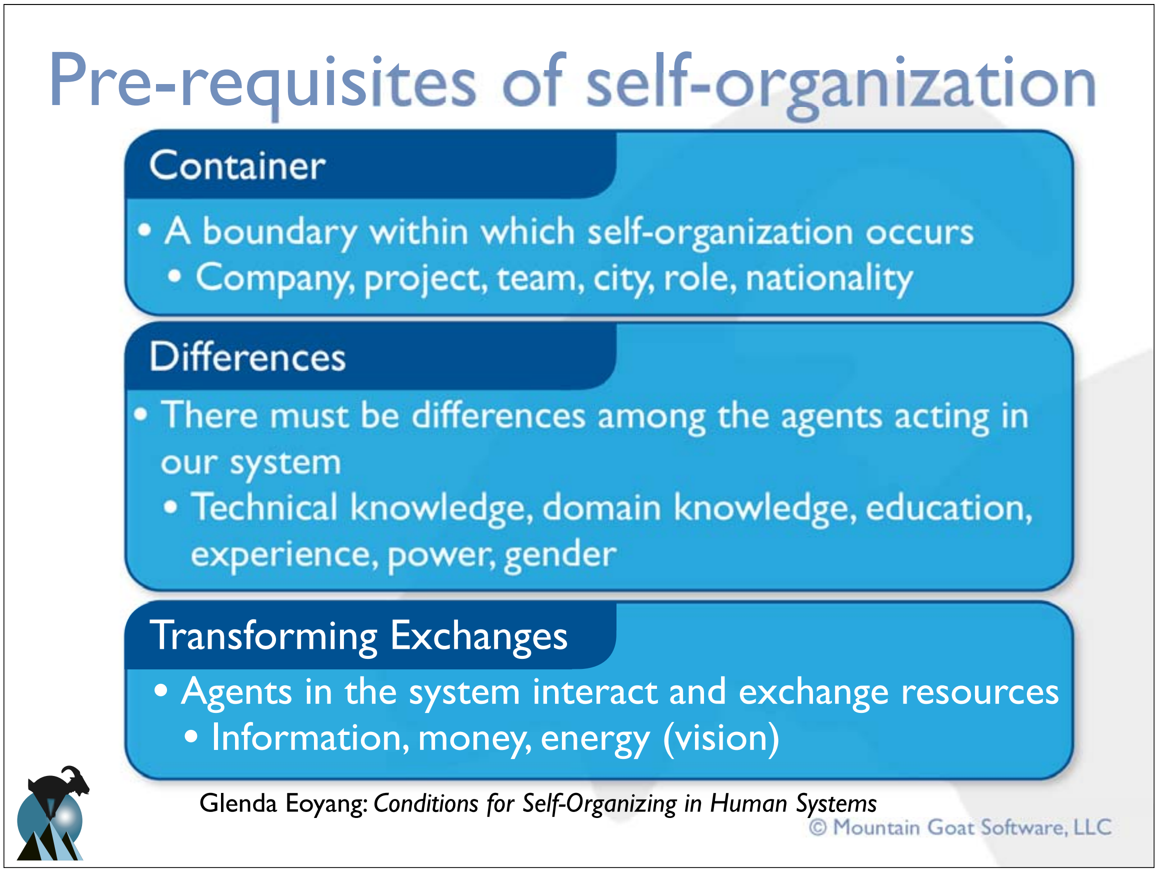
- People with big egos
  - Big egos fill the room; leave little space for others
  - Don’t understand their own limitations
- Snakes
  - Someone who poisons relationships among team members
- Reluctant participants
  - Lack time or enthusiasm
  - But may have needed expertise or political clout

The Role of Leaders



Leading an agile transition

- Transition team and other formal leaders must lead the transition
  - but cannot do so in the usual ways
- Self-organizing groups still require leadership
- Lead through example, questions, and focus
  - “Nudge” the organization; Poke and prod;
    - See how the organization responds



Pre-requisites of self-organization

Container

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

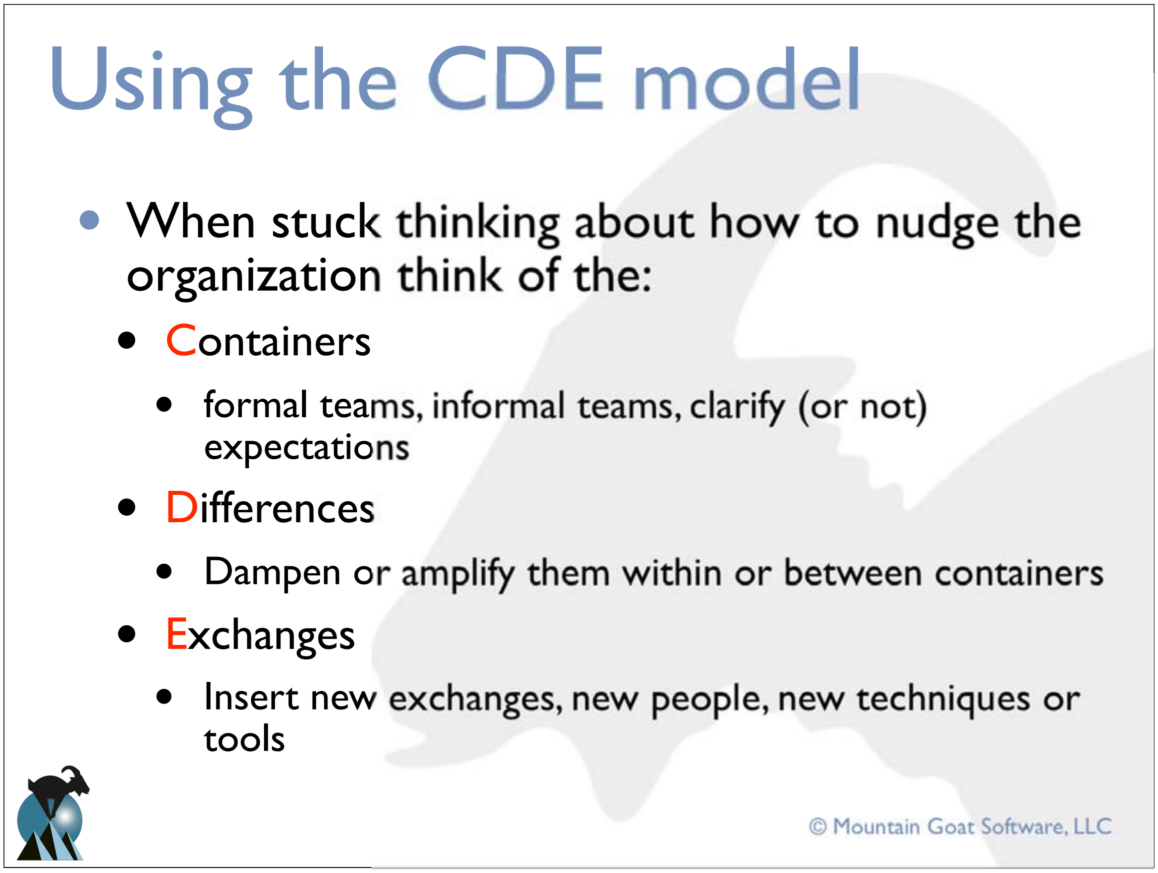
Differences

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

Transforming Exchanges

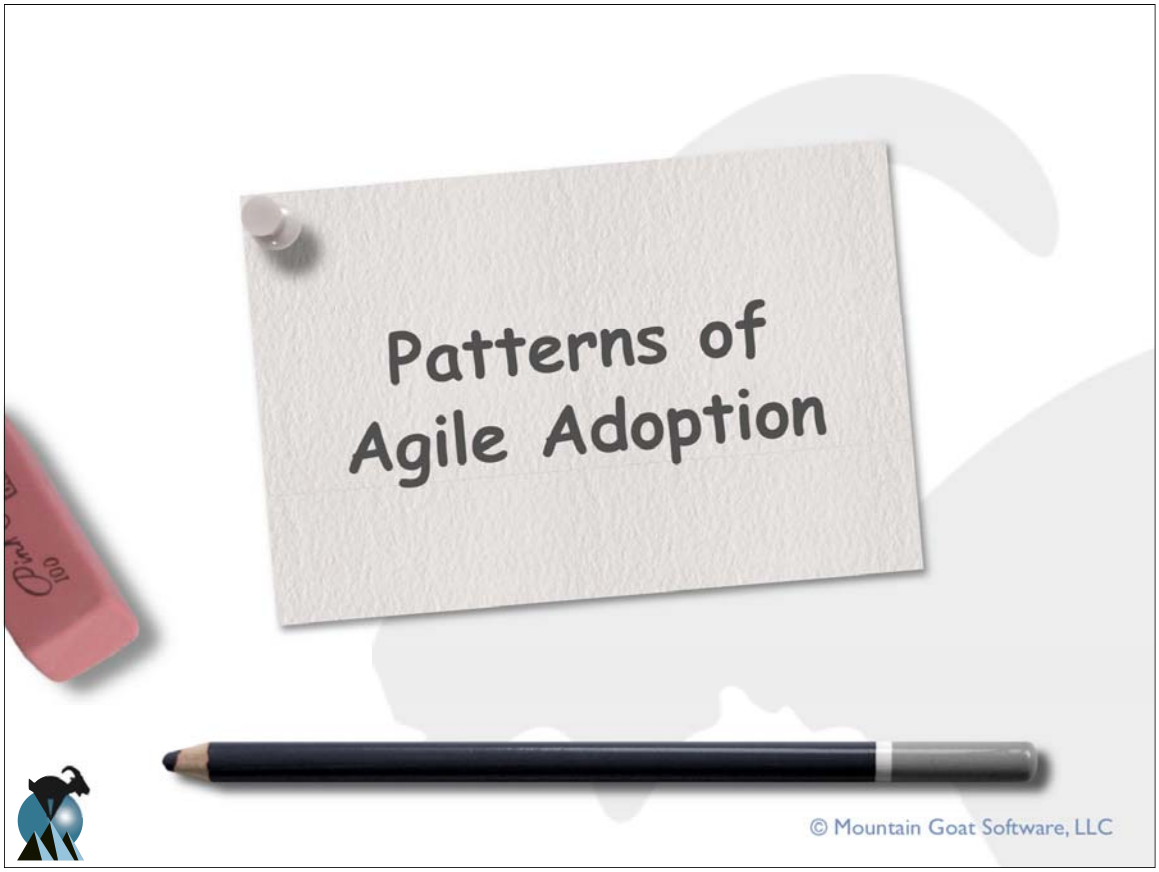
- Agents in the system interact and exchange resources
  - Information, money, energy (vision)

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

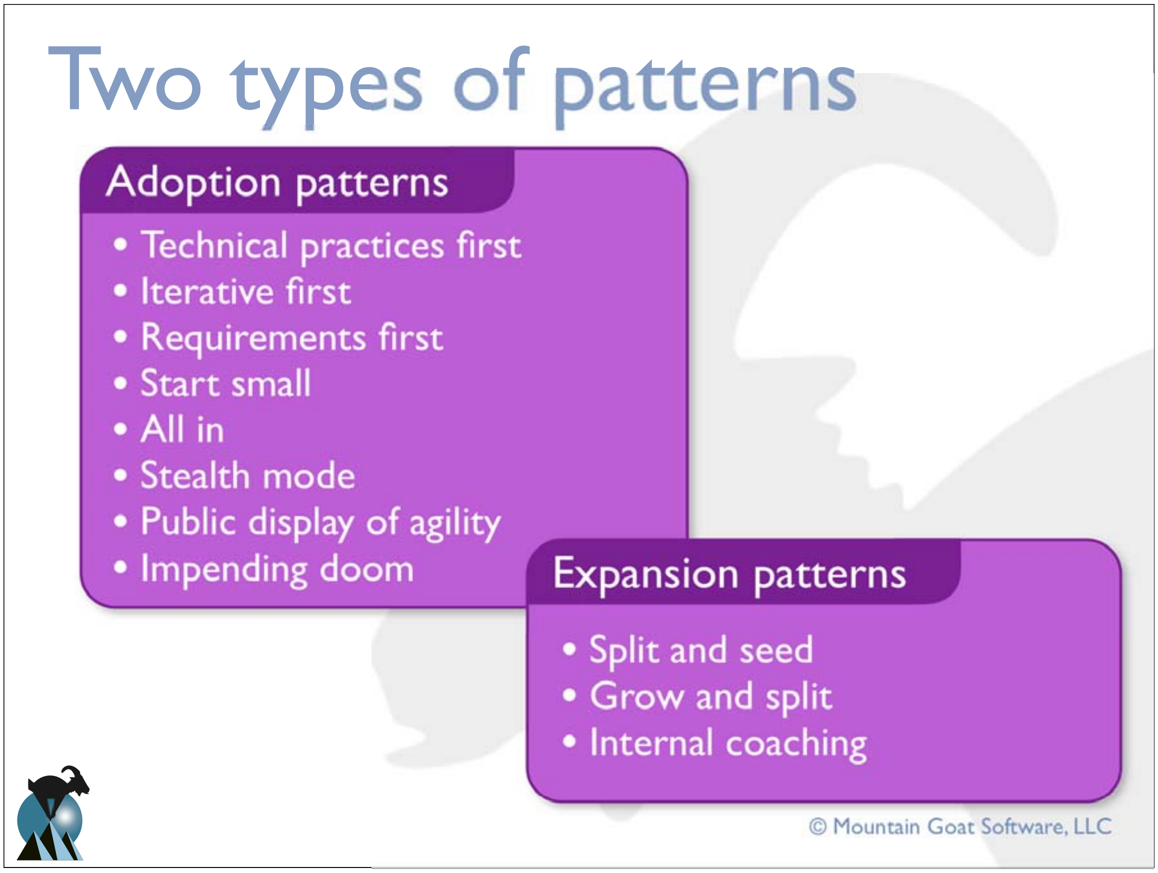


Using the CDE model

- 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



Patterns of Agile Adoption



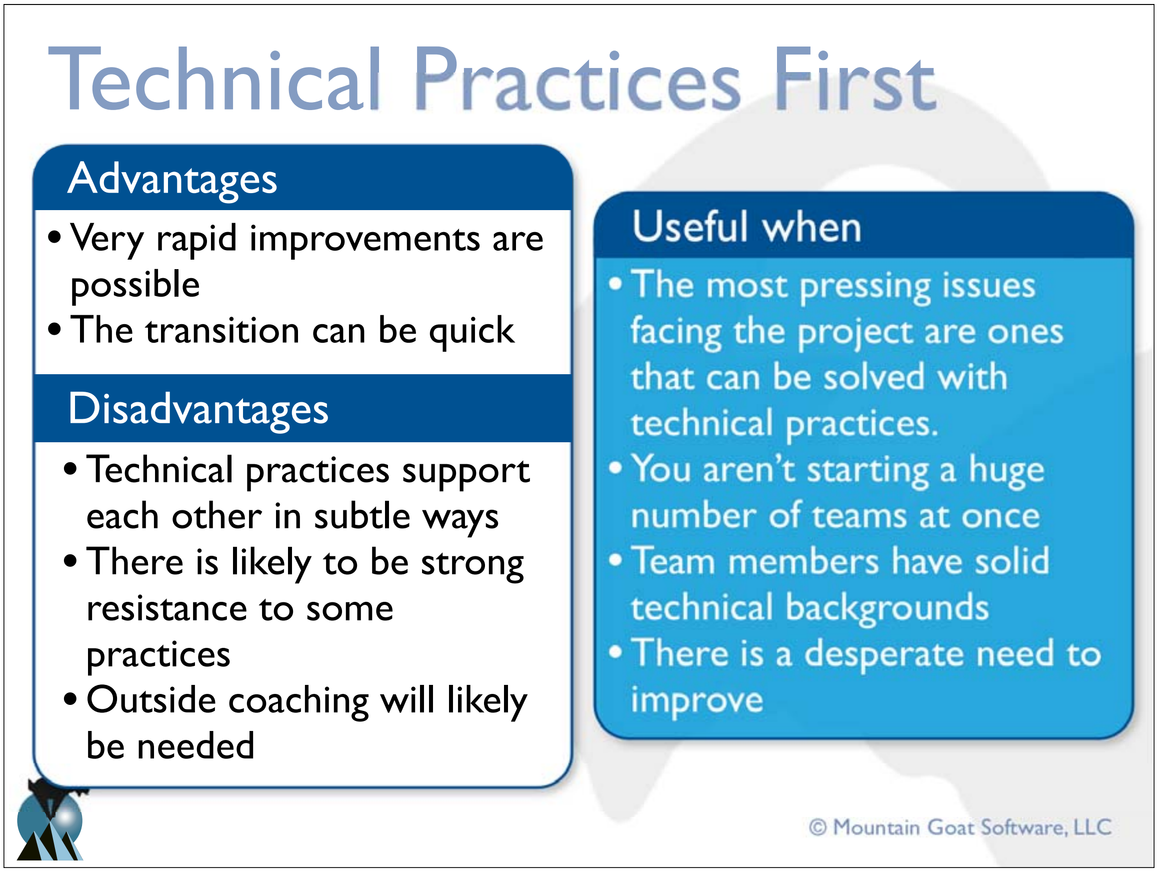
Two types of patterns

Adoption patterns

- Technical practices first
- Iterative first
- Requirements first
- Start small
- All in
- Stealth mode
- Public display of agility
- Impending doom

Expansion patterns

- Split and seed
- Grow and split
- Internal coaching



Technical Practices First

Advantages

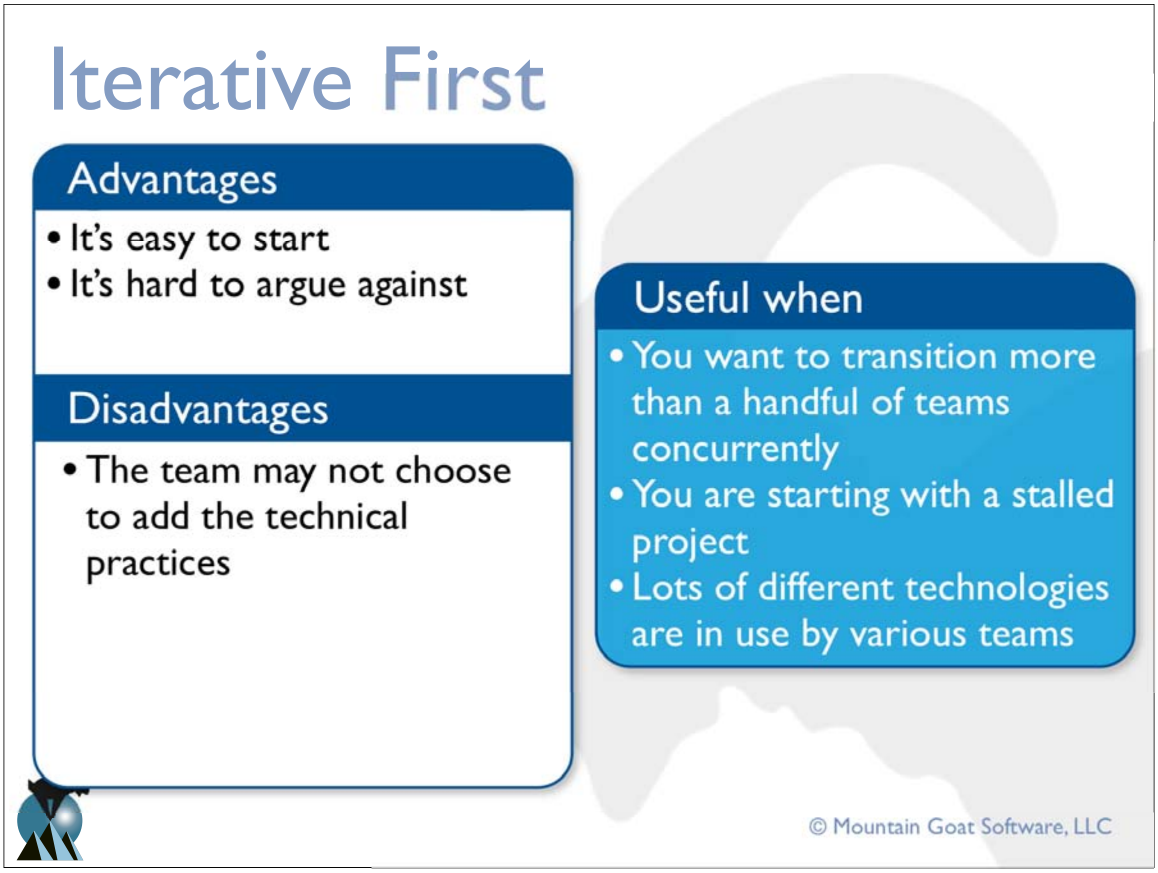
- Very rapid improvements are possible
- The transition can be quick

Disadvantages

- Technical practices support each other in subtle ways
- There is likely to be strong resistance to some practices
- Outside coaching will likely be needed

Useful when

- The most pressing issues facing the project are ones that can be solved with technical practices.
- You aren't starting a huge number of teams at once
- Team members have solid technical backgrounds
- There is a desperate need to improve



Iterative First

Advantages

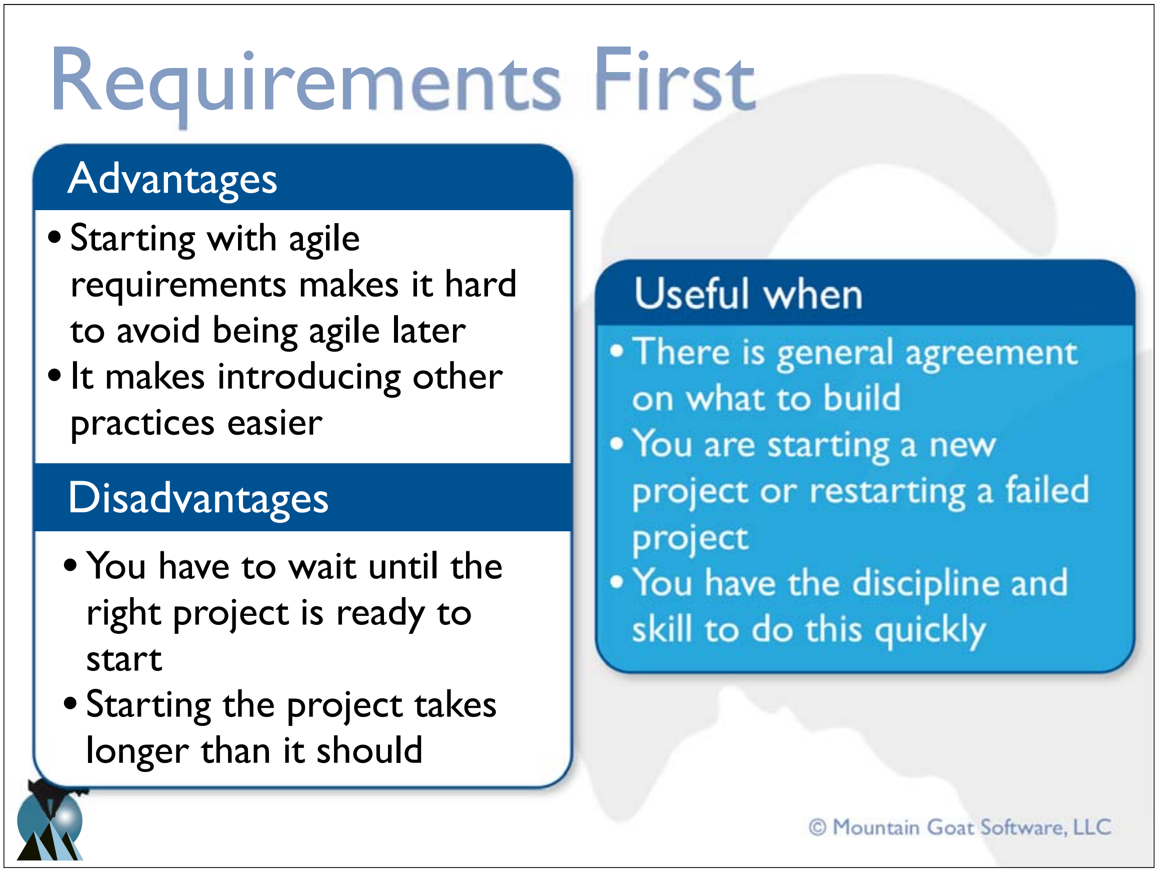
- It’s easy to start
- It’s hard to argue against

Disadvantages

- The team may not choose to add the technical practices

Useful when

- You want to transition more than a handful of teams concurrently
- You are starting with a stalled project
- Lots of different technologies are in use by various teams



Requirements First

Advantages

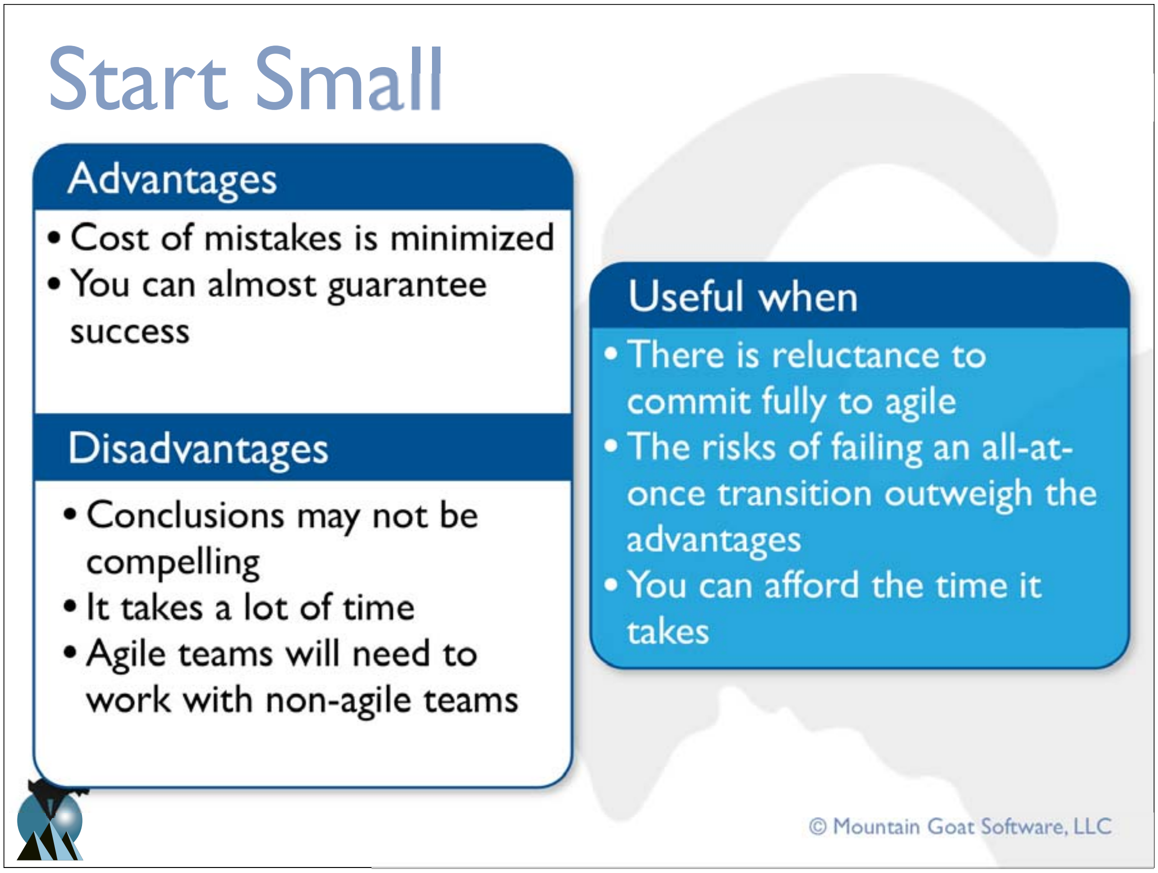
- Starting with agile requirements makes it hard to avoid being agile later
- It makes introducing other practices easier

Disadvantages

- You have to wait until the right project is ready to start
- Starting the project takes longer than it should

Useful when

- There is general agreement on what to build
- You are starting a new project or restarting a failed project
- You have the discipline and skill to do this quickly



Start Small

Advantages

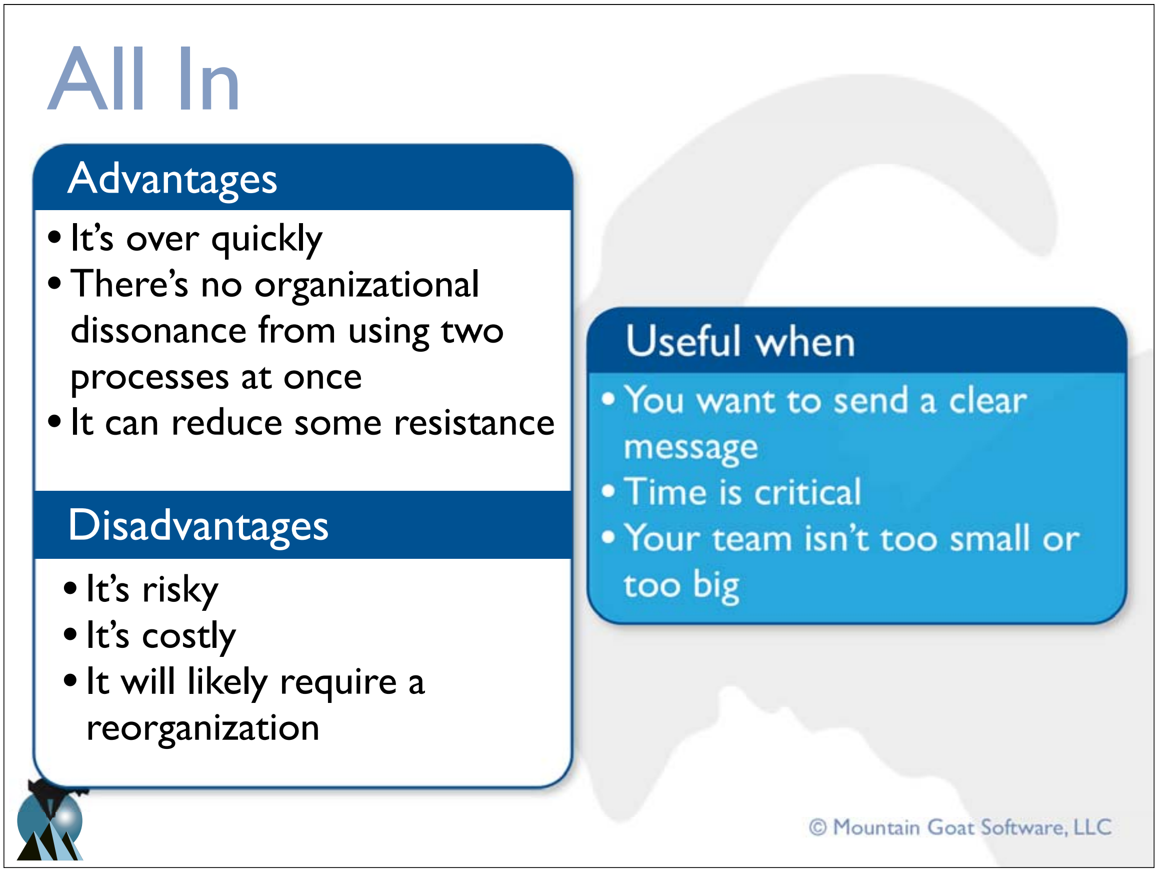
- Cost of mistakes is minimized
- You can almost guarantee success

Disadvantages

- Conclusions may not be compelling
- It takes a lot of time
- Agile teams will need to work with non-agile teams

Useful when

- There is reluctance to commit fully to agile
- The risks of failing an all-at-once transition outweigh the advantages
- You can afford the time it takes



All In

Advantages

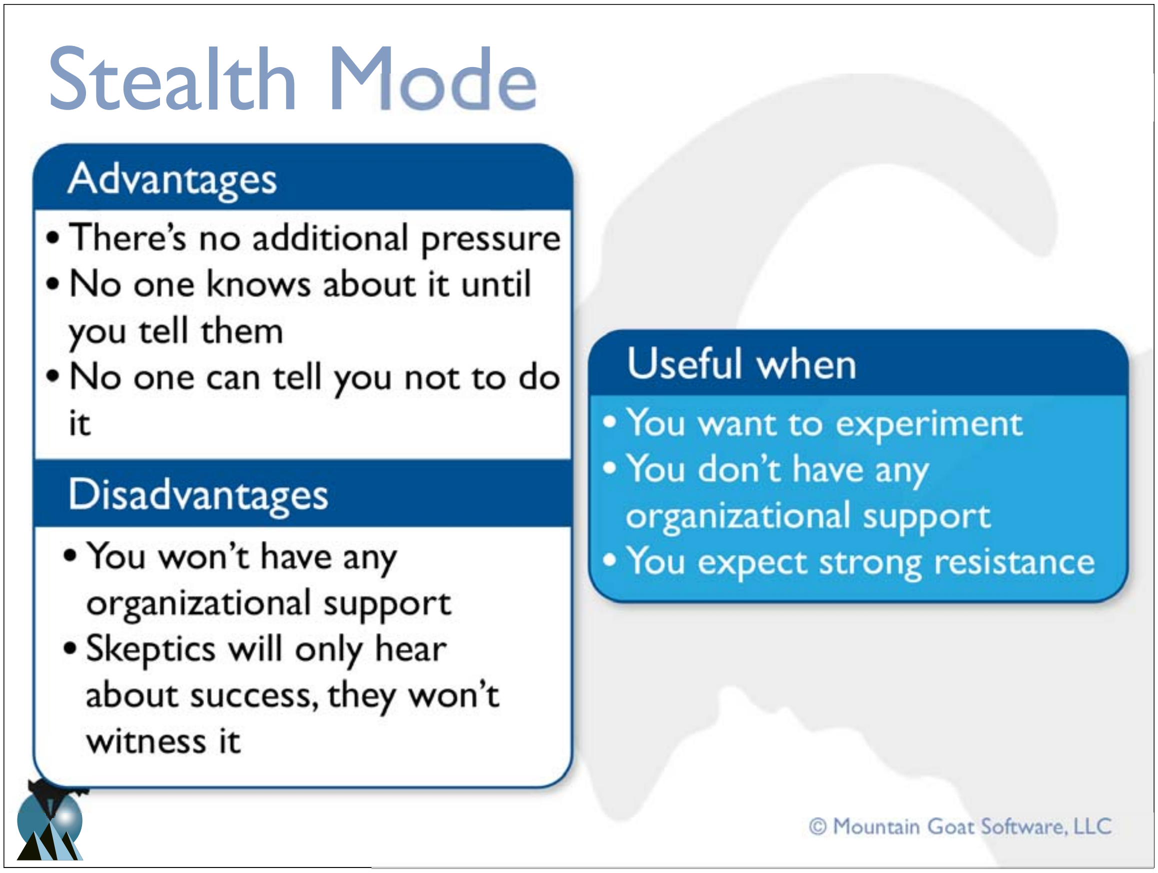
- It’s over quickly
- There’s no organizational dissonance from using two processes at once
- It can reduce some resistance

Disadvantages

- It’s risky
- It’s costly
- It will likely require a reorganization

Useful when

- You want to send a clear message
- Time is critical
- Your team isn’t too small or too big



Stealth Mode

Advantages

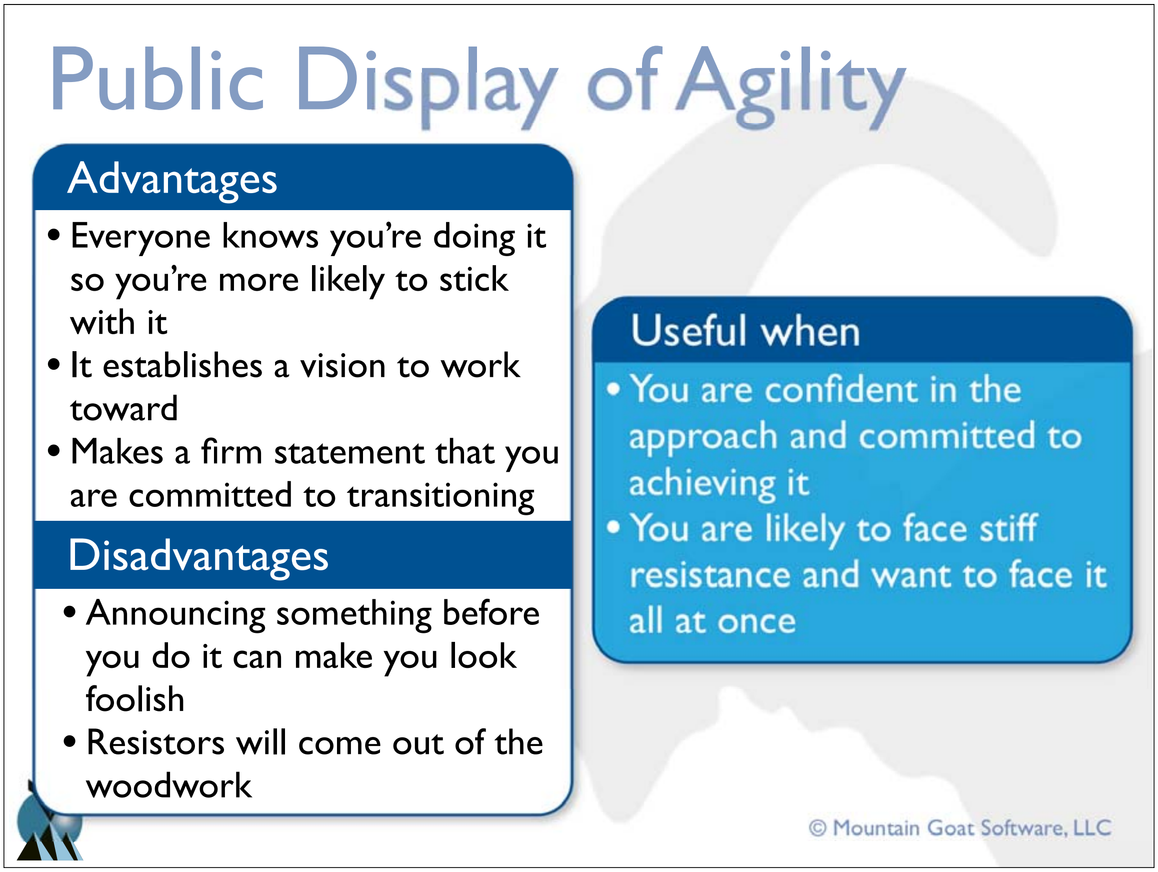
- There's no additional pressure
- No one knows about it until you tell them
- No one can tell you not to do it

Disadvantages

- You won't have any organizational support
- Skeptics will only hear about success, they won't witness it

Useful when

- You want to experiment
- You don't have any organizational support
- You expect strong resistance



Public Display of Agility

Advantages

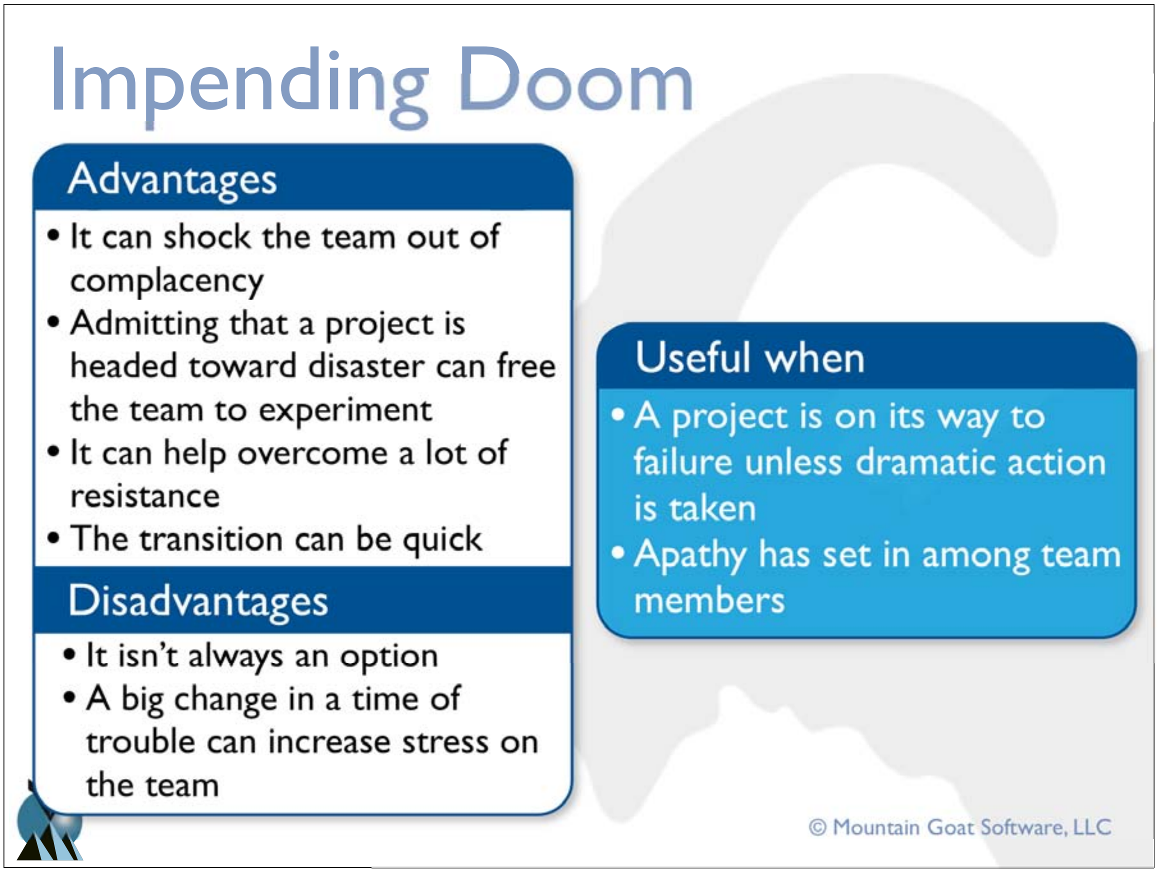
- Everyone knows you're doing it so you're more likely to stick with it
- It establishes a vision to work toward
- Makes a firm statement that you are committed to transitioning

Disadvantages

- Announcing something before you do it can make you look foolish
- Resistors will come out of the woodwork

Useful when

- You are confident in the approach and committed to achieving it
- You are likely to face stiff resistance and want to face it all at once



Impending Doom

Advantages

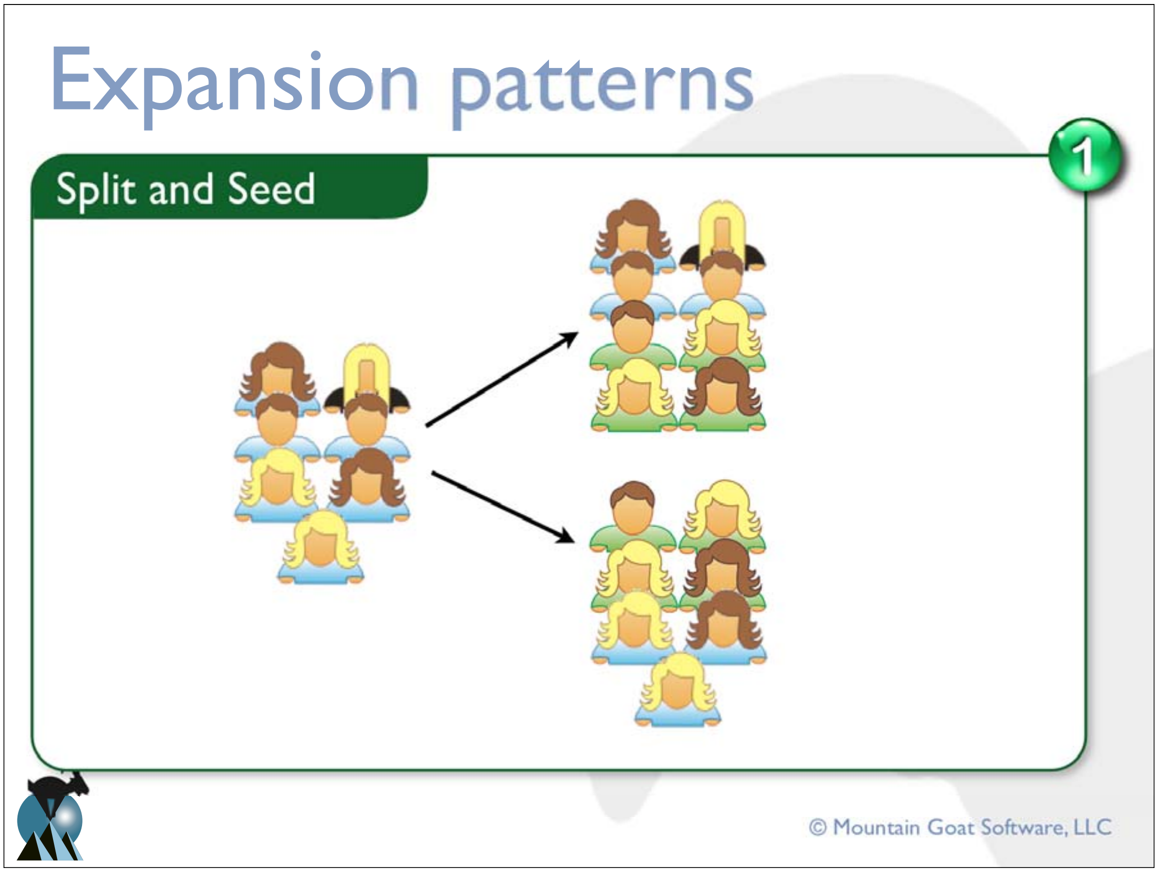
- It can shock the team out of complacency
- Admitting that a project is headed toward disaster can free the team to experiment
- It can help overcome a lot of resistance
- The transition can be quick

Disadvantages

- It isn't always an option
- A big change in a time of trouble can increase stress on the team

Useful when

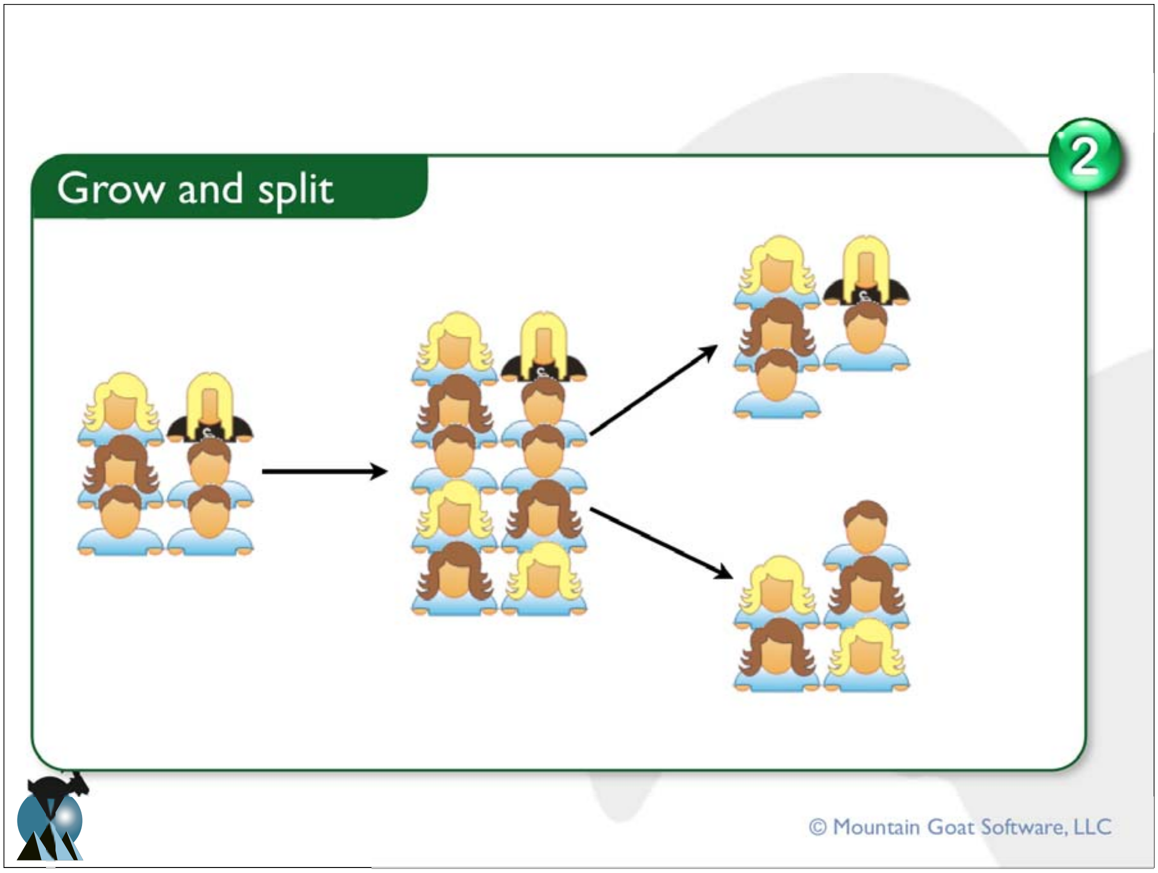
- A project is on its way to failure unless dramatic action is taken
- Apathy has set in among team members



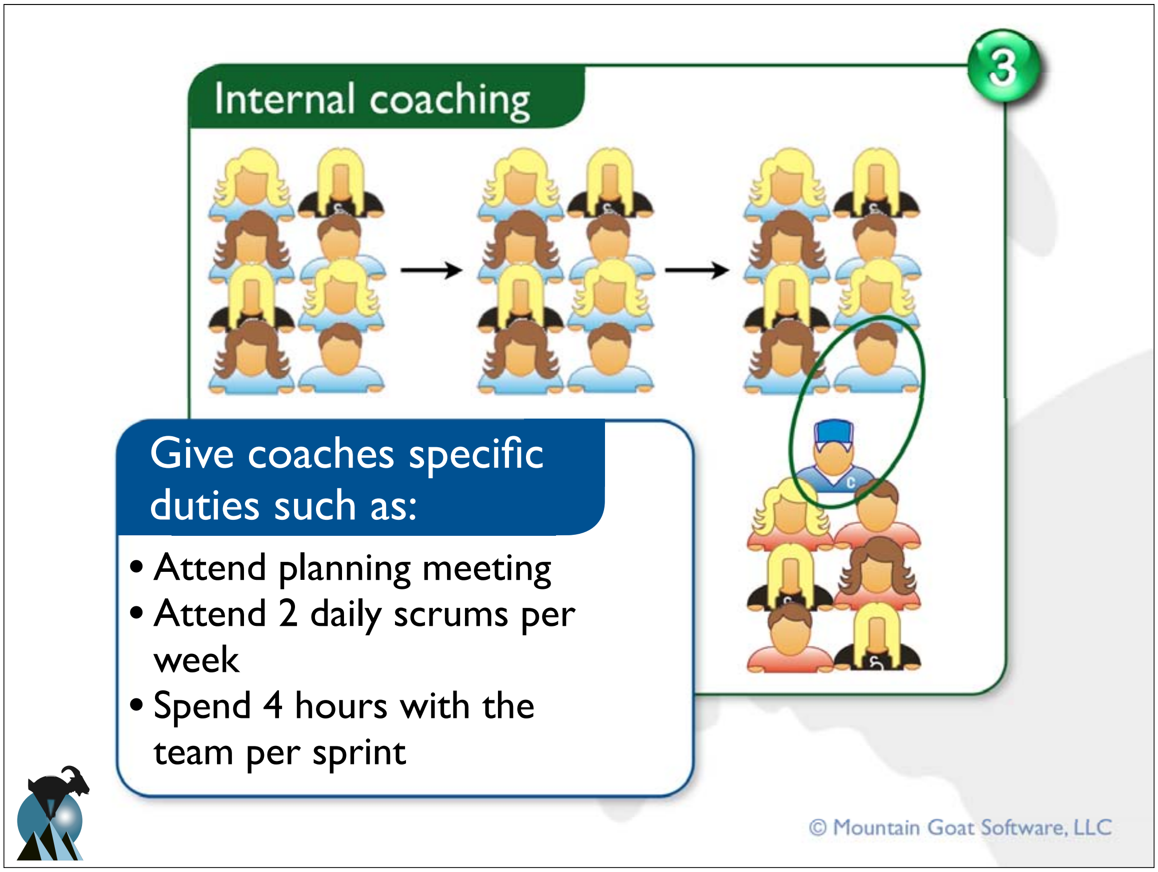
Expansion patterns

Split and Seed

1



Grow and split

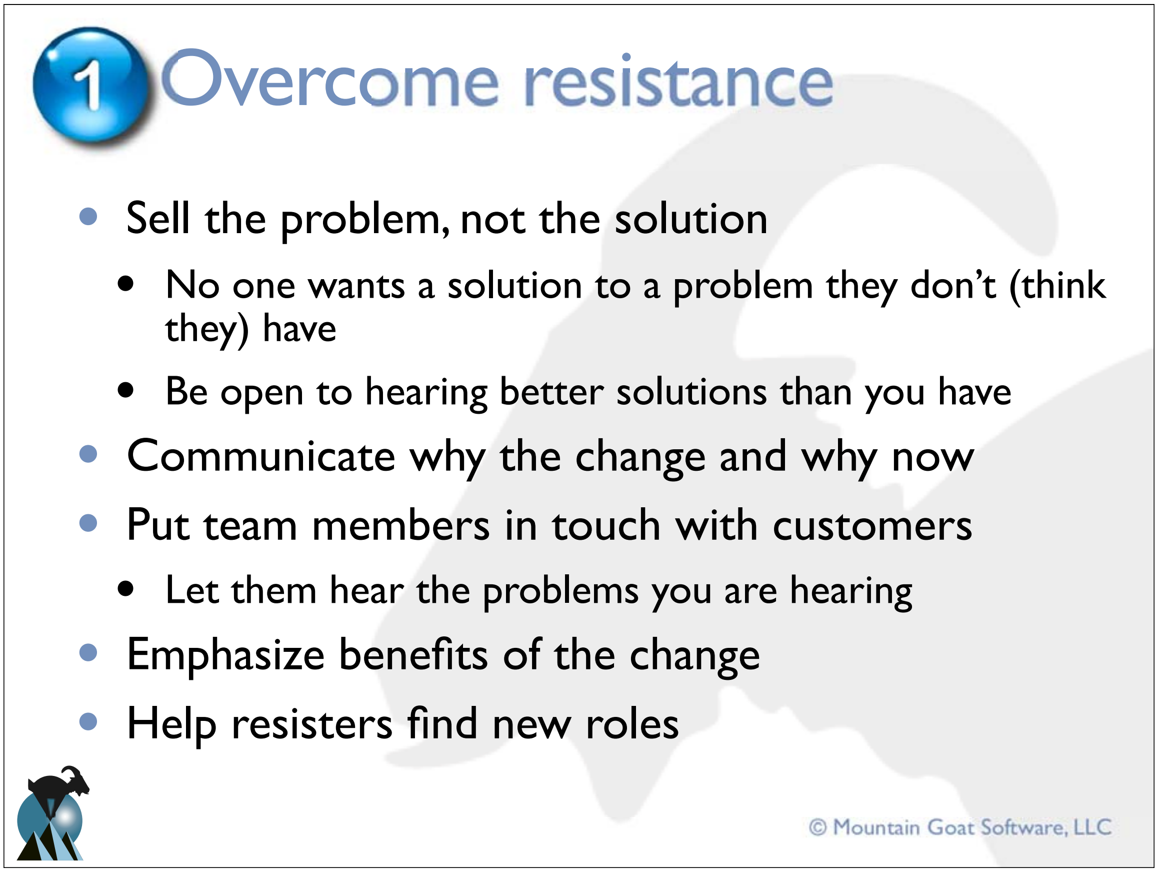


Internal coaching

Give coaches specific duties such as:

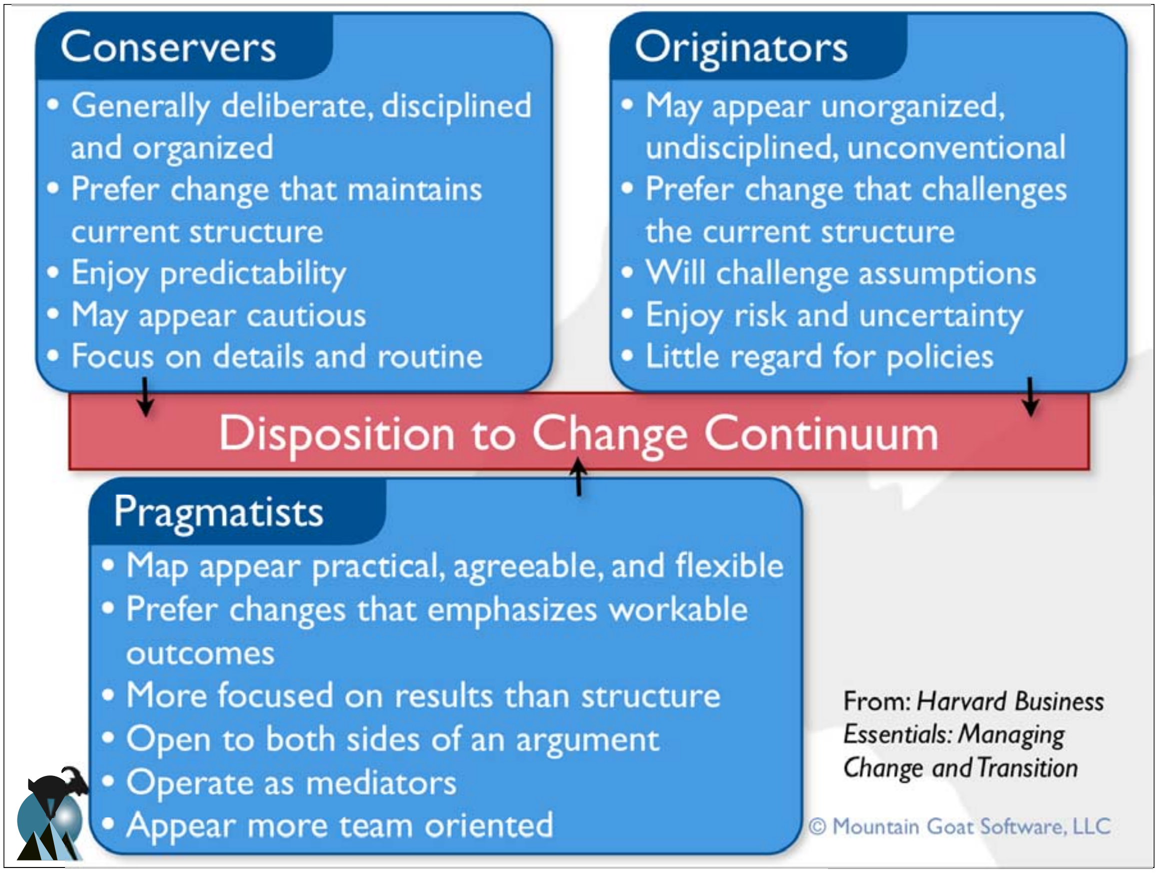
- Attend planning meeting
- Attend 2 daily scrums per week
- Spend 4 hours with the team per sprint

Some Items for Your Early Transition Backlog



1 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

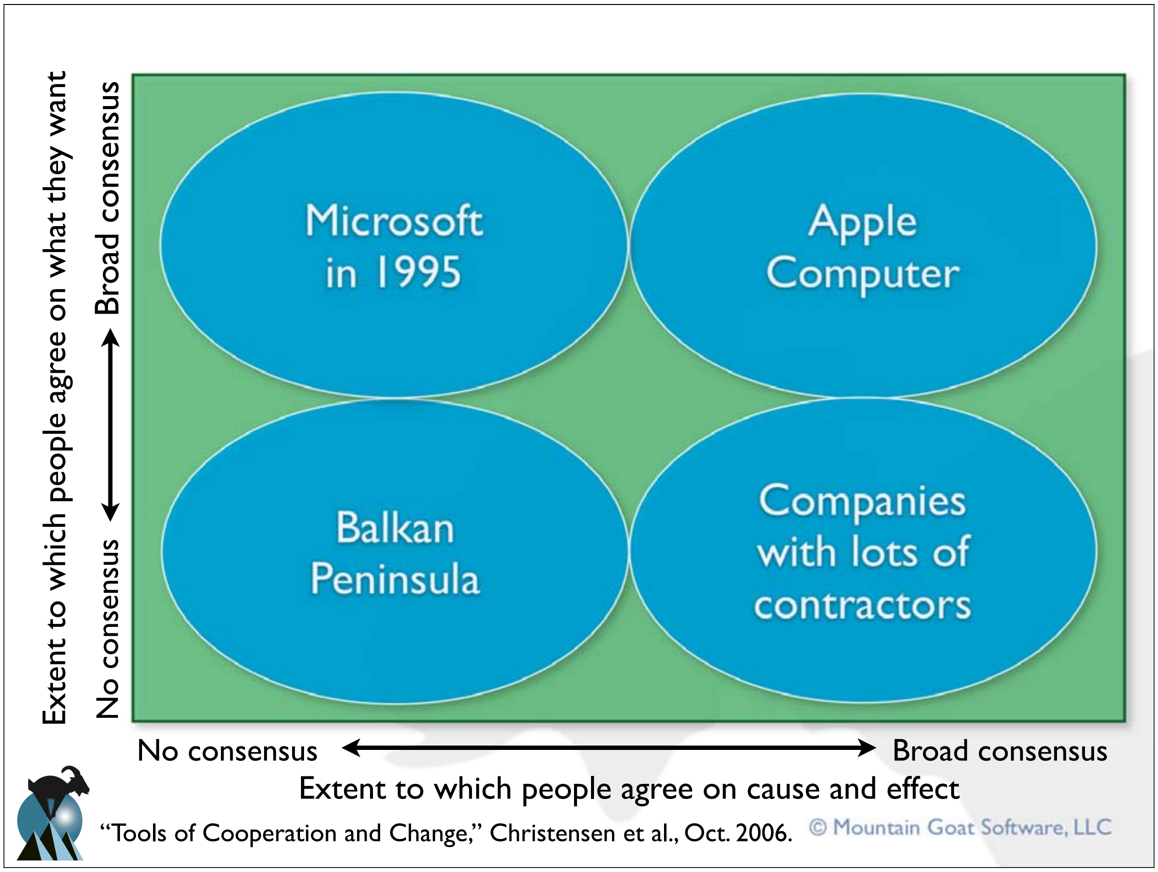
- May appear unorganized, undisciplined, unconventional
- Prefer change that challenges the current structure
- Will challenge assumptions
- Enjoy risk and uncertainty
- Little regard for policies

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*



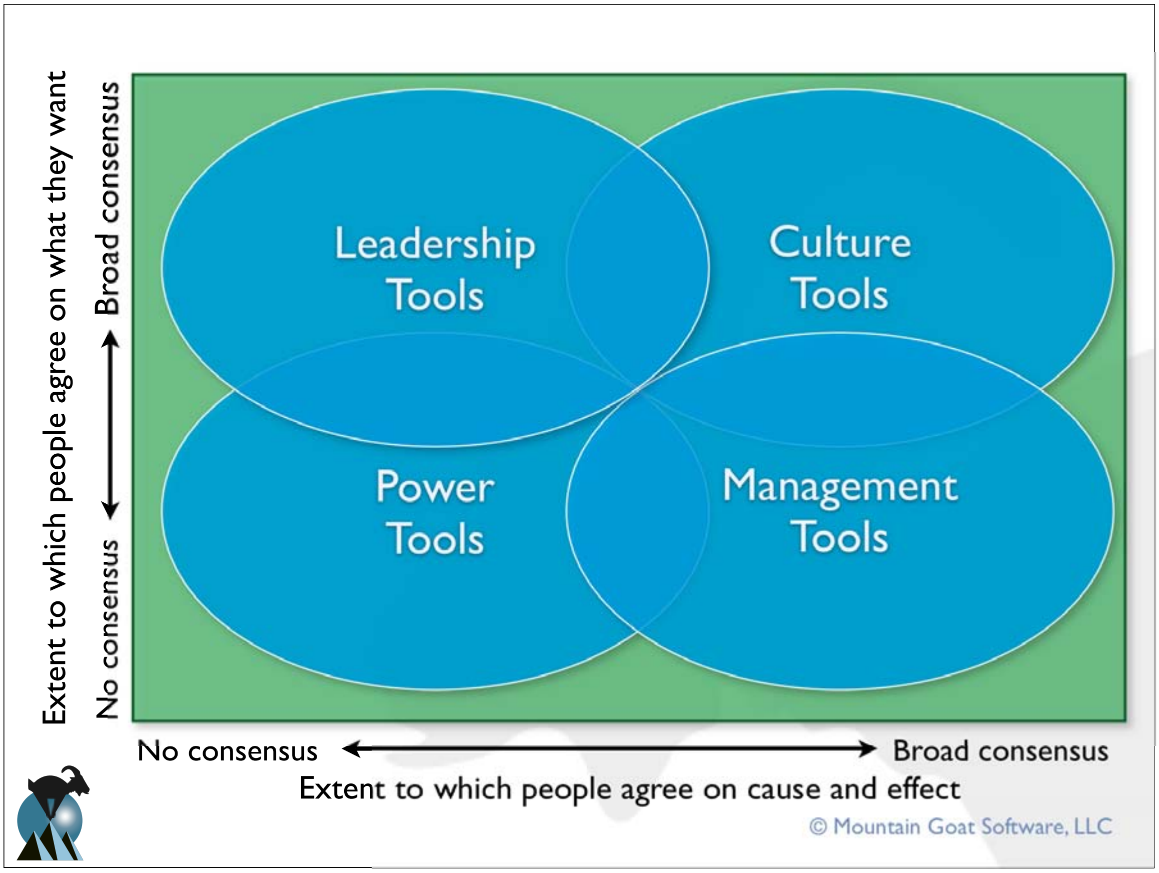
	No consensus ← → Broad consensus	
Broad consensus	Microsoft in 1995	Apple Computer
No consensus	Balkan Peninsula	Companies with lots of contractors

Extent to which people agree on what they want

No consensus ← → Broad consensus

Extent to which people agree on cause and effect

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



Extent to which people agree on what they want

No consensus ↔ Broad consensus

Leadership Tools

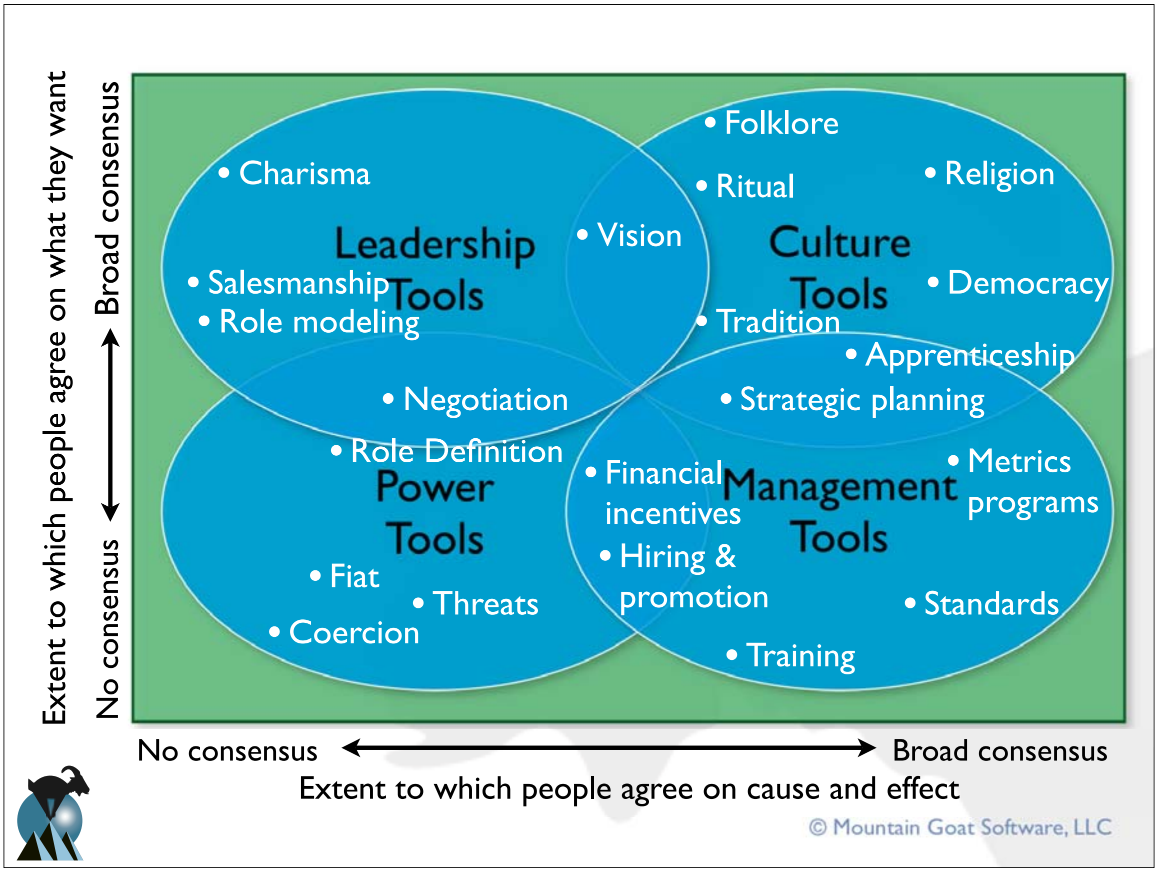
Culture Tools

Power Tools

Management Tools

No consensus ↔ Broad consensus

Extent to which people agree on cause and effect



Extent to which people agree on what they want

No consensus ↔ Broad consensus

Leadership Tools
- Charisma
- Salesmanship
- Role modeling

Vision

Culture Tools
- Folklore
- Ritual
- Religion
- Democracy
- Tradition

Apprenticeship

Negotiation

Strategic planning

Role Definition

Power Tools
- Fiat
- Threats
- Coercion

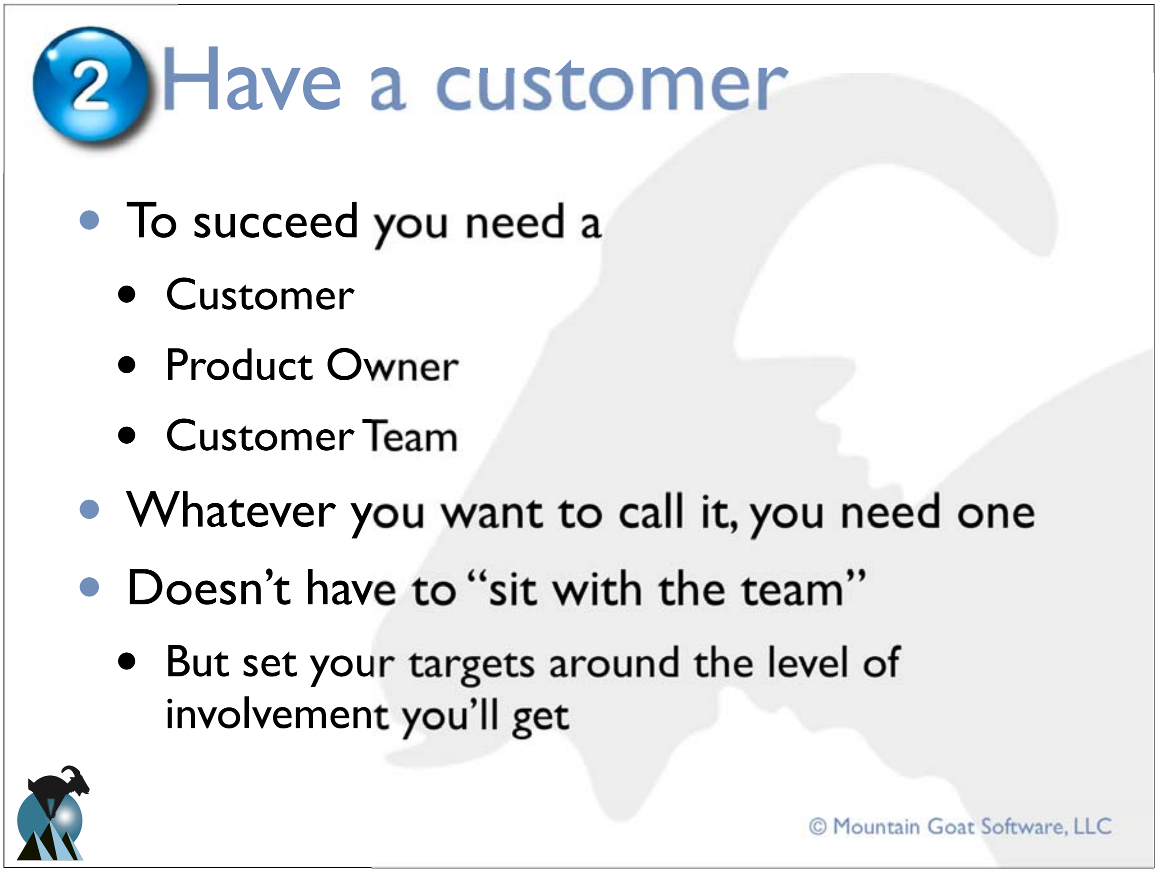
Financial incentives

Hiring & promotion

Management Tools
- Metrics programs
- Standards
- Training

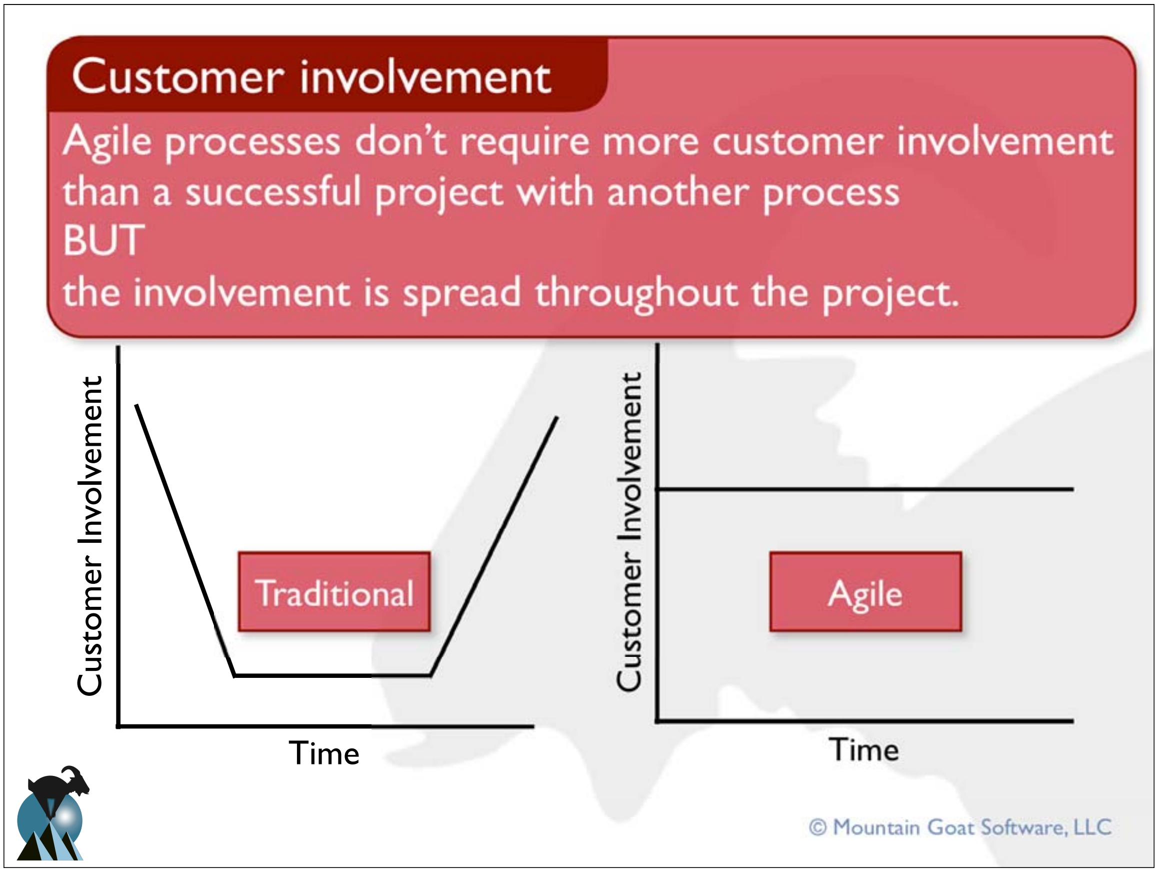
No consensus ↔ Broad consensus

Extent to which people agree on cause and effect



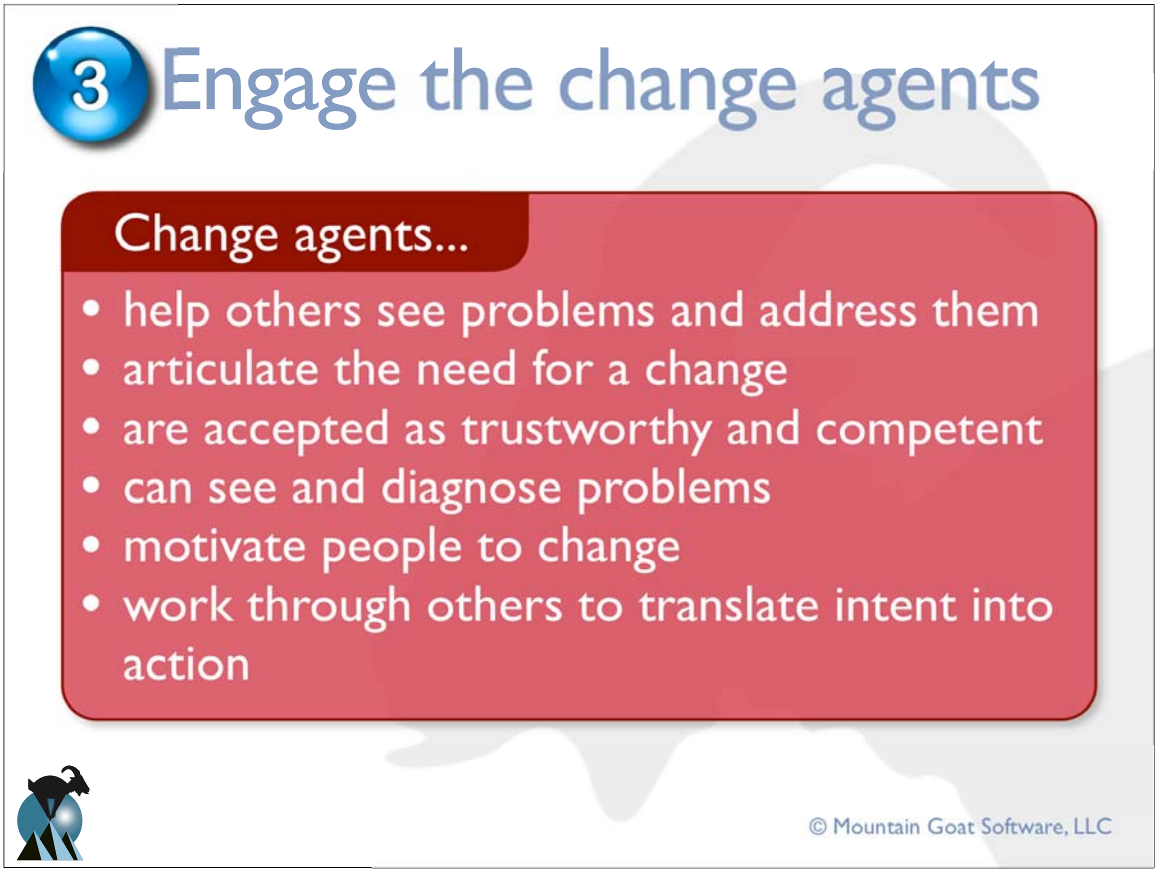
2 Have a customer

- 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

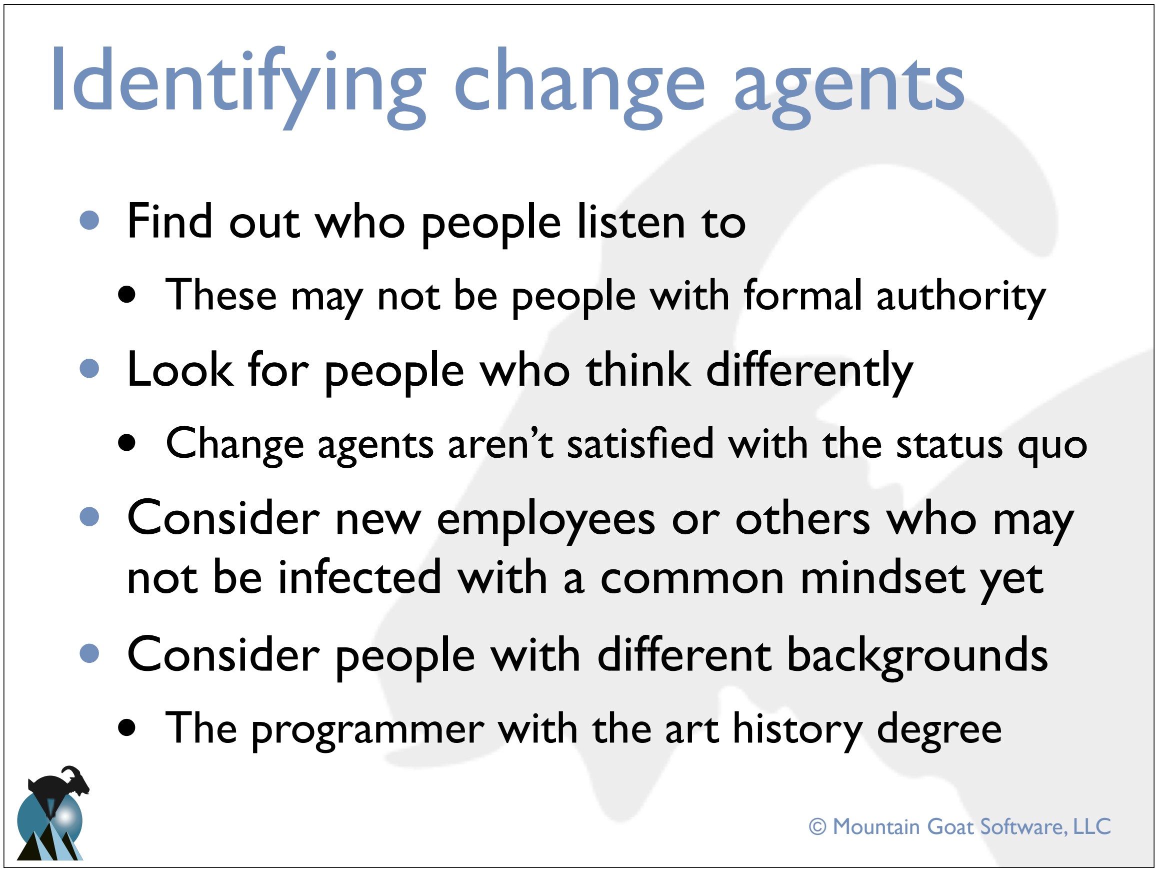
Agile processes don't require more customer involvement than a successful project with another process
BUT
the involvement is spread throughout the project.



3 Engage the change agents

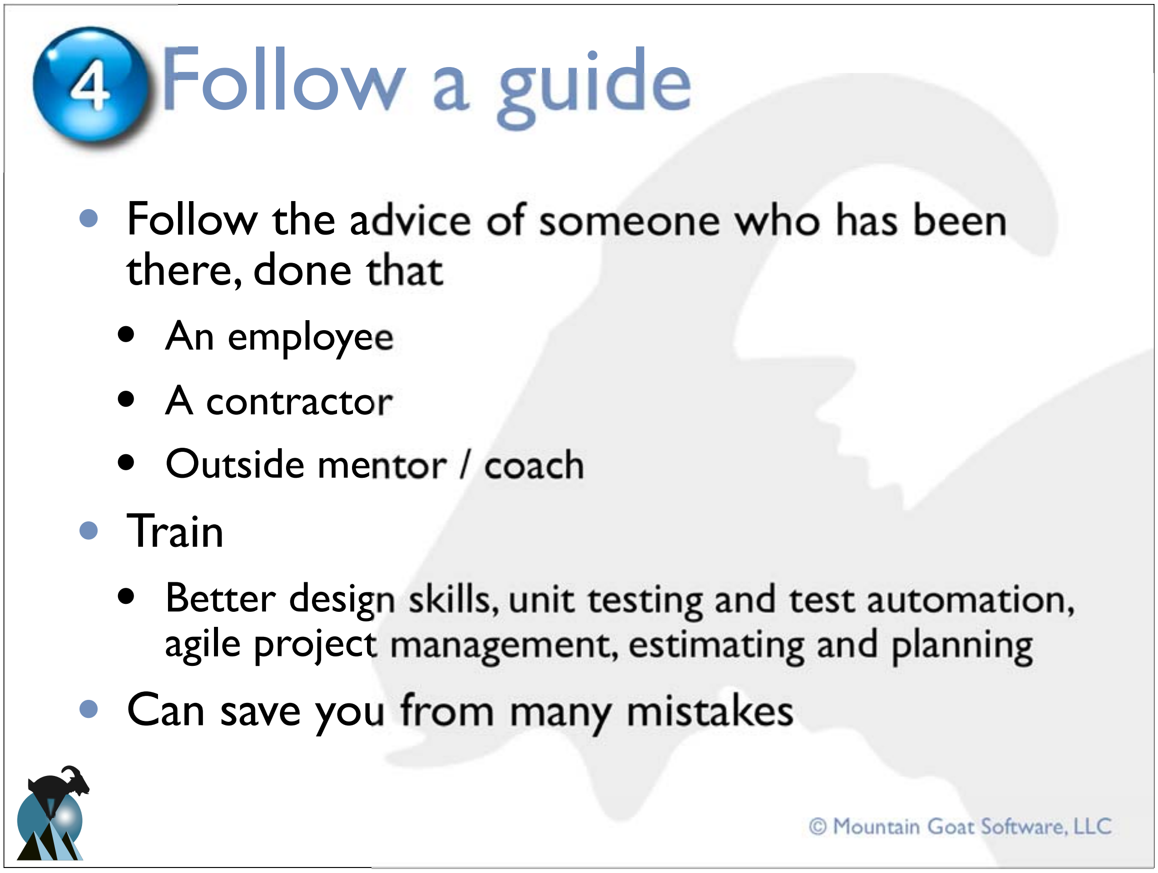
Change agents...

- 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



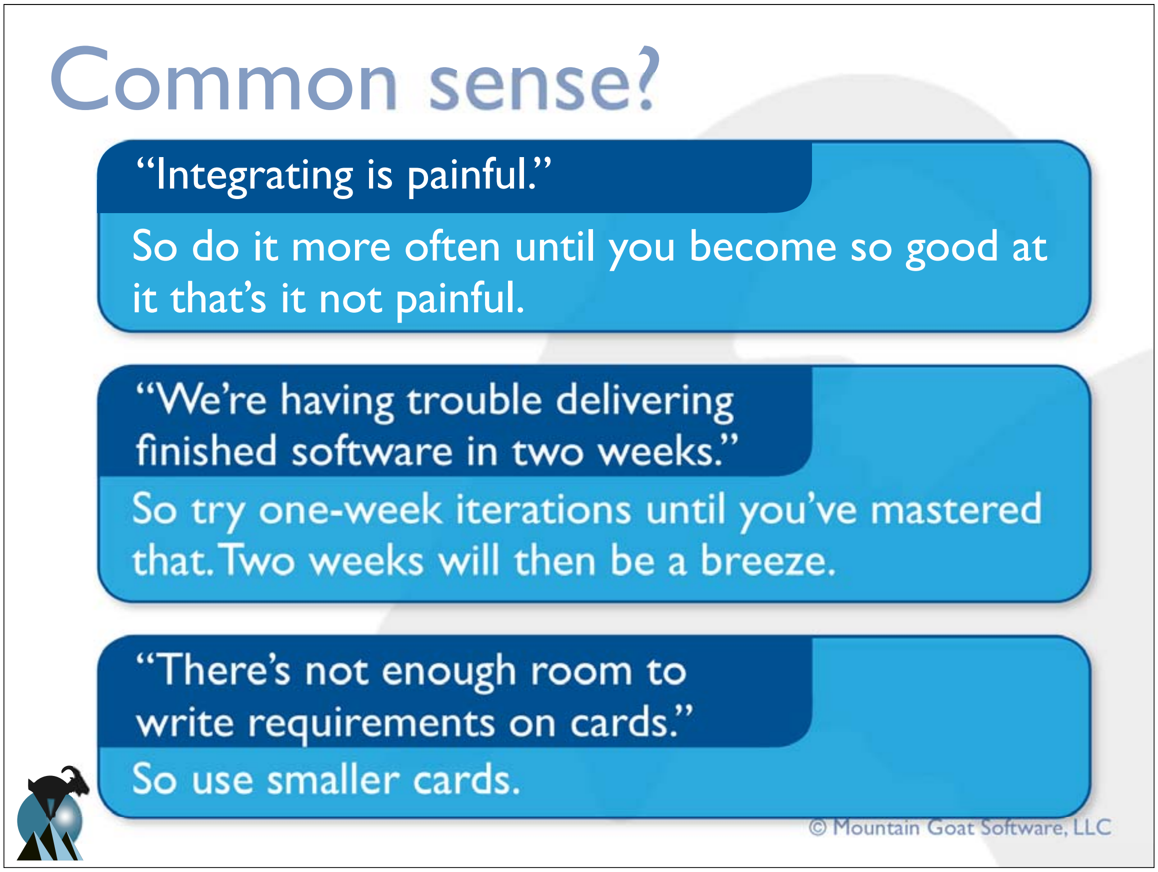
Identifying change agents

- 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



4 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.

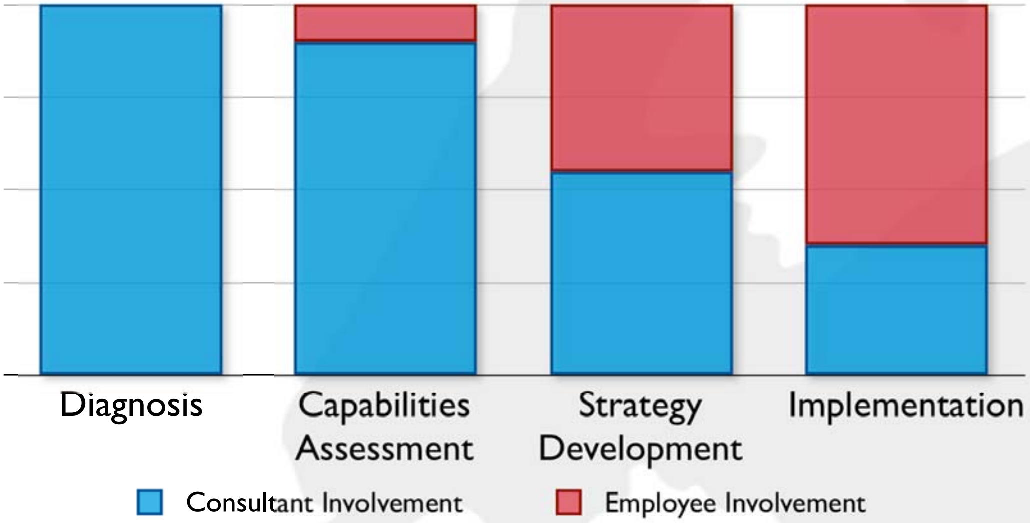
"We're having trouble delivering finished software in two weeks."

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



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

Mike Cohn contact info

The Addison-Wesley Signature Series
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