

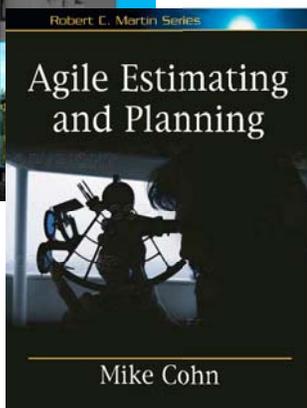
Succeeding With Agile: A Guide to Transitioning

Mike Cohn
14 November, 2007



1

Mike Cohn - background



2

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

5

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

6

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



© Mountain Goat Software, LLC

7

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



© Mountain Goat Software, LLC

8

“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*

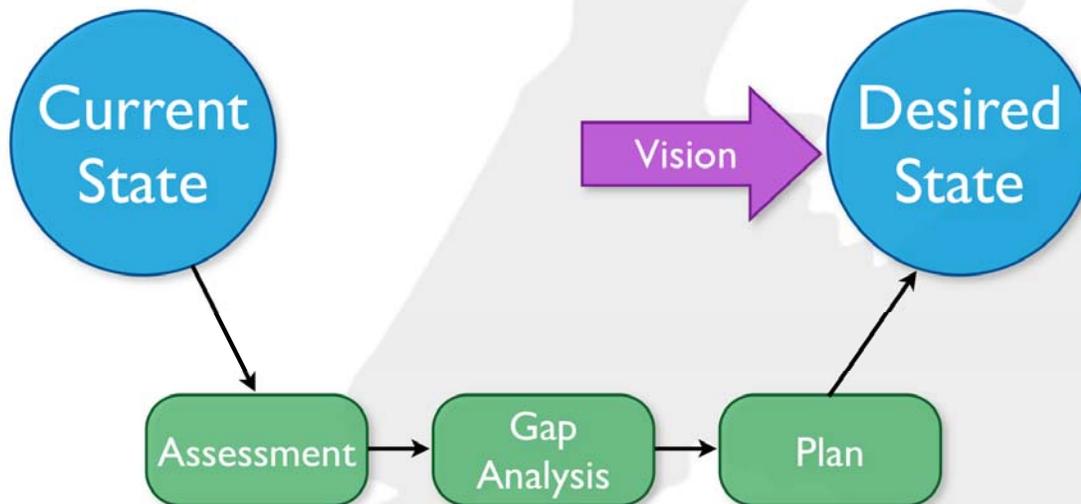


“This machine imagery [Newtonian view] leads to the belief that studying the parts is the key to understanding the whole. Things are taken apart, dissected literally or figuratively...and then put back together without any significant loss. The assumption is that the more we know about the workings of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



© Mountain Goat Software, LLC

11

We need a different mental model

- The organization as a **C**omplex **A**daptive **S**ystem (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



© Mountain Goat Software, LLC

12

Differing views of success

Newtonian view

Success =
closing the gap with the
desired state

CAS view

Success =
achieving a good fit with
the environment

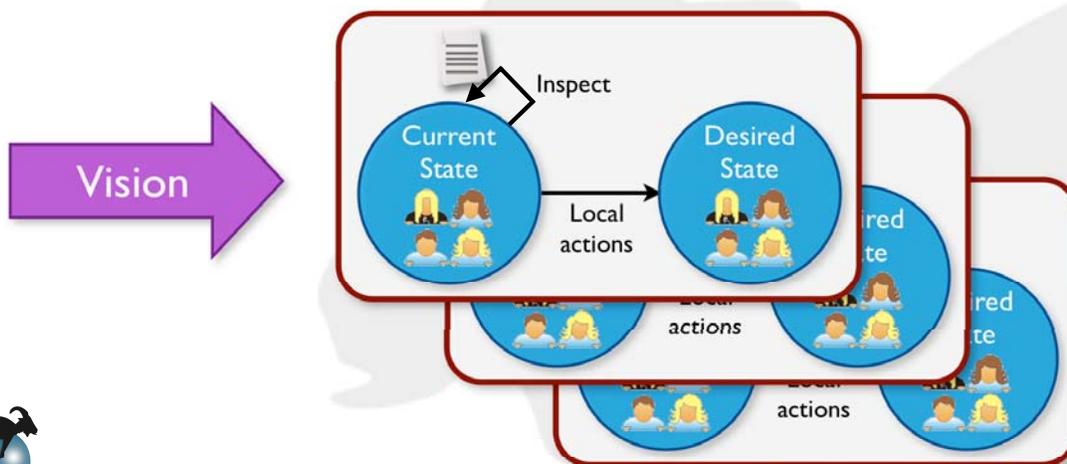


© Mountain Goat Software, LLC

13

Local goals and gaps

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



© Mountain Goat Software, LLC

14

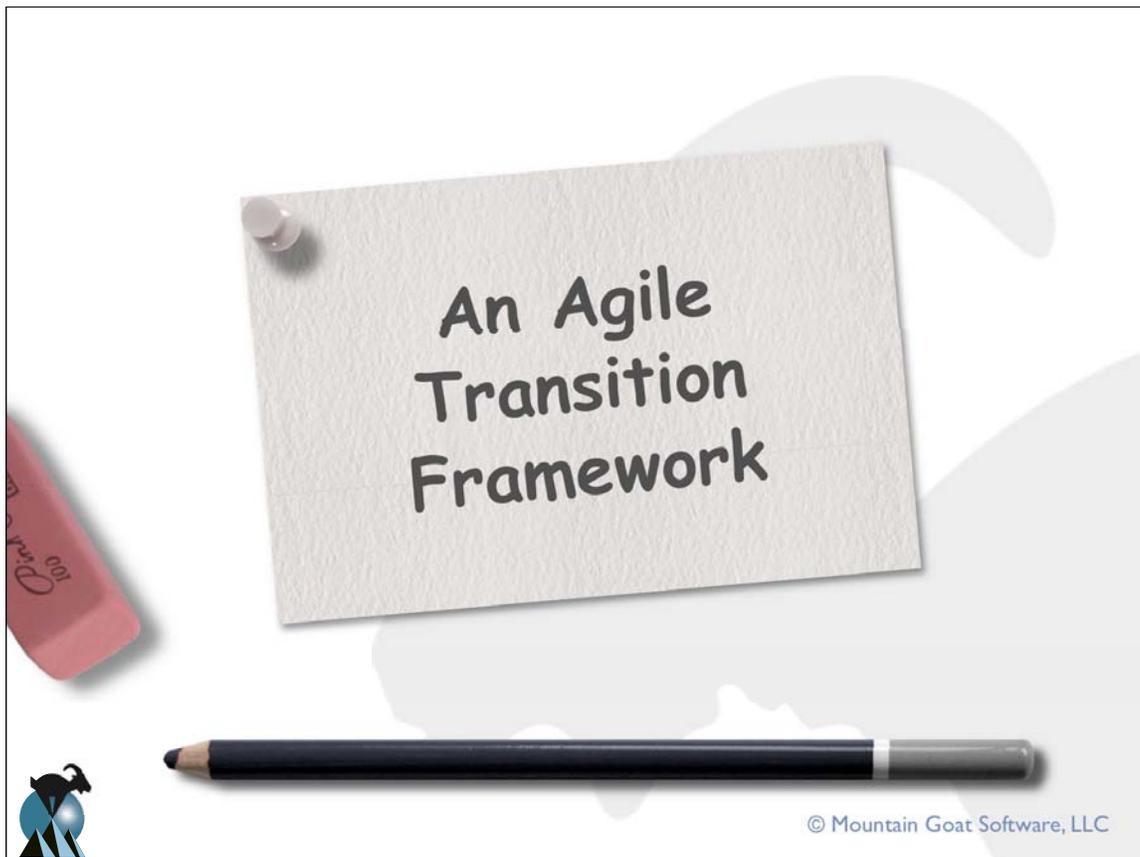
Traditional model of change	Complex, adaptive model of change
Behavior is predictable and controllable	Behavior is unpredictable and uncontrollable
Direction is determined by a few leaders.	Direction is determined through emergence and by many people
Every effect has a cause	Every effect is also a cause
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*.

© Mountain Goat Software, LLC

15



© Mountain Goat Software, LLC

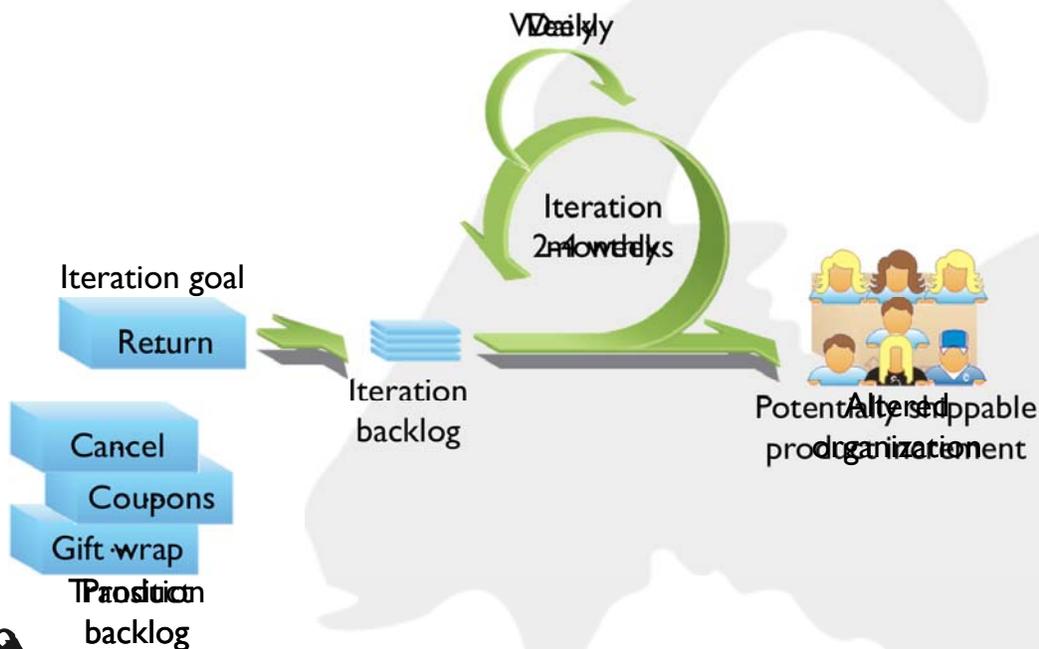
16

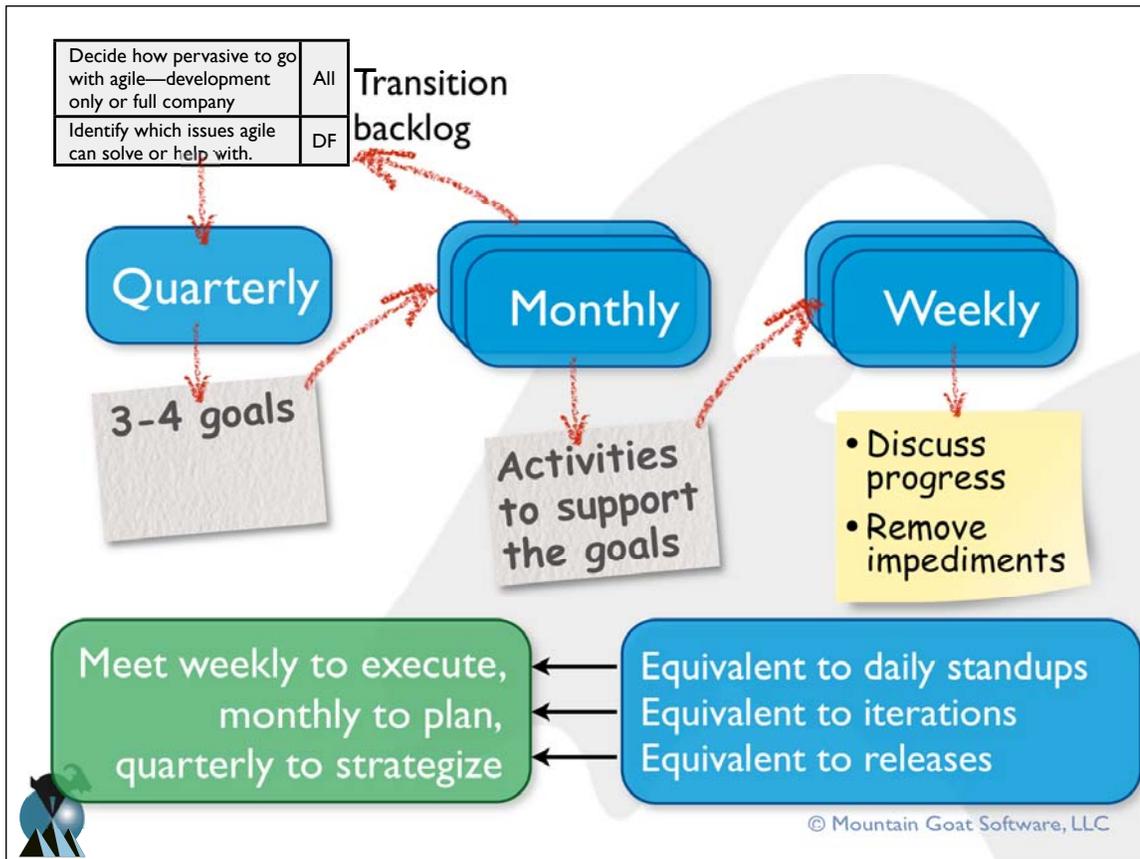
- On projects we learn we cannot precisely anticipate:
 - our users' requirements
 - how long it will take to develop a feature or entire system
 - which design will be best
 - the set of tasks necessary to develop a feature
- So we devise alternative approaches:
 - Rather than ask for upfront specs, 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

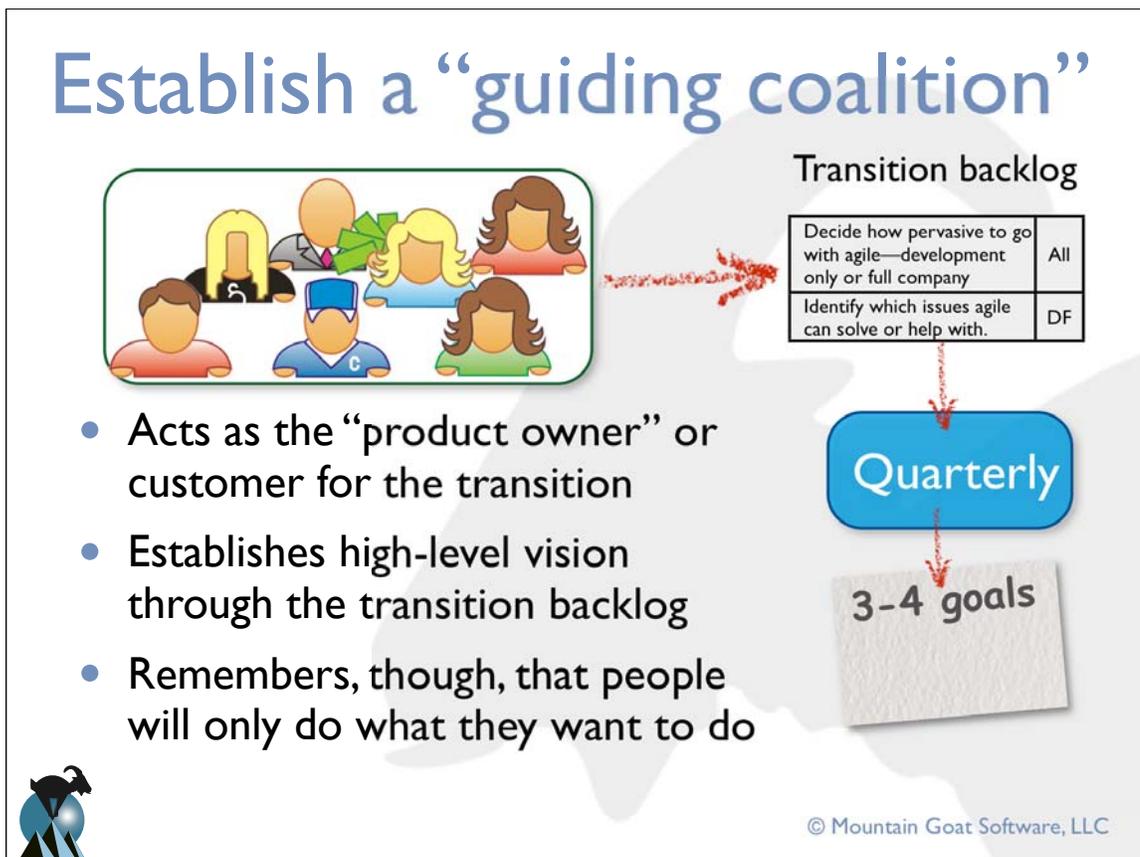


An agile transition process





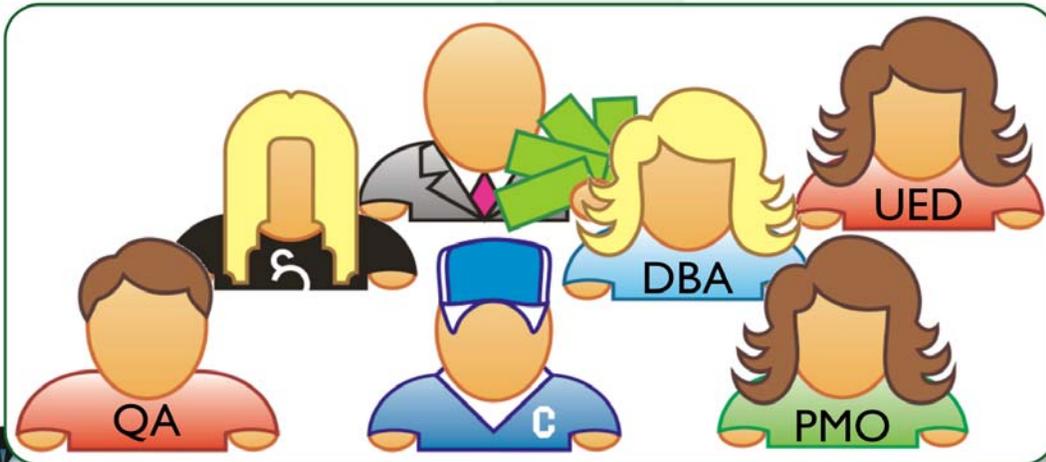
19



20

Guiding coalition members

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



© Mountain Goat Software, LLC

21

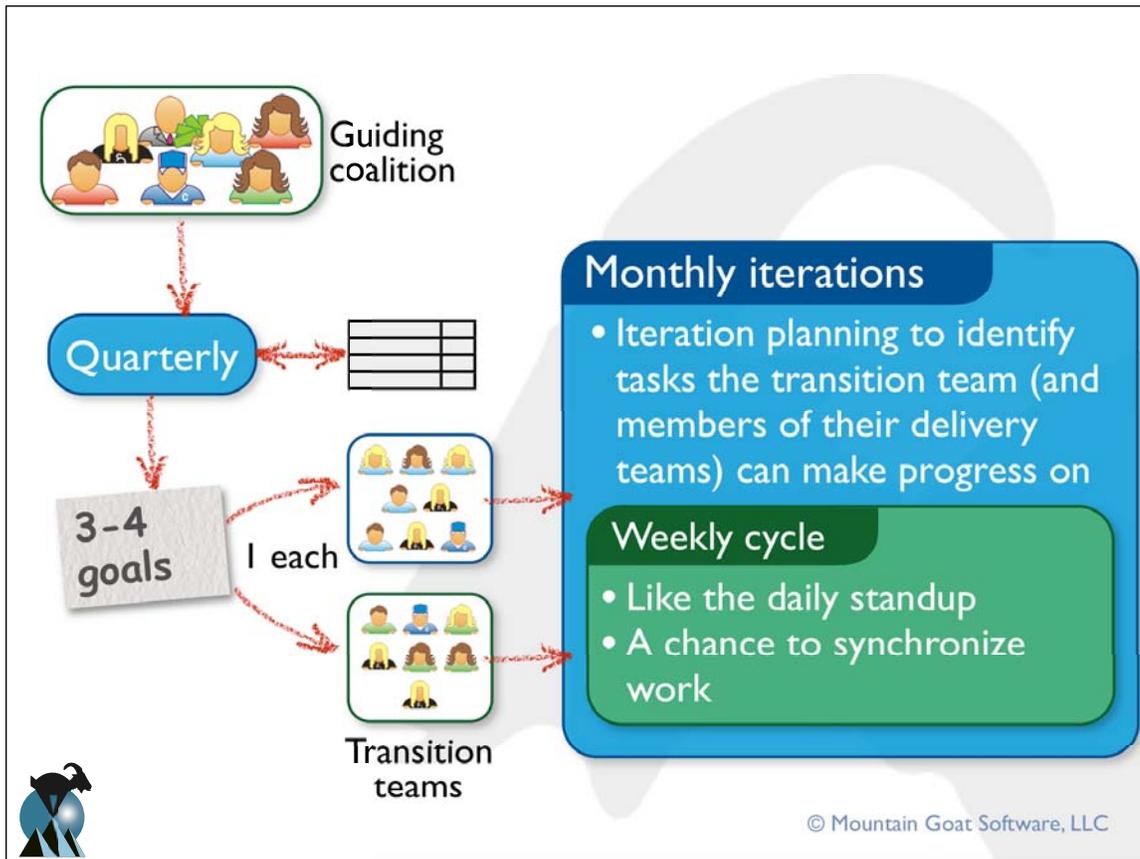
Transition teams

- Usually multiple teams pursuing different goals
- Organized around achieving specific goals in the organization
 - e.g., test automation or user experience design
- Some teams in an organization will be organic
 - Individuals notice something needs to be achieved
- Others will be formally-sponsored
 - Guiding coalition puts someone in charge of achieving a goal that hasn't been picked up
 - Usually best only if an organic team doesn't form



© Mountain Goat Software, LLC

22



23

Transition team members

- Try to form these teams organically
 - Possible with a point person to start the team
 - True product owner for the team is the guiding coalition
 - But this starting person acts as a combination day-to-day product owner and ScrumMaster
- Initial membership
 - Start with 1-3 members who “get it”
 - Ask each of those members to pick 1-2 more



24

Transition team member considerations

- 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 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)



Using the CDE model

- When stuck thinking about how to nudge the organization think of the:
 - **C**ontainers
 - formal teams, informal teams, clarify (or not) expectations
 - **D**ifferences
 - Dampen or amplify them within or between containers
 - **E**xchanges
 - Insert new exchanges, new people, new techniques or tools



Containers

- Enlarge or shrink teams
- Enlarge or shrink the responsibility boundary of teams
- Change team membership
- Create new teams or groups



Differences

- Don't require consensus
 - Creativity comes from tension
 - Quiet disagreement is not as good as fierce debate that leads to behavior change
- Ask hard questions
 - Then expect teams to find solutions



Transforming exchanges

- Encourage communication between teams and groups
 - Who isn't talking that should?
- Add or remove people from exchanges
 - Change reporting relationships
 - Relocate people
- Encourage learning





You are the ScrumMaster or coach...

- The next set of slides describes some teams with some trouble spots. Think about how you might help them by changing their **Containers**, amplifying or dampening **Differences**, or changing their **Exchanges**.
- For each case, identify at least one thing you'd do.
- Note whether you are tweaking their Container, Differences, or Exchanges. (You might be affecting more than one.)



1

The team consists of four developers, two testers, a database engineer and you. The developers and testers are not working well together. Developers work in isolation until two days are left in the iteration. They then throw the code “over the wall” to the testers.

2

The team is failing to deliver potentially shippable software at the end of each iteration. None of the items they start are 100% finished. Their close but work is always left to be done in the next iteration.





3

The team seems to be consistently undercommitting during iteration planning. They finish the work they commit but it doesn't seem like much. The product owner hasn't complained yet but you're worried she will soon.

4

Your organization has 20 different agile teams. Each team has its own testers who are starting to go in different directions in terms of preferred tools and approaches.



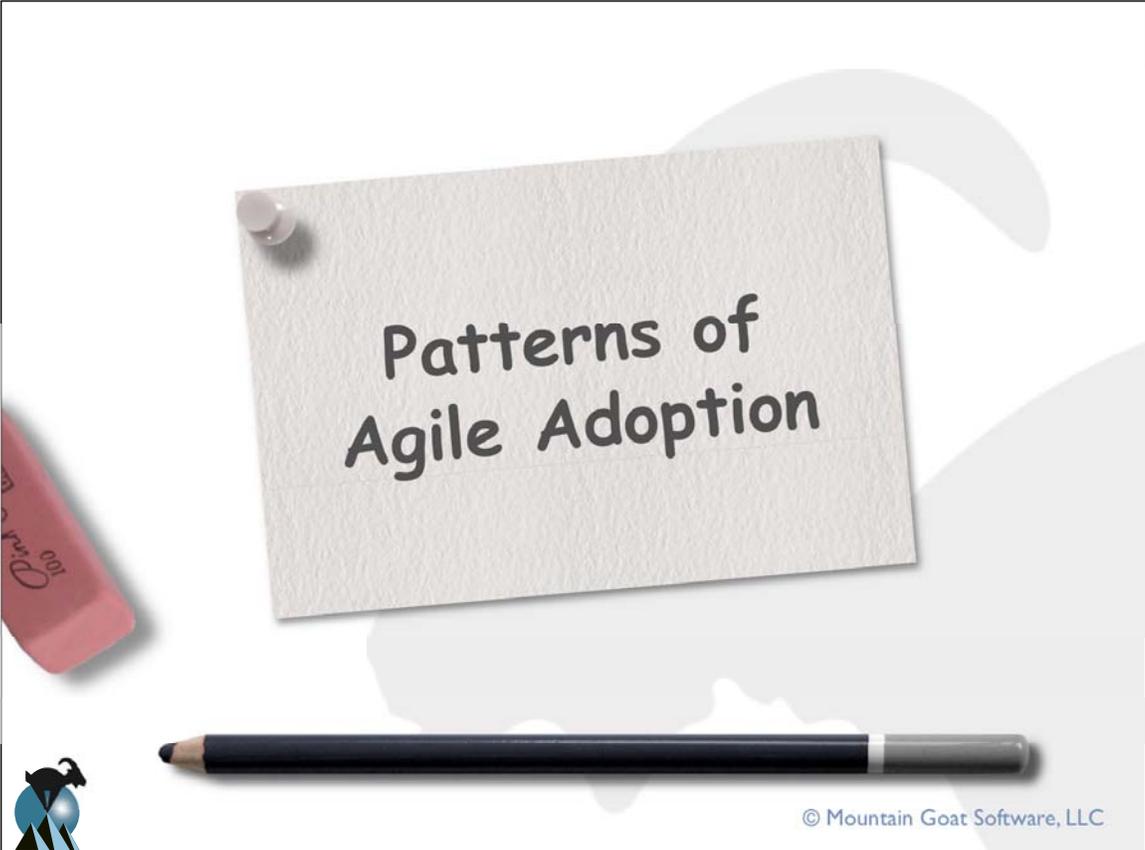
5

Jeff, a senior developer, is very domineering. During iteration planning the team defers to him on every decision even though he is a horrible estimator. You notice the glances that other team members exchange when he suggests very low estimates on some tasks.

6

You are responsible for two teams. Team members on one discuss all sides of various issues before making a decision. This has been working well. On the other team, discussions drag on endlessly because they pursue absolute consensus in all cases.





Patterns of Agile Adoption

© Mountain Goat Software, LLC

39

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

© Mountain Goat Software, LLC

40

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
- Resisters 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

© Mountain Goat Software, LLC

47

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

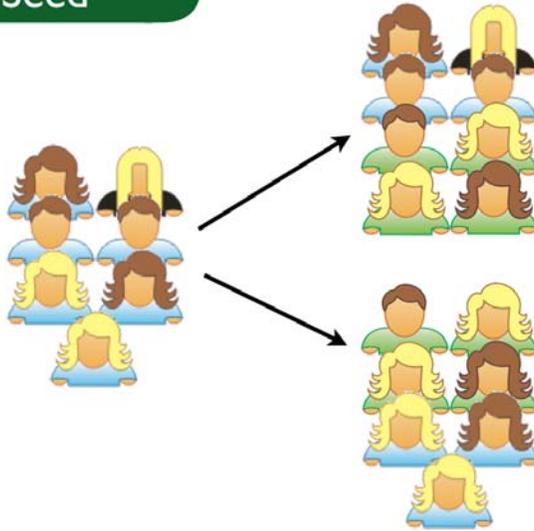
© Mountain Goat Software, LLC

48

Expansion patterns

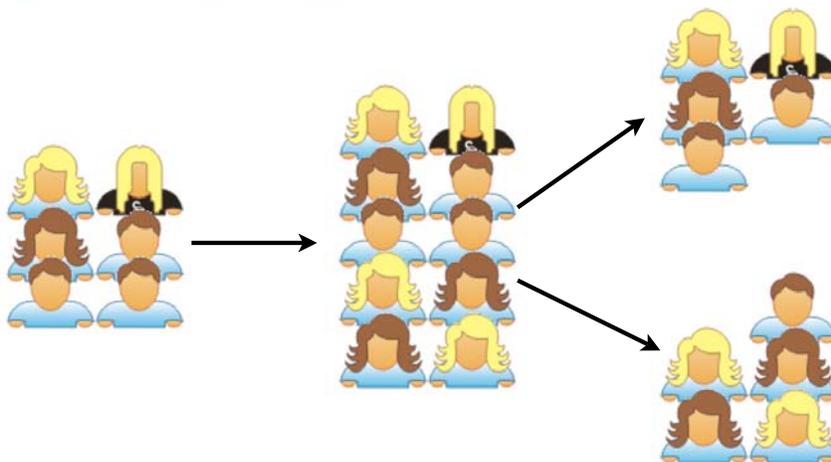
1

Split and Seed



Grow and split

2



Internal coaching

3

Give coaches specific duties such as:

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

© Mountain Goat Software, LLC

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.

Traditional

Agile

© Mountain Goat Software, LLC

Identifying change agents

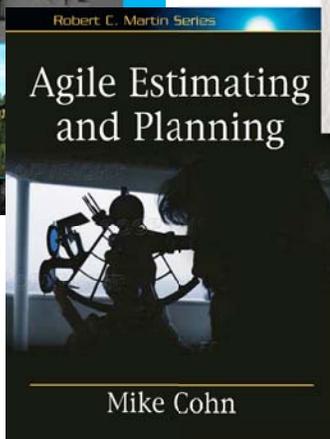
- Find out who people listen to
 - These may not be people with formal authority
- Look for people who think differently
 - Change agents aren't satisfied with the status quo
- Consider new employees or others who may not be infected with a common mindset yet
- Consider people with different backgrounds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



© Mountain Goat Software, LLC

53

Mike Cohn contact info



mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)



© Mountain Goat Software, LLC

54