

1

Mike Cohn

- Founding member and director of Agile Alliance and Scrum Alliance
- Founder of Mountain Goat Software
- Doing Scrum since 1995
- Started my career as a programmer



2

Agenda

- Why transitioning to agile is hard
- ADAPTING to agile development
- Iterating toward agility
- The role of leadership
- Overcoming resistance



© Copyright Mountain Goat Software®

3

I

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 both at the same time



© Copyright Mountain Goat Software®

4

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

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



© Copyright Mountain Goat Software®

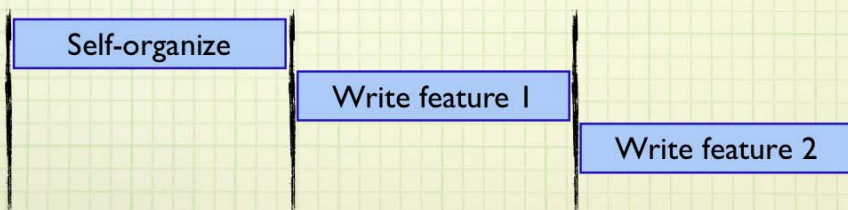
5

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!



© Copyright Mountain Goat Software®

6

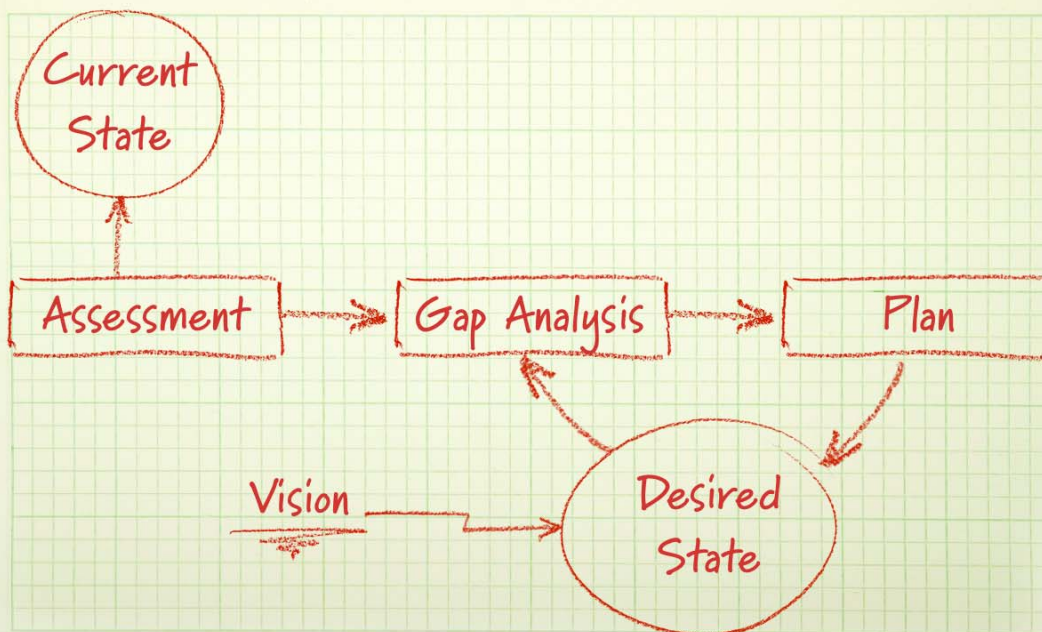
4

Organizations are unpredictable, living systems

- Traditional view of the organization is as a machine
 - We can disassemble it, study its part, put it back together
 - Once we set it in motion, it will continue in motion



A flawed view of organizational change



We need a different mental model

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

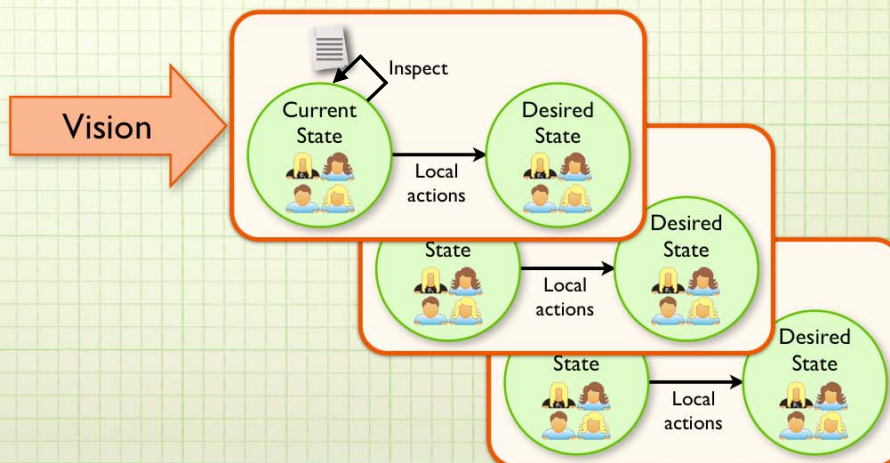
- A dynamic network of many agents
 - acting in parallel
 - acting and reacting to what other agents are doing
- Control is highly dispersed and decentralized
- Overall system behavior is the result of a huge number of decisions made constantly by many agents

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



Local goals and gaps

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



Differing views of success

Newtonian View of Success

Closing the gap with the desired state

CAS View of Success

Achieving a good fit with the environment



- Each paired statement below and on the next slide describes either the traditional or CAS view of how to change an organization
- Put an X in the appropriate column to indicate which describes the traditional view and which the CAS view

	Traditional view	CAS view
Behavior is predictable and controllable		
Behavior is unpredictable and uncontrollable		
Direction is determined through emergence and by many people		
Direction is determined by a few leaders.		
Every effect is also a cause		
Every effect has a cause		



	Traditional view	CAS view
Relationships are directive		
Relationships are empowering		
Responsiveness to the environment is the measure of value		
Efficiency and reliability are measures 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		



Agenda

- Why transitioning to agile is hard
- ADAPTING to agile development
- Iterating toward agility
- The role of leadership
- Overcoming resistance



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



© Copyright Mountain Goat Software®

15

“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.

We also have the issue with at least one other developer as well.”

?

1. Is this a problem of Awareness, Desire or Ability?
 - Why?
2. Thinking about ADAPT, what might you try?



© Copyright Mountain Goat Software®

16

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



On the following slides, identify at least five ways to build desire, create ability, promote the transition, and transfer its implications.



Desire


Ability



Promote

Transfer



 Stop!
My thoughts on desire,
ability, transfer and promote
are on the following pages.
You don't want to see them
before you think about your
own.



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



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

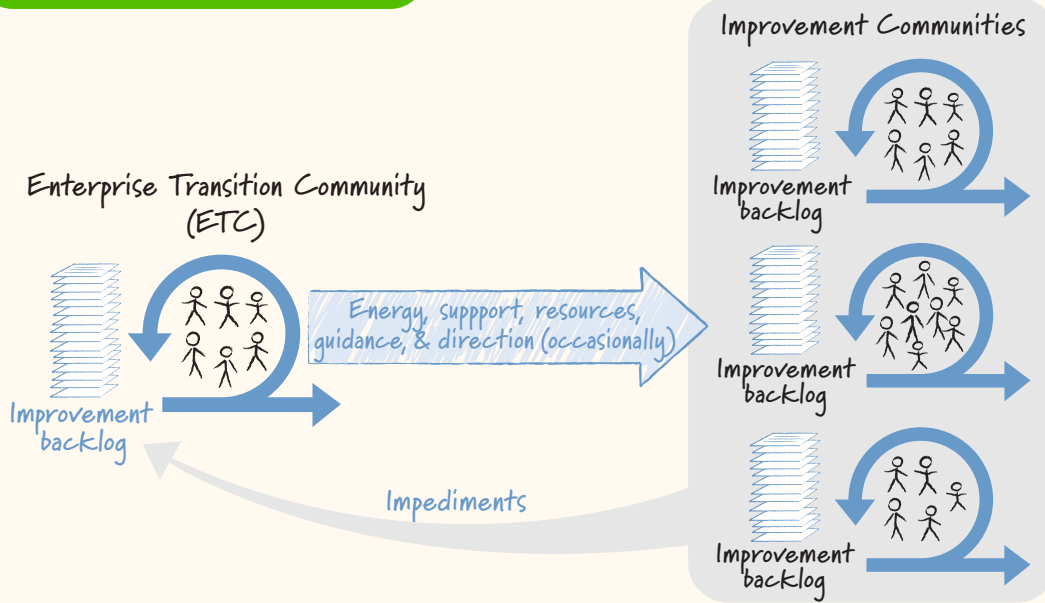


Agenda

- Why transitioning to agile is hard
- ADAPTING to agile development
- Iterating toward agility
- The role of leadership
- Overcoming resistance



Iterating toward agility



Enterprise Transition Community (ETC)

- Creates a culture in which passion and desire to improve 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.

...

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?

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



Improvement backlogs

1. Write some items that you would like an on your organization's initial improvement backlogs.
2. What improvement communities would you like to see form that could help with some of these improvements?
3. What obstacles are in the way of making these improvements?
4. What obstacles are in the way of forming an ETC or improvement community to get started?



Agenda

- Why transitioning to agile is hard
- ADAPTING to agile development
- Iterating toward agility
- The role of leadership
- Overcoming resistance



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



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

- You can influence how a team self-organizes by altering 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 who should?
- Add or remove people from exchanges
 - Change reporting relationships
 - Relocate people
 - Compliance with external groups
- Encourage learning



You are the ScrumMaster or coach...

- The next slides describes some teams with some trouble spots. Think about how you might help them by changing their **Cotrainers**, 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 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. They're 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 glances that other team members exchange when he suggest 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.



The self-organizing path

- Self-organization is not something that happens one time
 - A team is never done doing it
 - The team continually re-organizes in a sense-and-respond manner to its environment
- As you see the team self-organize you can influence, but not control or direct, its path
- We can view this as the evolution of a team



© Copyright Mountain Goat Software®

47

Self-organization proceeds from the premise that effective organization is evolved, not designed. It aims to create an environment in which successful divisions of labor and routines not only emerge but also self-adjust in response to environmental changes. This happens because management sets up an environment and encourages rapid evolution toward higher fitness, not because management has mastered the art of planning and monitoring workflows.

~Philip Anderson

†Anderson, P. “Seven Layers for Guiding the Evolving Enterprise” in *The Biology of Business*.



© Copyright Mountain Goat Software®

48

Variation, selection & retention

- Evolution is the result of three elements:
 - Variation, selection and retention
- Consider a giraffe:
 - **Variation:** A random mutation that leads to a longer neck
 - **Selection:** The long neck helps it reach food others can't; so it is more likely to survive and breed
 - **Retention:** The mutation is passed to its descendants



Three ways to influence team evolution

1. Define performance
2. Manage meaning
3. Evolve vicarious selection systems



Define performance

- The principle of selection tells us that the traits that help us survive will be the ones retained
- Managers and leaders send messages about which traits should survive
- What message is your organization sending about the relative importance of short vs. long-term performance?
 - What messages are sent if the organization:
 - Provides training
 - Supports working at a sustainable pace
 - Allows employees time to explore wild ideas
 - Doesn't exchange meeting a deadline for unmaintainable code



2 Manage meaning

- Individuals in a CAS respond to the messages they receive; e.g.,
 - bees responding to a “danger” message
 - ants responding to a “food found over here” message
- Leaders can push messages into the system
 - e.g., putting the the team in touch with customers
- Or keep messages out
- Meaning often comes from the stories, myths and rituals that are repeated
 - “We will become profitable this quarter.”
 - “Our GM counts the cars in the lot every day at 5 PM”



3 Evolve vicarious selection systems

- Variation—Selection—Retention
 - Selection was determining which variations will be retained
 - Can take a long time
- So we often use vicarious selection systems
 - This is an animal that can smell that a food is poisonous, rather than eating it
- Using only the marketplace as our selection mechanism takes too long
- Organizations also evolve vicarious selection systems
 - Retrospectives, Google's 20% policy, improvement communities



Agenda

- Why transitioning to agile is hard
- ADAPTING to agile development
- Iterating toward agility
- The role of leadership
- Overcoming resistance



How They Resist
Active
Passive

Diehards

Saboteurs

Followers

Skeptics

Like status quo

Dislike agile

Why They Resist



Handling resistance

- Pick one type of resistor. (Or two if you have time.)
- Identify some underlying reasons for the behavior and some things you could try to overcome it.

How They Resist
Active
Passive

Diehard

Saboteur

Follower

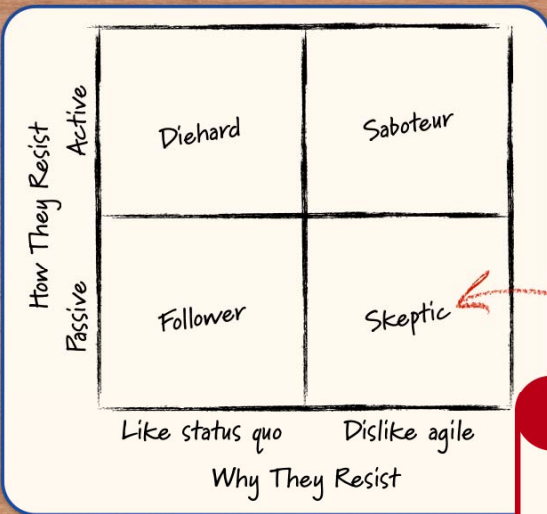
Skeptic

Like status quo

Dislike agile

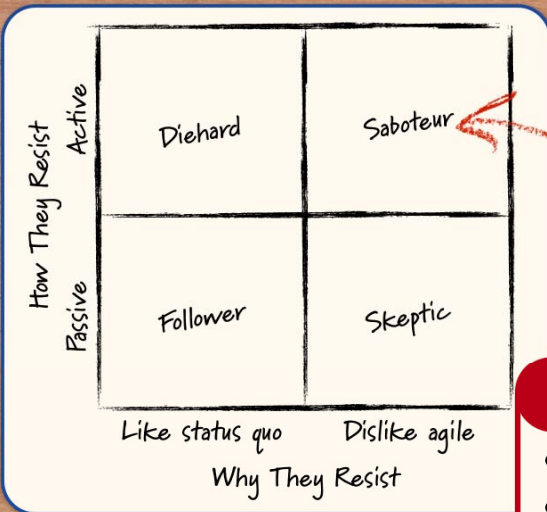
Why They Resist





Skeptics

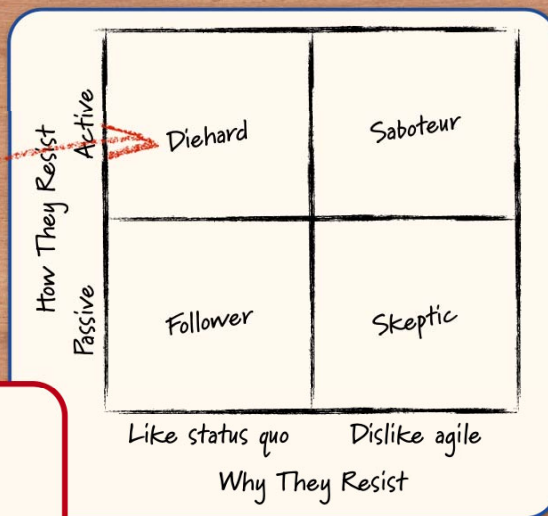
- Let time run its course
- Provide training
- Solicit peer anecdotes
- Appoint a champion skeptic
- Push the issue
- Build awareness



Saboteurs

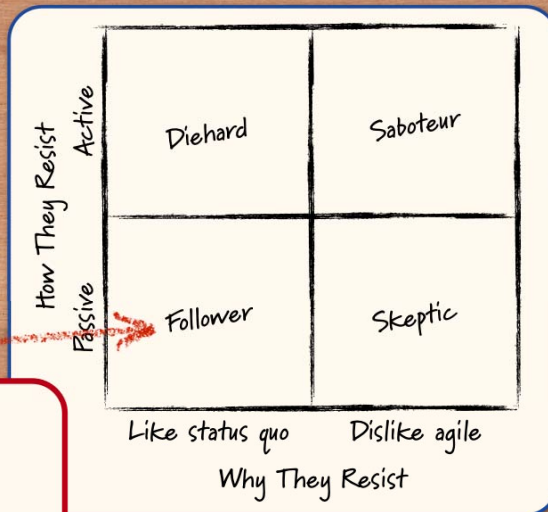
- Success
- Reiterate and reinforce the commitment
- Move them
- Fire them
- Encourage a thriving set of communities





Diehards

- Align incentives
- Create dissatisfaction with the status quo
- Acknowledge and confront fear



Followers

- Change team composition
- Praise the right behavior
- Model the right behavior
- Involve them
- Identify the true barrier (awareness, desire, ability)



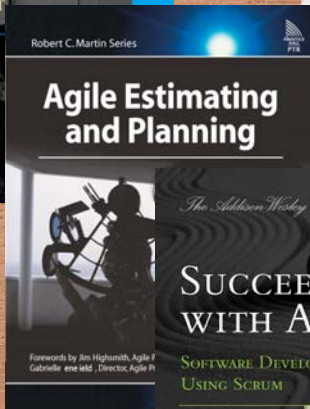
Upcoming public classes

Date	What	Where
July 19–20 July 21–22	Certified ScrumMaster Certified Scrum Product Owner	Orlando
August 23–24 August 25–26	Certified ScrumMaster Succeeding with Agile	Dallas
September 13–14 September 15–16	Certified ScrumMaster Certified Scrum Product Owner	Cupertino
October 11 October 12–13 October 14	User Stories for Agile Requirements Certified ScrumMaster Agile Estimating & Planning	Boulder
November 8–9 November 10–11	Certified ScrumMaster Succeeding with Agile	San Diego

See
mountaingoatsoftware.com
 for details



© Copyright Mountain Goat Software®



Mike Cohn
mike@mountaingoatsoftware.com
www.mountaingoatsoftware.com
 twitter: mikewcohn
 (720) 890-6110



© Copyright Mountain Goat Software®