

Leading a Self-Organizing Team

Mike Cohn
Scrum Gathering - Seattle
May 16, 2011

© Copyright Mountain Goat Software®

1

Agenda

- Self-organization and subtle control
- Containers, Differences and Exchanges
- Influencing how the team evolves
- Situational agile leadership

© Copyright Mountain Goat Software®

2

What is a self-organizing team?

- Self-organizing does not mean
 - the team gets to decide what goal they pursue
 - or even necessarily who is on the team
 - (some self-organizing teams are given this responsibility)
- Self-organizing is about the team determining how they will respond to their environment
 - (and managers/leaders can influence that environment)



© Copyright Mountain Goat Software®

3

Complex adaptive systems

A CAS is characterized by:

- 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



© Copyright Mountain Goat Software®

4

Some examples

- Ant colony or bee hive
- Flock of geese heading south
- Us right now
- A family preparing, eating, and cleaning up after a meal
- A crowd batched up to get into a concert or sporting event
- Cars and drivers on the highway
- A software team



Control is not evil

- Simple rules or incentives are used to guide or direct behavior
 - “Drive this direction and on this side on the highway.”
- For bioteams, these are provided by nature
 - “Produce honey”
- For our teams,
 - Rules and incentives can be added by managers or leaders...or in some cases by team members



Self-organization does not mean that workers instead of managers engineer an organization design. It does not mean letting people do whatever they want to do. It means that management commits to guiding the evolution of behaviors that emerge from the interaction of independent agents instead of specifying in advance what effective behavior is.

~Philip Anderson, *The Biology of Business*



Although project teams are largely on their own, they are not uncontrolled. Management establishes enough checkpoints to prevent instability, ambiguity, and tension from turning into chaos. At the same time, management avoids the kind of rigid control that impairs creativity and spontaneity.

~Takeuchi & Nonaka

"The New New Product Development Game,"
Harvard Business Review, January 1986.



To be sure,
control is still exercised;
but, it is subtle
and much of it is indirect.

~Peter DeGrace & Leslie Stahl
Wicked Problems, Righteous Solutions



What this is not

- We're not talking about
 - Being deceptive or sneaky
 - Manipulating people
- Nothing I'm going to advocate needs to be *secret*
 - But there may be reasons why you don't broadcast your reasons



Agenda

- Self-organization and subtle control
- Containers, Differences and Exchanges
- Influencing how the team evolves
- Situational agile leadership



© Copyright Mountain Goat Software®

11

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*



© Copyright Mountain Goat Software®

12

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 **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 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.



Agenda

- Self-organization and subtle control
- Containers, Differences and Exchanges
- Influencing how the team evolves
- Situational agile leadership



© Copyright Mountain Goat Software®

21

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
- Leaders can influence--but not control or direct--this path



© Copyright Mountain Goat Software®

22

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®



23

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



© Copyright Mountain Goat Software®

24

Influencing team evolution

1. Define performance
2. Manage meaning
3. Evolve vicarious selection systems
4. Add energy
5. Reduce or absorb complexity
6. Create vacuums

- Philip Anderson, "Seven Levers for Guiding the Evolving Enterprise."
- Gita Surie and James K. Hazy, "Generative Leadership: Nurturing Innovation in Complex Systems."
- James Kelly and Scott Nadler, "Leading From Below" in *MIT Sloan Management Review*.



© Copyright Mountain Goat Software®

25

1) 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



© Copyright Mountain Goat Software®

26

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”



Managing meaning

- Think of at least one story that is part of your corporate folklore
- What meaning does that story have about company principles, values, attitudes, or behaviors?



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, compensation



© Copyright Mountain Goat Software®

29

4) Add energy

- Unless energy is pumped into the system, entropy will set in
- Make sure the group has a “clear, elevating goal”[†] or an “igniting purpose”[‡]
- Motivation
 - Project chartering: Vision box, press release, magazine review, elevator statement
- Opportunity
 - To learn, a bigger role, to go onto even better projects, and so on
- Information
 - Customer visits, training, conferences, brown-bags

[†]Larson and LaFasto: *Teamwork*
[‡]Lynda Gratton: *Hot Spots*



© Copyright Mountain Goat Software®

30

May 25, 1995

To: All Microsoft Employees
Subject: Internet Tidal Wave

The Internet is a tidal wave. It changes the rules. It is an incredible opportunity as well as an incredible challenge. I am looking forward to your input on how we can improve our strategy to continue our track record of incredible success.

Bill G.



© Copyright Mountain Goat Software®

31

5) Reduce or absorb complexity

- Reduce complexity
 - Standardize work through
 - routines, standards, policies, and procedures
- Absorb complexity
 - Create relationships among people / departments to provide better access to information



© Copyright Mountain Goat Software®

32

6) Create vacuums

- Point out issues but don't point out answers or expected solutions
- Let others step into this leadership vacuum
- “What keeps me up at night is this, that, and such-and-such.”

“Leading From Below” by James Kelly and Scott Nadler,
MIT Sloan Management Review.



© Copyright Mountain Goat Software®

33

Agenda

- Self-organization and subtle control
- Containers, Differences and Exchanges
- Influencing how the team evolves
- Situational agile leadership



© Copyright Mountain Goat Software®

34

Team readiness

- The coach or ScrumMaster is responsible for knowing the readiness of the team
- Think about each team in terms of
 - Willingness to change
 - Ability
- Different teams need different styles of leadership

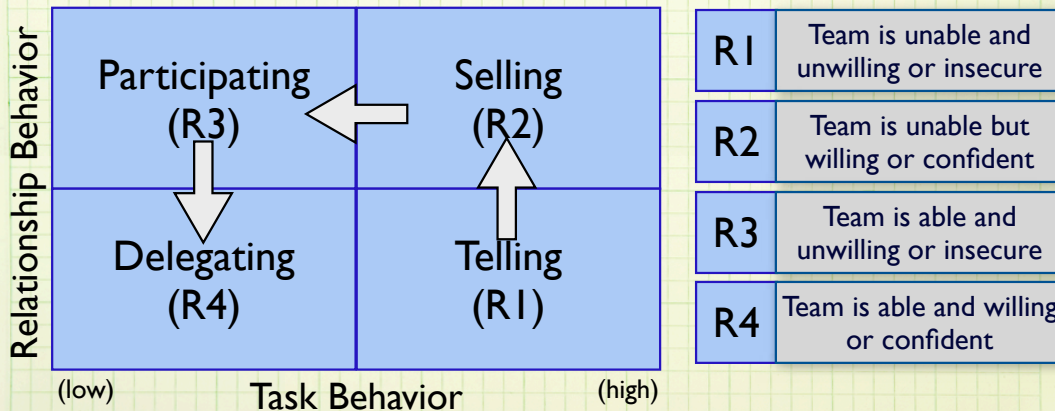


Assessing readiness level

Ability	Willingness	Readiness Level
Unable	Unwilling or insecure	R1
Unable	Willing or confident	R2
Able	Unwilling or insecure	R3
Able	Willing or confident	R4



Different leadership for different teams



Readiness level 1 teams



- Unable & unwilling or insecure
- Need a **telling** leadership style
 - Focus more on telling them what to do than on establishing relationships
- Cannot become immediately agile
 - This isn't command-and-control
- It's getting a low morale team ready to change
- Team needs to develop confidence



Building confidence in R I teams

- This team cannot hit a “home run project”
- Try for small victories first instead
- Provide day-to-day task guidance
- Use short (1–2 week) iterations so they (and you) can see how they’re doing
- Point out the improvements
- Build up foundational agile skills such as unit testing
- Get a nightly build running that emails nightly test results



Readiness level 2 teams



- Unable but willing or confident
- Need a **selling** leadership style
 - Need to exhibit high task direction and highly supportive relationship behavior
- Main goal is increasing team’s skills
- Can become agile
- Good coach / ScrumMaster shifts decision-making style
 - Starts to rely on the team to make its own decisions



Improving skills of R2 teams

- Have the team start making more of their own decisions
 - Be a safety net if the decision is critical
 - Good time to introduce retrospectives
- Stress that it's OK to focus on quality of code over rapid typing
 - Pair programming
 - Test-driven development



Readiness level 3 teams



- Able but unwilling or insecure
- Need a **participating** leadership style
 - Give less task direction
 - Have team make more of their own decisions
- Team will make some mistakes
 - So what?
- Team is probably truly in the agile space



Helping the R3 team

- Team is manically focused on the trees
 - So ScrumMaster / coach keeps an eye on the forest
- Watching the forest
 - Lookahead planning
 - What will the team need in 2 weeks? A month?
 - Make sure sprints continue to build toward the release plan
 - Identify high-value work
 - Put a project prioritization process in place



Readiness level 4 teams



- Able and willing and confident
- Need a **delegating** leadership style
 - Low amount of task direction
 - Low reliance on relationships to manage the team
- Team is undoubtedly agile
 - Highly skilled
 - Self organizing



Unleashing the R4 team

- No longer helping team make decisions
 - Now helping them learn how to defer decisions
- ScrumMaster focuses on maximizing throughput
 - Rather than meeting deadlines
 - Traditional Project Manager thinks of a project like a 10k race
 - ScrumMaster thinks of it as a 1-hour race
 - There's no finish line; clock runs out and you stop



© Copyright Mountain Goat Software®

45

My upcoming classes

Date	What	Where
Jul 11–12 Jul 13–14	Certified ScrumMaster Certified Scrum Product Owner	Santa Clara
Aug 15–16 Aug 17–18	Certified ScrumMaster Certified Scrum Product Owner	La Jolla
Aug 22 Aug 23–24 Aug 25	Effective User Stories Certified ScrumMaster Agile Estimating & Planning	Dallas
Oct 10–11 Oct 12–13	Certified ScrumMaster Succeeding with Agile	Santa Clara

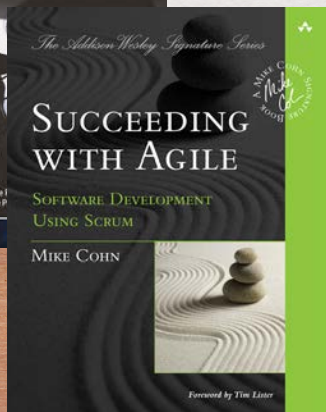
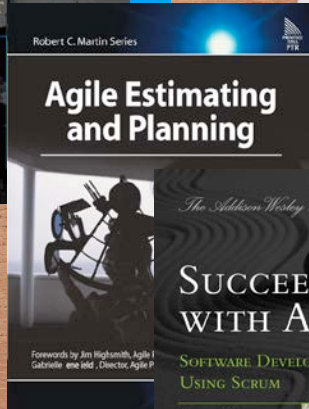
More info at
www.mountaingoatsoftware.com

Classes also in
London, Oslo, &
Copenhagen



© Copyright Mountain Goat Software®

46



Mike Cohn
mike@mountaingoatsoftware.com
www.mountaingoatsoftware.com
twitter: mikewcohn
(888) 61-AGILE



© Copyright Mountain Goat Software®