



Leading a Self-Organizing Team

Mike Cohn

Norwegian Developer's Conference
6 June 2014

1

Agenda

- ☐ Self-organization and subtle control
- ☐ Containers, Differences and Exchanges
- ☐ Influencing how the team evolves



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/leads 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



© Copyright Mountain Goat Software

4

Some examples

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



© Copyright Mountain Goat Software

5

Control is not evil

- Simple rules or incentives are used to guide or direct behavior
 - “Drive this direction and on this side of 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



© Copyright Mountain Goat Software

6

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”



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



© 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:
 - 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



© Copyright Mountain Goat Software

13

Containers

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



© Copyright Mountain Goat Software

14

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



© Copyright Mountain Goat Software

15

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



© Copyright Mountain Goat Software

16

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 programmers, two testers, a database engineer and you. The programmers and testers are not working well together. Programmers 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 the other team members exchange when he suggests very low estimates on some tasks.

6

You are responsible for two teams. Team members on on 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



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



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, *The Biology of Business*



Team evolution

- 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



7 levers for influencing evolution

1. Selecting the external environment
2. Defining performance
3. Managing meaning
4. Choosing people
5. Reconfiguring the network
6. Evolving vicarious selection systems
7. Energizing the system

Philip Anderson, "Seven Levers for Guiding the Evolving Enterprise."



© Copyright Mountain Goat Software

25

1 Select the external environment

- More than just the physical environment
 - What business are we in?
 - (OK, maybe you can't influence this one, but someone can)
 - The company's approach to innovation
 - Fast follower or innovator? Are mistakes OK? When?
 - Types of projects worked on and the rate at which they are introduced to the organization
 - Expectations about multitasking and focus



© Copyright Mountain Goat Software

26

2

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

27

3

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”



© Copyright Mountain Goat Software

28

4 Choose people

- Clearly, who is on the team influences how they self-organize
- Adjust
 - Team Size
 - Location
 - Background
 - Experience
 - Decision-making style
 - Gender
 - Motivation
 - Skepticism
- Some people are like “glue” and pull a team together and keep it there



© Copyright Mountain Goat Software

29

?

- Should a development team be allowed full control over who is on the team?
- Under all circumstances or only some? Which?
- What are the advantages and disadvantages?



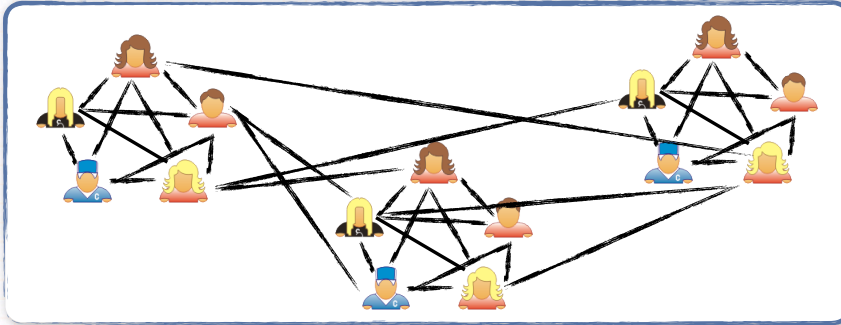
© Copyright Mountain Goat Software

30

5

Reconfigure the network

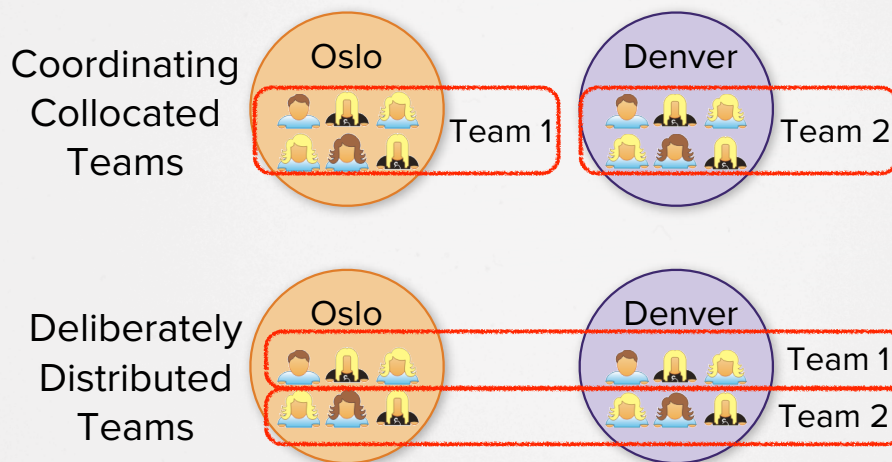
- Communication paths (formal and informal) can be more important than the individuals
- You can introduce or remove flows
 - To other teams, experts in the organization, customers



© Copyright Mountain Goat Software

31

Reconfiguring the network for a global team



© Copyright Mountain Goat Software

32

6

Evolve vicarious selection systems

- Variation—Selection—Retention
 - Selection determines 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 can evolve vicarious selection systems
 - Retrospectives, Google's 20% policy, compensation



© Copyright Mountain Goat Software

33

7

Energize the system

- Unless energy is pumped into a system, entropy sets in
- Make sure the group has a “clear, elevating goal”[†] or an “igniting purpose”[‡]
 - 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

34

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

35

FrontRowAgile.com

Sign In | Create Account **CART**

Courses Instructors FAQ

Get the best seat in the class

Front Row Agile teaches people all over the world part of their on how to s to help er.

LEARN
with agile training videos

PROGRESS
through online quizzes

EARN
certificates, PDUs and SEUs

GET STARTED TODAY! >

Take a look at some courses

AGILE ESTIMATING & PLANNING
with Mike Cohn

coming soon
The Scrum Field Guide
with Mitch Lacey

SCRUM REPAIR GUIDE
with Mike Cohn



© Copyright Mountain Goat Software

36

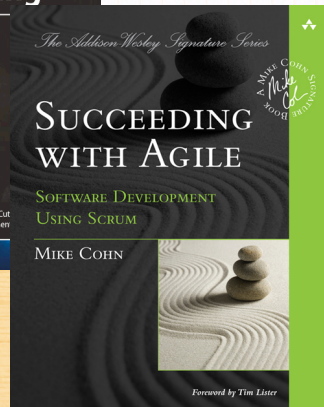
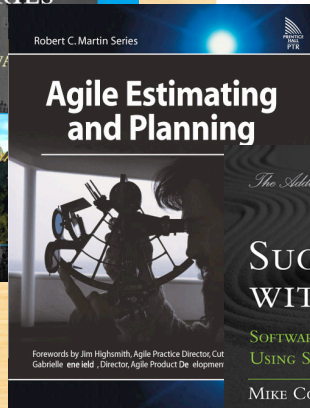
Mike Cohn

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

twitter: mikewcohn

(720) 890-6110



© Copyright Mountain Goat Software