



Scaling Agile with a Distributed Team

Mike Cohn
NDC London
6 December 2013

1

Scaling Issues

- Dependencies
- Iteration planning meeting
- Coordinating teams



2

Proactively manage dependencies

A Use rolling lookahead planning



Iteration 4

Tasks	Est.
Code the...	8
Test the...	16
Integrate with the...	8
Code the...	12
Design the...	8

Iteration 5

Iteration 6

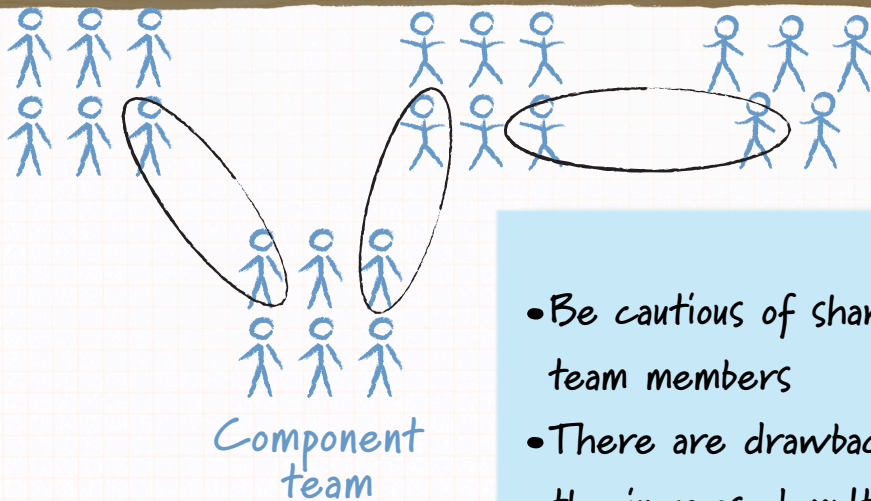


© Copyright Mountain Goat Software

3

B Share team members

Feature team 1 Feature team 2 Feature team 3



- Be cautious of sharing team members
- There are drawbacks to the increased multitasking



© Copyright Mountain Goat Software

4

Two types of interfaces to worry about

Unattended interfaces

At least one team is aware of the interface, but no one is doing anything about it

Unidentified interfaces

An interface that exists but that no one has discovered yet



C Use an integration team

- Takes on the unattended interfaces
 - While on the look out for unidentified ones
- Can be a virtual team with part-time members
 - Common up to perhaps a dozen teams
- Larger projects will have a full-time team
 - Maybe more than one
- Not a dumping ground for poor performers



Scaling Issues

- Dependencies
- Iteration planning meeting
- Coordinating teams



Scale up the iteration planning meeting

- Iteration planning meeting is the hardest to scale
 - Other meetings require less coordination
- Two general approaches
 1. Stagger by a day
 2. The big room



The Big Room



	Nautical Meaning	Our Meaning
	"I require medical assistance."	"We need the product owner."
	"We require assistance."	"We need the architect."
	"We require a tug."	"We require a pizza."
	"We are dragging anchor."	"We are on a break."

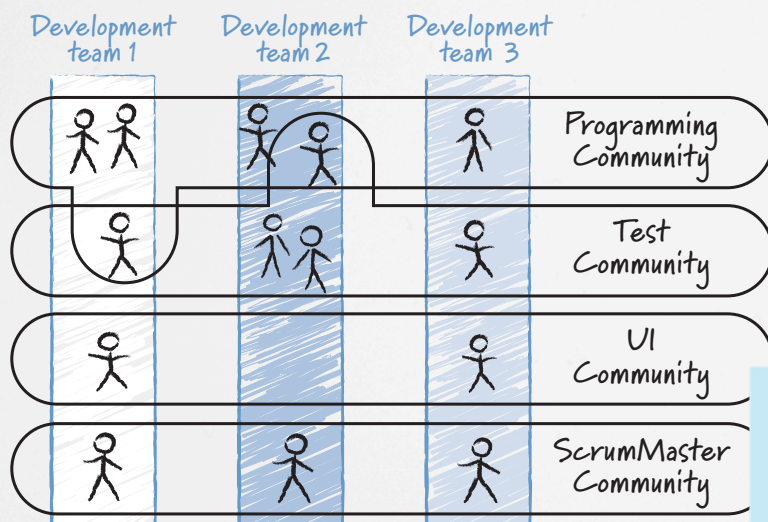


Scaling Issues

- Dependencies
- Iteration planning meeting
- Coordinating teams



Communities of practice



A group of like-minded or like-skilled individuals



Characteristics of communities

- Self-organizing
- Organic
- Can span projects
- Not a full-time job
- There's often a "community coordinator"
 - Typically 5-20 hours/month



Five types of communities

Unrecognized

Invisible to the organization and possibly even to its members.

Bootlegged

Visible but only to a small, select group of insiders.

Legitimized

Officially sanctioned as a valuable entity.

Supported

Provided with resources (time, money, facilities, people).

Institutionalized

Given an official status and responsibilities in the organization.



Creating an environment for communities

1

Design for evolution.

5

Focus on value.

2

Open a dialogue between inside and outside participants

6

Combine familiarity with excitement.

3

Invite different levels of participation.

7

Create a rhythm for the community.

4

Have both public and private events.



© Copyright Mountain Goat Software

15

Scrum of Scrum of Scrums



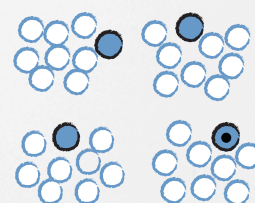
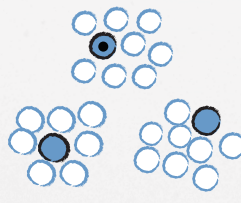
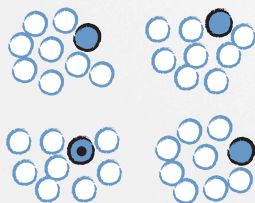
1 / week

Scrum of Scrums



2-3 / week

Daily Scrums



© Copyright Mountain Goat Software

16

Agenda

Three questions (15 minutes at most)

- What has my team done since we last met that might affect other teams?
- What will my team do before we meet again that might affect other teams?
- What problems are my team having that other teams might be able to help with?

Discussion (as long as needed)

- Discuss items kept on an Open Issues Backlog



© Copyright Mountain Goat Software

17

Distributed teams



• Decide how to distribute

• Create coherence

• Change how you communicate

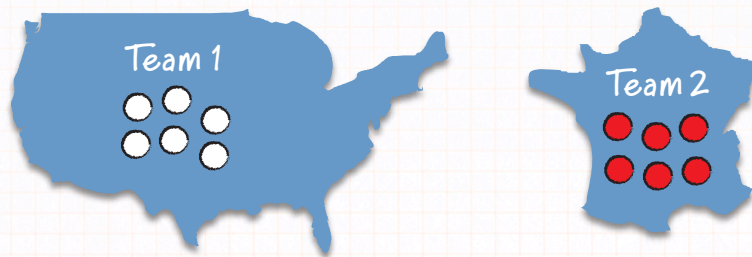


Mountain Goat Software

18

Collaborating colocated teams

- Each team has all needed skills
- Teams in different locations work independently but collaborate to coordinate their work



Deliberately distributed teams

- Each location has all needed skills
 - We could form collaborating colocated teams
 - But we choose not to
- Individuals in different cities work together as one team



Distributed teams



- Decide how to distribute
- Create coherence
- Change how you communicate



Mountain Goat Software

21

Creating coherence

- *Coherent* is from the Latin *cohaerent*
 - “*sticking together*”
 - We want a team that will stick together
- So we’ll
 - Acknowledge big cultural differences
 - Acknowledge small cultural differences
 - Strengthen functional and team subcultures
 - Build trust by emphasizing early progress



© Copyright Mountain Goat Software

22

Create coherence

① Acknowledge cultural differences

- Big cultural differences
 - Attitudes toward power, individualism, achievement, uncertainty, and long-term vs. short-term
 - Geert Hofstede found significant differences among IBM employees in these areas
- Smaller cultural differences
 - Holidays
 - Working hours



More ways to create coherence

② Strengthen functional and team subcultures

- Establish a shared vision
- Establish working agreements

③ Build trust by emphasizing early progress

- Early emphasis on relationship building encourages subgroups to form around surface-level attributes[†]
- Defer relationship building until team members have learned more significant things about each other

[†]Gratton, Voigt, and Erickson. "Bridging Faultlines in Diverse Teams."



Distributed teams



- Decide how to distribute
- Create coherence
- Change how you communicate



Mountain Goat Software

25

Get together in person

- Seeding visits
 - Ideally, whole team meets in person at start
 - Stay together an iteration or more when possible
- Contact visits
 - Whole team, Quarterly, face-to-face
- Traveling Ambassadors
 - Individuals who travel more frequently among locations to ensure good working relationships



© Copyright Mountain Goat Software

26

Change how you communicate

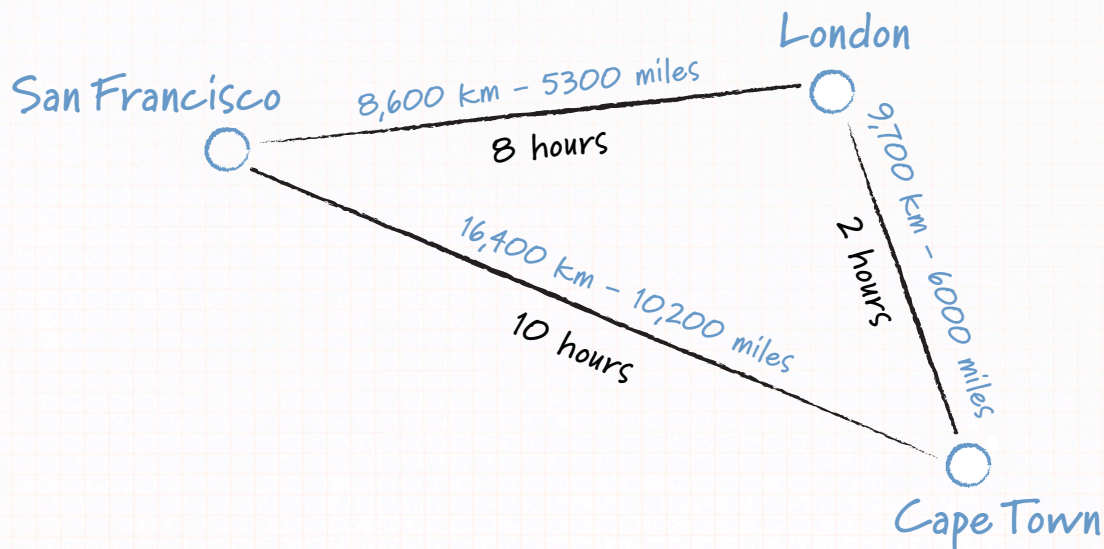
- Add back some documentation
 - Cannot rely as much on talking
- Add detail to the product backlog
- Encourage lateral communication



© Copyright Mountain Goat Software

27

It's not the distance, it's the timezones



© Copyright Mountain Goat Software

28

Useful advice for all meetings

- Include time for small talk
- Share the pain
- Make sure everyone knows who is talking



© Copyright Mountain Goat Software

29

Iteration Planning—Approach #1 The Long Phone Call

- Everyone on the phone at once

Pros

- Can lead to good discussion if people remain engaged
- Planning is finished in a day
- Is consistent with approach used when collocated

Cons

- People mentally disengage during long calls
- Only feasible with significant overlap of workdays



© Copyright Mountain Goat Software

30

Iteration Planning—Approach #2 Two Calls

- First call: understand what the product owner wants built
- Local subteams figure out what they can commit to
- Second call the next day: Subteams share commitments

Pros

- Can be a more efficient use of time
- Can be used whenever work hours can be made to overlap even a little

Cons

- Usefulness varies based on how widely distributed the team is
- Not all knowledge is shared with everyone, leading to misunderstandings
- Takes two days



Daily Standup—Approach #1 Single Call

- Everyone on the phone at once

Pros

- Similar to what is done with collocated teams so there's nothing new to learn
- Discussions involve the whole team
- Everyone hears all issues, leading to greater commitment

Cons

- Can be extremely inconvenient for some
- Not sustainable if people are forced to work outside of normal work hours



Daily Standup—Approach #2

Writing the meeting

- Everyone emails a written report or updates a wiki with status information
- Variation: A local group meets and others email updates

Pros

- Sustainable over the long term
- Helps overcome language problems

Cons

- No guarantee updates are read
- Issues are not discussed and may lay dormant
- Doesn't take advantage of daily interaction to improve relationships and knowledge sharing
- Reduced feeling of accountability to teammates



Daily Standup—Approach #3

Regional Meetings

- Have separate regional phone calls, e.g., western hemisphere and eastern hemisphere
- Follow these with a written summary shared between teams
- Or have one person from each region also participate in the other calls

Pros

- Pain of off-hours calls is greatly reduced
- Allows local subteams to share information most relevant to them

Cons

- Information relayed from one meeting to another may be incorrect or incomplete
- Can lead to us/them feelings
- Not everyone is involved in all discussions
- Information may not be shared in timely manner



Mike Cohn

mike@mountaingoatsoftware.com

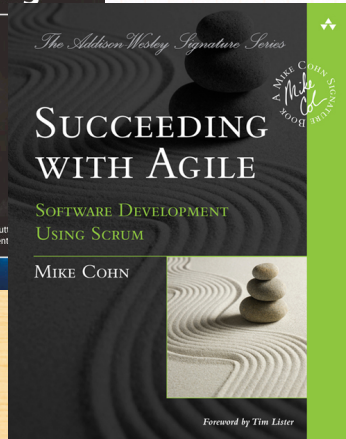
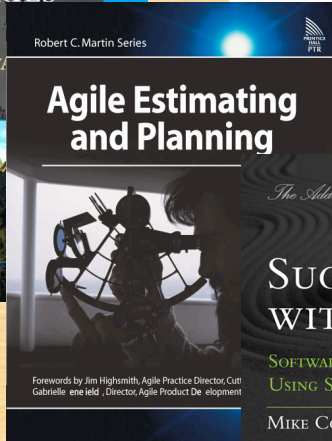
www.mountaingoatsoftware.com

twitter: mikewcohn

(720) 890-6110



MOUNTAIN GOAT
SOFTWARE



© Copyright Mountain Goat Software