



**The Remote Agile Guide—  
A Hands-on Manual from the Trenches  
by Stefan Wolpers**

Version: 1.7 | 2020-08-05

# The Remote Agile Guide



## Table of Content

<b>PREFACE</b>	<b>5</b>
<b>MASTERING THE BASICS</b>	<b>6</b>
<b>PRACTICES &amp; TOOLS FOR SCRUM MASTERS &amp; AGILE COACHES</b>	<b>6</b>
INTRODUCTION	6
GENERAL TIPS	6
GEAR	8
APPLICATIONS	8
TIPS & TRICKS	9
THINGS TO WATCH OUT FOR	10
CONCLUSIONS	11
<b>VIRTUAL LIBERATING STRUCTURES</b>	<b>12</b>
LIBERATING STRUCTURES	12
LIBERATING STRUCTURES FOR SCRUM	13
DESIGN ELEMENTS OF VIRTUAL LIBERATING STRUCTURES	13
HOW TO PRACTICE INDIVIDUAL VIRTUAL LIBERATING STRUCTURES	13
CONCLUSION	16
ATTRIBUTION	16
<b>REMOTE AGILE ANTI-PATTERNS — PITFALLS SUCCESSFUL DISTRIBUTED TEAMS AVOID</b>	<b>17</b>
INTRODUCTION	17
CULTURE	17
COLLABORATION ANTI-PATTERNS	18
COMMUNICATION ANTI-PATTERNS	19
FACILITATION ANTI-PATTERNS	20
CONCLUSIONS	21
<b>REMOTE AGILE EVENT PLANNING: A CHEAT SHEET</b>	<b>23</b>
INTRODUCTION	23
THE REMOTE AGILE EVENT PLANNING PHASES	23
THE PLANNING OF THE MAIN EVENT	25
CONCLUSION	27
<b>TECHNOLOGY FOR VIRTUAL EVENTS</b>	<b>28</b>
<b>MASTERING ZOOM</b>	<b>28</b>
INTRODUCTION	28
ZOOM BASICS	28
BREAKOUT ROOMS	30
CONCLUSIONS	32
<b>USING UNHANGOUT TO HOST A VIRTUAL BARCAMP OR OPEN SPACE</b>	<b>34</b>
INTRODUCTION	34
OPEN SPACE TECHNOLOGY	34
UNHANGOUT'S PROMISE	35
LESSONS LEARNED FROM USING UNHANGOUT TO HOST A VIRTUAL BARCAMP AT THE 24TH HANDS-ON AGILE MEETUP	36
THE SCRIPT OF THE 24TH HANDS-ON AGILE MEETUP	38
A VIRTUAL BARCAMP W/ UNHANGOUT: DISCUSSION, FEEDBACK, AND LESSONS LEARNED	38

# The Remote Agile Guide



CONCLUSIONS	39
<b>REMOTE SCRUM EVENTS</b>	<b>40</b>
<b>REMOTE RETROSPECTIVES</b>	<b>40</b>
INTRODUCTION	40
REMOTE RETROSPECTIVES WITH VIRTUAL LIBERATING STRUCTURES AND ZOOM BREAKOUT ROOMS	40
THE DESIGN ELEMENTS OF VIRTUAL LIBERATING STRUCTURES	40
THE 5 STAGES OF A RETROSPECTIVE	41
SUPPORTING LIBERATING STRUCTURES	44
APPLICATIONS TO RUN A REMOTE RETROSPECTIVE	45
GOOD PRACTICES FOR A REMOTE RETROSPECTIVE	45
ANTIPATTERNS OF REMOTE AGILE RETROSPECTIVES	46
CONCLUSIONS	46
<b>REMOTE SPRINT PLANNING</b>	<b>47</b>
INTRODUCTION	47
THE PURPOSE OF THE SPRINT PLANNING	47
VIRTUAL LIBERATING STRUCTURES FOR A REMOTE SPRINT PLANNING	48
SUPPORTING LIBERATING STRUCTURES	49
REMOTE SPRINT PLANNING ANTIPATTERNS	50
CONCLUSION	51
<b>REMOTE SPRINT REVIEW</b>	<b>52</b>
INTRODUCTION	52
THE PURPOSE OF THE SPRINT REVIEW	52
WHAT DOES THE SCRUM GUIDE SAY ABOUT THE SPRINT REVIEW?	53
VIRTUAL LIBERATING STRUCTURES FOR A REMOTE SPRINT PLANNING	53
VIRTUAL SPRINT REVIEW PART 1: WHAT HAS SCRUM TEAM ACCOMPLISHED DURING THE SPRINT?	54
VIRTUAL SPRINT REVIEW PART 2: IS OUR PRODUCT BACKLOG PREPARED FOR THE NEXT SPRINT?	54
VIRTUAL SPRINT REVIEW PART 3: CLOSING	55
SUPPORTING LIBERATING STRUCTURES	55
REMOTE SPRINT REVIEW ANTIPATTERNS	56
CONCLUSION	57
<b>REMOTE DAILY SCRUM</b>	<b>58</b>
INTRODUCTION	58
THE PURPOSE OF THE DAILY SCRUM	58
VIRTUAL LIBERATING STRUCTURES FOR A REMOTE DAILY SCRUM	59
SUPPORTING LIBERATING STRUCTURES	61
REMOTE DAILY SCRUM ANTIPATTERNS	61
CONCLUSION	62
<b>REMOTE AGILE TRANSITIONS</b>	<b>63</b>
<b>REMOTE AGILE TRANSITIONS — THE TOP-TEN CHALLENGES</b>	<b>63</b>
INTRODUCTION	63
WHY REMOTE AGILE ACCELERATES CHANGE	63
CONCLUSIONS	65
<b>AGILE LAWS &amp; DISTRIBUTED TEAMS: FROM CONWAY TO GOODHART TO PARKINSON</b>	<b>66</b>
INTRODUCTION	66
AGILE LAWS: CONWAY, BROOKS, HACKMAN, GOODHART, LARMAN, AND PARKINSON	66

# The Remote Agile Guide



# The Remote Agile Guide



## Preface

Welcome to my Remote Agile Guide, detailing practices, tools, and anti-patterns of how to work as Scrum Master, Product Owner, or agile coach with a distributed team.

This ebook is work in progress, and I add more chapters to it every week.

Please note that I will not be able to automatically provide you with new versions of this ebook if you unsubscribe from the *Food for Agile Thought* newsletter. In doing so, you also delete your email address from the list of readers of this ebook.

Thank you for your understanding!

Best,  
Stefan

Berlin, in May, 2020.

# The Remote Agile Guide



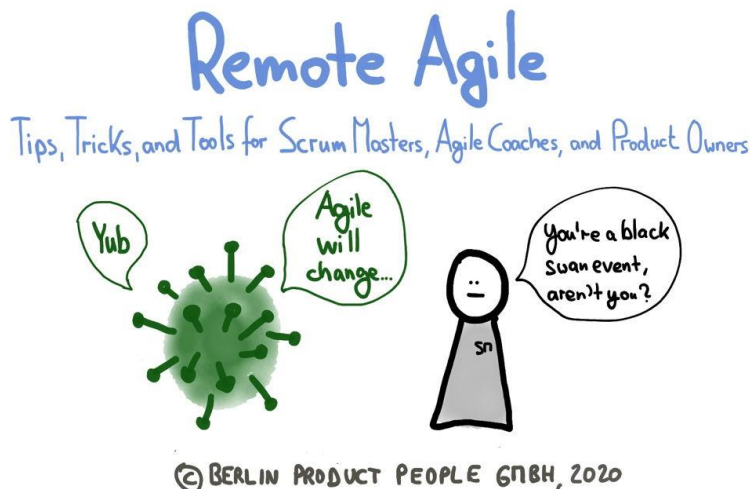
## Mastering the Basics

### Practices & Tools for Scrum Masters & Agile Coaches

#### Introduction

Agile has always been about colocation, about direct communication, physical boards, and all the other haptic and analog moments to create value in the digital realm. As a Professional Scrum Trainer, I strive to provide training classes without using PowerPoint or digital devices. Moreover, admittedly, I love index cards. However, times are changing. So, let's dive into the particularities of practicing remote Agile with distributed teams.

We start with basic techniques and tools and follow up with how to apply Liberating Structures in the remote realm. If you like to get a hands-on introduction, consider joining my [live virtual classes on Remote Agile](#).



#### General Tips

Remote Agile cannot be reduced to running a Retrospective in Zoom, sharing your screen. However, understanding the techniques on how to do so in the most effective way for your team is the primary education any facilitator needs to acquire.

Regarding this facilitation know-how, there is good news: Remote Agile is neither complex nor chaotic; hence there are good practices to start from without the necessity to reinvent the wheel. Here we go:

# The Remote Agile Guide



1. **Keep sessions short:** No one can effectively follow more than 3 to 4 hours of online sessions per day. Preferably, reduce the session length to 2-hour blocks per day and include frequent breaks and activities that entice the engagement of the participants. Have regular breaks to allow for leg stretching, too.
2. **Avoid hybrid meetings:** At the moment, the first remote team member joins, everyone joins the events via “dial-in,” too, no matter whether others are co-located or not. The reason for this is that a successful meeting requires an equal distribution of airtime and bandwidth among participants. The moment you allow co-located people direct access, there will be an imbalance at the expense of the remote participants. This imbalance may lead to exclusion and diminishing the contribution of the remote folks, as they cannot join the informal conversation among the co-located people, they probably won’t be aware of body language, etc. All of this is creating a less safe situation for the remote people that typically affects their contribution.
3. **Be the first one:** As the facilitator of an event, be up & running 10 to 15 minutes early to ensure, that the setup is working — remember VPN and firewall issues — and those early attendees are admitted to the event and feel appreciated.
4. **Working agreement:** Have a working agreement that addresses the basic rules of remote agile events such as mute yourself if you are not talking, or raise your hand if you want to talk, or that the video signal is always on. If the team is working remotely all the time, adding these to the team charter is a good practice. Otherwise, agree on the rules of the game at the beginning of the meeting.
5. **Prep the students:** You cannot expect that students are familiar with all the tools you will be using during a virtual event or class. If they feel overwhelmed technically, they may cognitively drop out of the session. Alternatively, you may need to spend much more time on technical issues than planned. Both cases result in an inferior experience. Consider therefore offering a preparatory meeting to clarify technical issues.
6. **Privacy:** Discuss privacy or confidentiality issues in advance. For example, recording a video of a session might be a good idea if everyone is aware of it and not objecting. Otherwise, the idea of including everyone, giving them a voice, will fail from the start.
7. **Be always on:** Be strict with the “video is always on” policy. If you allow avatars instead, people will hide behind them, probably becoming distracted by another task, they decide to tackle. Humans are bad at multi-tasking.
8. **Workbooks:** Provide a briefing/workbook for your remote agile event in advance, detailing how to get access to the event technically, how to participate, its ground-rules, what preparation is expected from the participants, or providing the handouts the participants need for the planned session. Typically, there won’t be time to clarify questions around these issues during the event.

# The Remote Agile Guide



9. **Time zones:** Be mindful of time zones. Spreading a session over more than 4 to 6 time zones is challenging.
10. **Noise:** Make sure that attendees understand that it is vital for the success of the remote agile event that they participate from a quiet place, not the bean sack in the chill corner of the open office. No noise cancellation software is able to eliminate that level of background noise.
11. **Chat:** Use a backchannel for group-wide and private communication, for example, the build-in chat of video-applications or Slack or MS Teams.

For the facilitator, remote agile events mean more preparation before a session and more documentation after the session. You will spend more time at your desk.

## Gear

We can keep the gear section short; there are numerous articles out there detailing every conceivable aspect of that issue. So, I like to stick with a few fundamental principles:

- **Audio 1:** Get the best microphone your budget can afford: It sounds contradictory, but audio quality is significantly more important than the video of cameras for the success of a remote agile event. (Also, consider installing a [noise cancellation software such as Krisp](#).)
- **Audio 2:** Get a headset instead of using your iPhone thingies.
- **Screen space:** Get a large screen. Your iPad or laptop screen will make working with large virtual whiteboards and the numerous other windows of the other necessary applications quite exhausting.
- **Camera:** You do not need a stand-alone 4K camera. Built-in HD cameras of most devices are good enough.

## Applications

Similar to the gear section, we can limit the list of applications to a few prominent ones that will get you going with remote agile events:

1. **Video:** Of course, we start with [Zoom](#). The killer feature is, besides its video-codec that uses only little bandwidth, breakout rooms. If you want to divide a large group into small groups and have them work on an issue, this is the way to go forward. For example, I use the breakout rooms for Liberating Structures' 1-2-4-All. (Here is a [short video explaining how Zoom breakout rooms](#) work.) Also, Zoom's freemium policy is quite generous.



# The Remote Agile Guide



2. [QigoChat](#) is a collaboration suite that also supports breakout rooms.
3. **Office 365** and **Google Apps** have also proven to be handy collaboration tools. (In corporate environments, there may be firewall issues with Google Apps.)
4. You will also need a **digital whiteboard** for group collaboration. There are simple free applications like [Whiteboard Fox](#). On the other side of the spectrum, you find professional applications like Miro (formerly known as RealTimeBoard) or [Mural](#). [Josh Seiden summarizes their utility](#): “We set up a Mural board for each team, and have one that we use with the whole group. are able to set up sections on each board for specific exercises, and we pre-load those sections with templates, examples, and other materials that people will need during class.”)
5. **Kanban boards**: Trello, a free Kanban board, adds to the category of collaboration tools. Of course, there is also its big brother, Jira. (Which is also free for a limited number of team members.)
6. **Retrospectives**: There are numerous applications out there that support Retrospectives. Retrium is a well-known subscription service, while [FunRetro.io](#)'s Freemium approach allows using the free retrospective for voting purposes, for example, as a Lean Coffee tool.
7. Of course, there is also a specialized commercial service for that purpose, the [Lean Coffee Table](#).
8. **Chat**: As mentioned earlier, you need to provide a backchannel for group communication. Slack, as well as MS Teams, offer the leading experience for this category. In both cases, the free versions should satisfy all communication requirements your team might have.
9. **Surveys**: You will need to run surveys to gather data, for example, for Retrospectives. Google Forms is a useful tool for that purpose. If your firewall restricts access to Google Apps, try [Typeform](#) instead; up to three surveys are free.
10. **Polls**: Finally, if you need to run **polls** at your remote agile events, [Poll Everywhere](#) is a suitable candidate. Again, its Freemium approach allows using the application to a certain level for free.

If you are looking for a more comprehensive list, check out [Lisette Sutherland's list of tools](#).

## Tips & Tricks

There is more to practicing remote Agile that meets the eye. Try the following:

# The Remote Agile Guide



1. **Script:** Prepare a time-line of the remote agile event with a script so that you have canned questions, instructions, and other texts at hand without breaking into a sweat. You will need your energy for other tasks.
2. **Starting:** Always start with some icebreaker questions; for example, by facilitating an Impromptu Networking. Never start with a teaching block.
3. **Teaching blocks:** If presentations or teaching blocks are necessary, keep them between 5-10 minutes at the maximum. Follow up on every teaching block with an engaging group activity.
4. **Time:** Be strict with time-boxes.
5. **ELMO:** Hence make use of the “enough-let’s-move-on” principle (Elmo) if discussions get too engaged. Employ a parking lot to take a discussion outside the remote agile event.
6. **Patience:** Make everyone aware of the latency in communication — it takes a few seconds to unmute yourself and make a contribution.
7. **Real whiteboards?** No one can decipher what is written on a physical whiteboard and transmitted by a room camera. Use a digital whiteboard instead and share your screen.
8. **Zoom in, please:** Make everyone sharing a screen zoom in; otherwise, their content might be practically unreadable.
9. **Toolbox:** Get a box with all required equipment — from dongles to cables to power supplies — if you move around within a building.

## Things to Watch out for

I like to close this first article with some remote Agile anti-patterns:

- **Lack of preparation:** No matter how attentive and organized you are, your participants probably won’t read your instructions to prepare for the event. How hard can it be to write some stickies, right? (I once tried to run an introductory meeting with the applicants for a new change team in a traditionally structured organization. I chose a Lean Coffee for that purpose and detailed everything necessary in advance. Nevertheless, the meeting failed miserably after 25 minutes, as everyone pretended to understand the format but did not have a clue and also did not ask. Later, participants accused me of a complete lack of preparation — of course, via a backchannel. I did not see that coming from future “change agents.”)
- **Silent people:** Actively address participants who are not contributing at the same level as other participants.

# The Remote Agile Guide



- **Sarcasm:** Choosing words in a remote realm is much more difficult in comparison to face-to-face communication. For example, sarcasm can be difficult to detect remotely. Watch for clean language to avoid unnecessary misunderstandings.
- **Surveillance:** Don't go rogue; the prime directive still rules in a remote agile setup. [Trust in people and do not spy on them](#) — no matter how tempting it might be.

## Conclusions

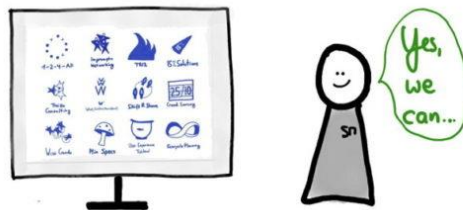
Losing face-to-face communication is without question an obstacle to becoming an agile team. However, it does not mean that remote Agile is a futile undertaking from the start. It only requires a better-prepared approach to compensate for the loss of nearness. This article addresses the basics of that approach. Next week, we delve into how to practice Liberating Structures with a remote agile team.




## Virtual Liberating Structures

Last week, we addressed basic practices and tools of remote agile with distributed teams. Based on that article, I also ran a live virtual class, the recording of which will be made available soon on the [Age-of-Product's Youtube channel](#). This follow-up post now delves into virtual Liberating Structures, answering the question of how we can make use of the powerful toolbox of inclusive and collaborative practices in a remote setting.

### Remote Agile - Live Virtual Class How to Use Virtual Liberating Structures



 Live Virtual Class on March 31, 2020, 4-6:30 pm CET  
© BERLIN PRODUCT PEOPLE GMBH, 2020

## Liberating Structures

Created by Keith McCandless and Henri Lipmanowicz, [Liberating Structures](#) cover a set of easy to learn, yet powerful ways to collaborate as a team—even as a (very) large team by Scrum standards—, overcoming traditional communications approaches like presentations, managed discussions, or another disorganized brainstorming at which the loudest participants tend to prevail.

Generally, Liberating Structures are well suited to improve the level of engagement among participants of Scrum events, thus stimulating the kind of outcomes that are necessary to create learning organizations. Liberating Structures also provide an excellent toolbox to handle Product Backlog refinements or improving the Definition of Done of an engineering organization.

Moreover, Liberating Structures are a great tool when peers come together to jointly figure out how to improve as an individual as well as a professional.

# The Remote Agile Guide



## Liberating Structures for Scrum

More than a year ago, my Berlin Hands-on Agile meetup started exploring the possibilities to use Liberating Structures specifically in the context of Scrum. As a result, we have explored 15-plus Liberating Structures so far and started tailoring Liberating Structures strings to specific situations, such as Scrum events, or the Megabrain.io scenario, our struggling startup. ([Read all articles of the Liberating Structures for Scrum series.](#))

In this article, I revisit most of those microstructures and provide initial thoughts on how to use them as virtual Liberating Structures.

## Design Elements of Virtual Liberating Structures

Virtual Liberating Structures share a set of common design principles:

- **Breakout rooms** are used to divide the whole group of participants into smaller workgroups, starting with pairing up two participants. (I am using Zoom for that purpose.)
- **Muting/unmuting** is used — beyond the purpose of reducing noise — to mark different states of participants. For example, in the Conversation Café exercise during rounds 1, 2, and 4, everyone is muted except the individual that is sharing his or her thoughts.
- **Video on/off** is used to distinguish between roles, for example, between the inner circle and the outer circle of the User Experience fishbowl. Here, the outer circle members turn their video off as well as mute themselves.
- A **shared workspace** is needed to aggregate findings, for example, as the result of a 1-2-4-All session. This can be a simple Google slide or a [FunRetro.io](#) board.
- **Workbooks** are useful to provide participants with instructions when working in breakout rooms; for example, a detailed description of how an individual Liberating Structures works.
- A **chat channel** is used to facilitate communication within the whole group.

## How to Practice Individual virtual Liberating Structures

Let's explore the microstructures for their potential to become virtual Liberating Structures. The ordering is based on the LS Menu of the Liberating Structures website:

1. [1-2-4-All](#): To cover 1-2-4-All, we need breakout rooms and a place to aggregate the findings. We start with everyone in the whole group for a minute in silence; then, we split the whole group into pairs using Zoom's breakroom feature for 2 minutes. After that round, we merge two pairs into a group of four for five minutes—this has to be done manually

# The Remote Agile Guide



by the host—and the group aggregates its findings, for example, on a Google sheet prepared for each group in advance. We can introduce each group’s findings to the whole group by screen sharing in a Shift & Share.

2. [Impromptu Networking](#): This is a simple application of breakout rooms; just make sure that after each round, the pairs are created a new. Provide the invitation and the three questions in the workbook in advance.
3. [Appreciative Interviews \(AI\)](#): AI is another application of breakout rooms, workbooks, and a shared workspace. Introduce the steps in the whole group, break into pairs for the individual interviews, merge two pairs into one breakout room, then gather insights within the group workspace to share later with the whole group. AI works well with 1-2-4-All and Shift & Share. Make sure that you provide a description of AI in the workbook, and also consider time-keeping via the breakout room broadcasting function for the admin. (Participants tend to get lost otherwise.)
4. [TRIZ](#): Again, TRIZ is a combination of basic elements of virtual Liberating Structures: breakout rooms, embedded 1-2-4-All, joined workspaces, Shift & Share when several groups are working on the problem. Consider time-keeping via the breakout room broadcasting function, as participants are likely to be highly engaged and may lose track of time.
5. [15% Solutions](#): We use a similar procedure as with TRIZ. Consider aggregating all suggestions in the whole group’s shared workspace for clustering and ranking by voting. (I like to use a FunRetro.io board for that purpose: it is simple and does not need much explaining.)
6. [Troika Consulting](#): We start by creating breakout rooms for groups of three. Consultants and the consultee have the initial conversation; then, the consultee turns around on his or her chair for the consulting phase. Alternatively, both consultants stop broadcasting their video, so the consultee is just listening to what they have to say. Again, as the facilitator consider time-keeping on behalf of the groups.
7. [What, So What, Now What?](#): Also, W3 is a sequence of individual work and group work based on breakout rooms, aggregating findings in shared workspaces to be shared with the whole group in the end.
8. [Shift & Share](#): That is a simple one: each workgroup presents its findings to the whole group by screen sharing. Alternatively, if the shared workspace has been created in advance, for example, Google Slides with a slide per workgroup, the moderator can share his or her screen while someone from the team is explaining the findings to the whole group. This reduces the stress of switching screen sharing on and off among several groups.

# The Remote Agile Guide



9. [25/10 Crowd Sourcing](#): This one LS that I haven't yet tried in the virtual realm. I am still thinking about how to make this happen. (I would probably miss the dancing while swapping the suggestions...)
10. [Min Specs](#): Like What, So What, Now What?, Min Specs is a sequence of individual work and group work based on breakout rooms, aggregating findings in shared workspaces to be shared with the whole group in the end.
11. [Conversation Café](#): Create groups with the breakout room function, and identify a host for time-keeping. During rounds 1, 2, and 4, where one participant is talking while the others are listening, use mute for the listeners. Once the timebox has expired, the previously talking participant “hands over” the microphone by calling out the next one in line and then muting him- or herself. As the facilitator, also consider providing a matrix — rounds by speakers with checkboxes — to the hosts to ensure that everyone has a fair share of airtime.)
12. [User Experience Fishbowl](#): Working in the whole group, use mute and video off to distinguish between the inner circle and the outer circle of the User Experience fishbowl. Here, the outer circle members turn their video off as well as mute themselves. Gather additional questions through the chat channel from the out circle members. A facilitator should pass on these new questions in due time. (While discussing the topic at hand, the inner circle members should try not to read these new chat messages at the same time.) Use W3 to debrief in the whole group.
13. [Heard, Seen, Respected \(HSR\)](#): A classic application of Zoom's breakout room function. Again, as a facilitator, consider becoming the external time-keeper.)
14. [Lean Coffee](#): Lean Coffee is an excellent example of a workaround for virtual Liberating Structures. Gather all the input in the usual way, for example, engaging in 1-2-4-All, and gather those on a FunRetro.io board while voting is turned off. (Use several columns if the whole group is large to speed up the gathering process.) Then ask the whole group to cluster similar topics, then turn on the voting and order the remaining entries by votes. For here, you continue with a whole group discussion, or you engage smaller groups with breakout rooms.
15. [Ecocycle Planning](#): Principally, we apply the techniques as before, from breakout rooms to shared workspaces. Speaking of which, given the large number of “stickies” that you usually create during Ecocycle planning, you may want to consider a specialized online board application such as Miro or Mural. (Please note that both tools are not self-explanatory and require a prep session with participants to avoid frustrating them.)

# The Remote Agile Guide



## Conclusion

While there will always be a bit of the magic missing that in-person Liberating Structures sessions have, I do believe that there is plenty of opportunities to create a sound experience with virtual Liberating Structures.

Of course, they are different — which can be attributed mostly both to the lack of nearness among the participants as well as the limited technical support of virtual Liberating Structures. However, given the alternatives, virtual Liberating Structures again prove to be far superior to other competing virtual practices.

## Attribution

[Liberating Structures](#) are developed by Henri Lipmanowicz and Keith McCandless and are licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#). What experience have you made with applying virtual Liberating Structures? Please share it with us in the comments.



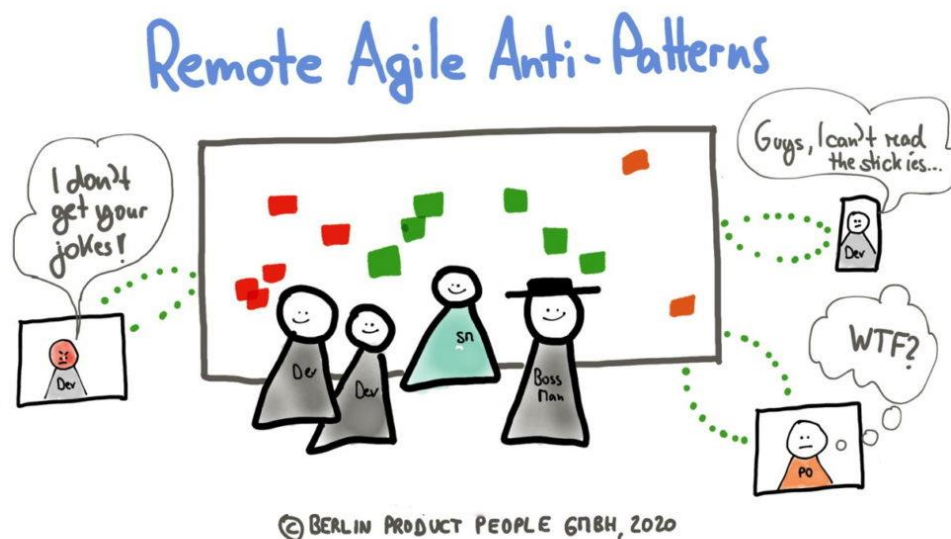
# The Remote Agile Guide



## Remote Agile Anti-Patterns — Pitfalls Successful Distributed Teams Avoid

### Introduction

We started this series on remote agile with looking into practices and tools, followed by delving into virtual Liberating Structures, and how to master Zoom. This fourth article now addresses basic remote agile anti-patterns — the pitfalls any distributed team wants to avoid to become successful.



### Culture

**What about our culture?** The organization is not doubling down on transferring or preserving its culture in a remote agile environment. (My favorite definition of company culture goes as follows: Culture is what happens when you are not looking. Now apply this model to a situation where there are no watercoolers and coffee machines that can act as waterholes for the tribe, and where having lunch is not an option. Going remote requires more effort to cultivate your organization's cultures, particularly at the leadership level.)

**Remote Agile is just standard work-life plus Zoom:** Pretending that working remotely is the same as usual except for the video cameras. (This approach ignores all the challenges that distributed team face, for example, not investing enough in getting to know each other better to build trust. We are Social animals and need to meet In person sooner or later to build lasting trust among teammates, thus creating psychological safety. Moreover, there are difficulties in reading the virtual room in general, which means that taking decisions to the team or calling out intro-

# The Remote Agile Guide



verts manifest themselves differently in a remote working setup. Trust is the beginning of all; without it, transparency, inspection, and adaptation would be able to work their magic, and we end up as distributed feature factories.)

## Collaboration Anti-Patterns

Ignoring the etiquette and good practices of remote work:

- **Neither fish nor meat:** Hybrid events create two classes of teammates — remote and co-located — where the co-located folks are calling the shots. (Beware of the distance bias—when out of sight means out of mind—thus avoiding the creation of a privileged subclass of teammates: “Distance biases have become all too common in today’s globalized world. They emerge in meetings when folks in the room fail to gather input from their remote colleagues, who may be dialing in on a conference line.” ([Source](#).) To avoid this scenario, make sure that once a single participant joins remotely, all other participants “dial in,” too, to level the playing field.)
- **Too many meetings:** Every communication feels like a (formal) meeting. (This is often resulting from a trust issue at the management level or the felt need to come to unanimous decisions across the team. Instead, put trust in people, uphold the prime directive, and be surprised what capable, self-organizing people can achieve once you get out of their way.)
- **Surveillance:** Trust won’t be built by surveilling and micro-managing team members. Therefore, don’t go rogue; the prime directive rules more than ever in a remote agile set-up. [Trust in people and do not spy on them](#) — no matter how tempting it might be. (Read more about the damaging effect of a [downward spiraling trust dynamic from Esther Derby](#).)
- **Mindless rituals:** Leadership belief and or facilitation practices turn once useful routines into mindless rituals. (For example, think of Groundhog Day-style retrospectives over and over again. Answering the same three questions every single time is the easiest path to kill any form of creativity and collaboration. While this is hard to avoid in face-to-face environments, it requires much more dedication and energy in a remote agile setting.)
- **Death by PowerPoint:** Meetings still revolve around an individual broadcasting a slide deck. (While you might get away with this approach for some time in face-2-face environments, it will not fly with distributed teams. Sessions need to be inclusive, interactive, and engaging to entice collaboration, think Liberating Structures, and Training from the Back of the Room.)
- **Delaying important work:** Saving the ‘hard work’ for later to be done in person. (This delay tactic is merely increasing lead time and cycle time by creating new queues. The right time for handling important work is right here, right now.)

# The Remote Agile Guide



- **A misconception of time:** Applying the understanding of time from face-to-face environments to remote collaboration and interactions. Typical symptoms of this fallacy are:
  - Remote sessions are too long and exceed the recommended 2-4 hours.
  - Remote sessions do not provide enough breaks, for example, 10 minutes per hour.
  - The timeboxes are too short for remote collaboration and create stress rather than a focus.
  - The facilitators do not enforce timeboxes, and discussions are meandering aimlessly.
  - Management insists on a 9-to-5 schedule although asynchronous communication does not require this mental model from the factory era. Instead, agree on blocks of time that are overlapping for everyone when virtual events can take place.

## Communication Anti-Patterns

- **Unstructured communication:** “Didn’t you get the memo?” (There is no clear practice on how to communicate which kind of information to whom. Are we talking about email, Slack, the team wiki, a comment in Github, or the biweekly remote brow bag session? This lack of structure and agreement leads to stress—how can I avoid missing important news now that there is no longer a watercooler; do I have to monitor all Slack channels in real-time—and probably a feeling of being excluded. Maybe, this effect is just a missing update to the working agreement or team charter. But what if it is done deliberately? (Honi soit qui mal y pense.) Coping with an remote agile environment always requires to overcommunicate and be completely transparent.)
- **Instant replies:** The organization expects that all communication needs to be synchronous. (There is undoubtedly a need to be available for synchronous communication over the day; think of remote Daily Scrums or virtual watercooler meeting. However, expecting that all communication is immediate causes too much disruption for focused, deep work. The practice of ‘do not disturb hours’ also applies to distributed teams.)
- **Complexification:** The wrong tools or too many tools are used in the belief that technology will fix all collaboration issues:
  - Ask yourself: What is the marginal utility of an additional tool? (The law of diminishing returns applies to your toolbox, too.)
  - Therefore, keep it simple. (Soft factors such as trust at the team level drive successful collaboration; it is not a faster VPN connection.)
  - Are you using the right toolset? (If your team members do not have access to the right tools, making remote agile collaboration successful is challenging, if not impossible. (If your organization still insists on using Skype for Business as the primary collaboration tool, they might be in for a surprise. Working in a distributed

# The Remote Agile Guide



team is demanding enough; trying to do so with inferior tools is adding an unnecessary layer of complexity.)

- **Tech is your problem:** The organization does not provide support with technology or connectivity in the home-office. (Typical manifestations are the lack of a budget for audio or video hardware or accessories such as green screens or proper lighting at the home-office. Or, the necessary software is not licensed, and the organization does not pay for the Internet connection. All of these symptoms fall into the category of penny-wise and pound-foolish.)
- **No training:** The organization is not actively training team members in the technology, software, and practices that the distributed team relies on. (Don't assume that people know how to use the technology and software involved. Instead, offer regular training classes to signal that not knowing how remote agile collaboration works can be resolved. Additionally, at the facilitator level, check-in with novel participants of important remote events beforehand to make sure that their setup is working to reduce their stress levels during the meeting—stressed people are rarely top collaborators.)
- **A lack of Netiquette:** The team ignores the first principles of remote communication: no interrupting, mute yourself when not talking, use a clean, humane, and kind language in messages. (This ought to be a no-brainer, given its principles have been around for decades. Nevertheless, a weak facilitator can easily wreck a team's potential if those remote agile anti-patterns go unanswered. By the way, Matt Mullenweg has an interesting opinion on muting: “One heterodox recommendation I have for audio and video calls when you're working in a distributed fashion is not to mute, if you can help it. When you're speaking to a muted room, it's eerie and unnatural — you feel alone even if you can see other people's faces. You lose all of those spontaneous reactions that keep a conversation flowing. If you ask someone a question, or they want to jump in, they have to wait to unmute. I also don't love the “unmute to raise your hand” behavior, as it lends itself to meetings where people are just waiting their turn to speak instead of truly listening. So what should you do? Use the latest and greatest hardware and software to have the best of both worlds, a fantastic auditory experience for you and your interlocutors and little to no background noise. [...] Save mute for coughs and sips of drinks.” ([Source](#).)

## Facilitation Anti-Patterns

- **5 minutes before 12:** The host joins a remote event only a few minutes before the start. (My suggestion: Be the first one in the meeting about 10-15 minutes early. You may have to handle the waiting room, plus other participants may have technical issues. Also, prepare a time-line of the remote agile event with a script so that you have canned questions, instructions, and other texts at hand without breaking into a sweat. You will need your energy for other tasks. Facilitating remote agile events requires more preparatory work, more documentation, and more debriefing. As a facilitator, you will spend more time at

# The Remote Agile Guide



your desk creating documents.)

- **Lone maverick:** Facilitating events with more than 30 people without a co-facilitator. (Depending on the task, its context, and the experience of the participants, there will be a maximum number of attendees to a remote agile event that a single facilitator can accommodate. Beyond that threshold, hire a co-facilitator for the event.)
- **Mr./Mrs. Speaker:** Becoming too domineering as a facilitator. (While it is true that a host of a remote agile event needs to direct the event more intentionally by comparison to the face-to-face version, he or she shall not slip into a “Speaker” role, having the final say in each matter.)
- **Everyone knows Liberating Structures:** No, they do not. Therefore, do not assume that participants in remote events are familiar with Liberating Structures. (While some are so simple that they can be picked up along the way, others require preparation and probably even a training session in advance. For example, think of Critical Uncertainties or Ecocycle Planning.)
- **Everyone did the homework:** Anticipating that everyone is well prepared for the event is a fallacy. (Principally, this is not different from real-world experience. No matter how attentive and organized you are, your participants probably won’t read your instructions on how to prepare for the event. However, in the remote realm, we face a compound effect as we use additional technology to host the event as well as potentially unfamiliar practices such as Liberating Structures to entice collaboration during the event. Be prepared to deal with these situations.)
- **Using flipcharts:** Using real whiteboards or flipcharts in a remote event—captured by a video camera—and expecting everyone to understand what has been written or sketched. (This is a useless exercise as remote events in corporate settings normally do not take place in a studio with appropriate lighting and 4K cameras. Instead, consider using digital whiteboards or drawing apps like Paper on an iPad to ensure the inclusion of everyone.)
- **Not having a plan B:** How do you continue working, if Zoom, Teams, or Webex fail? (The host of a remote agile event should plan for the inevitable: Internet access is down, the firewall is acting up, the video conferencing system is no longer working. There should be a contingency plan, providing alternative tools to the team. Moreover, this plan B should be tested beforehand. It is a bit like restoring a backup file: if you never test this as an exercise, do not expect it to work in the case of an emergency.)

## Conclusions

Working as a distributed agile team is not merely the continuation of usual practices with a video conferencing tool but practically a paradigm shift, emphasizing self-organization, trust, transparency, and the inclusion of everyone. While this may look obvious on paper, realizing the required

# The Remote Agile Guide



changes at a team or organizational level is a massive undertaking that many might oppose. Hence you encounter numerous remote agile anti-patterns in practically all organizations that did not start as a distributed organization from the beginning.

# The Remote Agile Guide



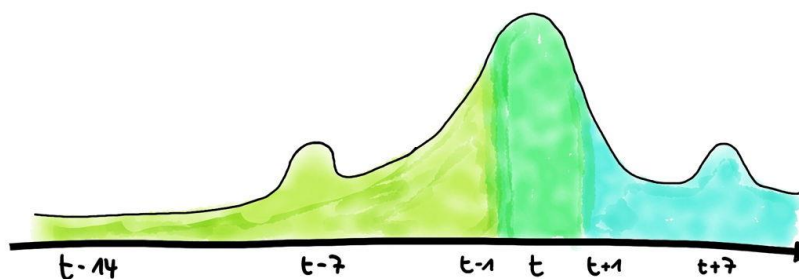
## Remote Agile Event Planning: A Cheat Sheet

### Introduction

Imagine, you are supposed to facilitate a remote user-story mapping with 26 people from all over our organization. Or, think about a remote meta-level Retrospective of a recent release of your organization's cash cow that went sideways. You may ask yourself: How am I planning for this given that participants likely have different levels of experience, and everyone needs to be included fully in the discussion?

My solution to this challenge is going the extra mile regarding the remote agile event planning. The concept I describe below is tested, proven, and modeled after my virtual Professional Scrum training classes.

## Remote Agile Event Planning A CHEAT SHEET



© BERLIN PRODUCT PEOPLE GMBH, 2020

### The Remote Agile Event Planning Phases

In this post, I want to draw your attention to the three phases of your remote agile event planning: The pre- and post-event phase, and the event itself:

#### *T-14 Days: The Pre-Event Planning Phase Kick-off*

The pre-event planning phase for one of my Professional Scrum classes starts two weeks in advance of the training, with a comprehensive introduction email to all participants, and points to the essential steps they need to contribute to making the class a worthwhile experience. I believe

# The Remote Agile Guide



in overcommunicating everything while being transparent regarding practices, tools, and deliverables.

In this email, I address the following topics:

- I provide an invitation to the upcoming (remote) technical prep session to ensure everyone has his or her device set-up properly.
- I also list the technical requirements that every attendee needs to address to ensure the best possible learning experience: What devices are suitable for participation, the necessity to use a microphone and probably headphones, have a back-up Internet connection, and choose a silent space.
- I remind students to update their browsers to the latest versions, install the Zoom app, and ensure access to Google Apps or the chosen equivalent like Office365. (We cannot deal with firewall, VPN- or general networking issues once the big event starts.)
- I describe the schedule of the upcoming event: starting and end times as well as planned breaks.
- I provide the link to the main Zoom session, in my case requiring a registration.
- Should I require additional information from the participants, I explain my need for this and include a link to a prepared survey.
- As we will use Liberating Structures frequently during the remote agile event, I point to an article series covering Liberating Structures for Scrum.
- 

Attached to this email, I also provide several useful PDFs, for example, the Scrum Guide or the [Scrum Guide Reordered](#), as well as a homework assignment for the upcoming technical prep session. Of course, I am available for all upcoming questions in advance of the tech prep session.

## *T-7 Days: The Technical Prep Session*

Approximately a week before the main remote agile event, I run the technical prep session of about 60 minutes, typically in the late afternoon, so the participants are likely to attend. A few hours before the prep session, I mail a reminder to all invitees.

During the prep session, I keep track of who is attending. We start the session by running an [Impromptu Networking](#) among those who joined. This way, we can test whether Zoom and Google Apps are working and build initial rapport among the participants. After the introduction, we explore Mural, as we will use this online whiteboard during the remote agile event to visualize our work. (To do so, we accomplish a few small tasks such as adding your name and avatar to the “who is participating” section of the Mural, or we have a scavenger hunt on the board.)



# The Remote Agile Guide



I also introduce the Google Drive we will use for storing any form of work results. Moreover, we visit all the prepared documents for the main agile event so everyone can familiarize themselves with them. Introducing the documents at this point reduces the chance of surprises during the main event, reducing the level of uncertainty and lessening the cognitive load on the participants. We also try to fix technical issues we run into or at least take note of what problems need to be fixed during the next week before the main event. However, the tech prep session is not supposed to be merely a support event.

Immediately after the tech prep session, I reach out to those invitees of the main remote agile event that missed it, and I offer an alternative appointment for a second tech prep session.

## *T-1 Day: The Reminder*

A day before the remote agile event, I resend the invitation to all participants, including a listing of essential links for the workshop, for example, to the main Zoom session, the Google Drive including all the files we will be using during the main event, or the Murals we will complete. I also include the best ways of how to reach me by email, Slack, or phone.

Do not anticipate that the attendees pay the same level of attention to preparing themselves for the remote agile event as you do. Repeated communication of the essentials is hence useful and necessary.

## *T+1 Day: The Event Summary*

A day after the remote agile event, I provide all the files created during the session(s) to the participants. These include the chat protocol files, sketches, or PDF exports of the Mural boards. I also download all Google App documents as individual files. In a corporate environment, I would point to the documentation where a summary of the event and its findings is available for further study.

## *T+7 Day: The Follow-up*

Depending on the nature of the remote agile event, it may be useful to meet a week after its end to follow-up on the latest learnings and share experiences. (In the context of a Professional Scrum class, we meet to learn from those who already took the certification, thus helping their classmates prepare for their certification attempts.)

## **The Planning of the Main Event**

Regarding the main remote agile event planning, I prepare myself by creating a script that details the aspects of the event itself: What happens when in which sequence for what purpose? What is the intended outcome of an exercise or discussion, how will we collaborate to get there, and where will we store the results? How long will each step approximately take, and when will we likely have a break?

# The Remote Agile Guide



I create those scripts in Excel. The following screenshot is from the first hour of the [Distributed Agile Masterclass](#), see below:

Remote Agile Class: Day 1				
Date:	2020-03-26			
Vers.	1.0		127	
#	Purp Purpose	[min]	Handouts/URLs	Copy for Main Chat in Zoom
<b>1 Introduction</b>				
1	Purpose of the live virtual class, backchannel, Google docs	4		
2	Impromptu Networking:			
3	a) In silence: answering the three questions [2 min]	3		In silence, answer the three questions: 1. What concern do you bring to this class? 2. What can you give to your peers to make this class a success for everyone? 3. What do you hope to take away in return?
4	b) Round 1: Breakout room of 2 [4 min]	5		Please swap roles, 2 minutes left.
5	c) Round 2: Break out room of 2: record the major issue of the partner and describe your partner in one sentence. [4 min]	5	<a href="https://docs.google.com/spreadsheets/d/1nQlX...">https://docs.google.com/spreadsheets/d/1nQlX...</a>	When talking to your partners, note their greatest – professional – concern, and describe who they are in a single sentence.
<b>2 Teaching Block #1</b>				
6	Essential rules for facilitators of virtual events			
7	a) Introduction	5	Slide #2	
8	b) ELMO	2	Slide #3	If you believe that we should move on, please post ELMO to the main chat.
9	c) Parking lot	2	<a href="https://funretro.io/publicboard/eCl...">https://funretro.io/publicboard/eCl...</a>	
<b>3 Lean Coffee with FunRetro.io, Part 1</b>				
10	Continue working with the insight from Impromptu Networking:			
11	a) Transfer of concerns to FunRetro.io	4	<a href="https://funretro.io/publicboard/...0v8d...">https://funretro.io/publicboard/...0v8d...</a>	Post the greatest – professional – concern of your partner from round 2 of the Impromptu Networking to FunRetro.io. (Choose any of the three columns.)
12	b) Clustering of similar concerns	5		As a group, cluster similar topics by merging entries via drag & drop.  Merge all three columns into one at the end.
13	c) Transfer of descriptive introductions to the Google Doc	2	<a href="https://docs.google.com/document...">https://docs.google.com/document...</a>	Add your description of your partner to the Google Doc.
<b>4 Teaching Block #2</b>				

The script comprises all the information that the moderator requires during the different practices. For example, it contains the exact instructions for all exercises, including the URLs to necessary documents, for easy use in the chat room of the event. This way, the moderator can focus on interacting with the attendees.

The purpose of this script is not to create a comprehensive plan to follow to its letters. It is much more about getting an understanding as a moderator about the paths the remote agile event may take, depending on what will happen, and how to achieve the goal of the event nevertheless. As the saying goes: “[Plans are worthless, but planning is everything.](#)” Creating such a script is a considerable yet worthwhile investment. Starting from scratch with designing the workshop itself, it may take several working days to complete.

# The Remote Agile Guide



## Conclusion

Over-communicate everything, be transparent, be available for everyone, and repeat the same messages multiple times on different occasions. This approach requires additional work regarding documentation, communication, and preparation. However, this extra work provides the basis for a moderator's calmness, which in my experience affects the attendees' experience so positively.

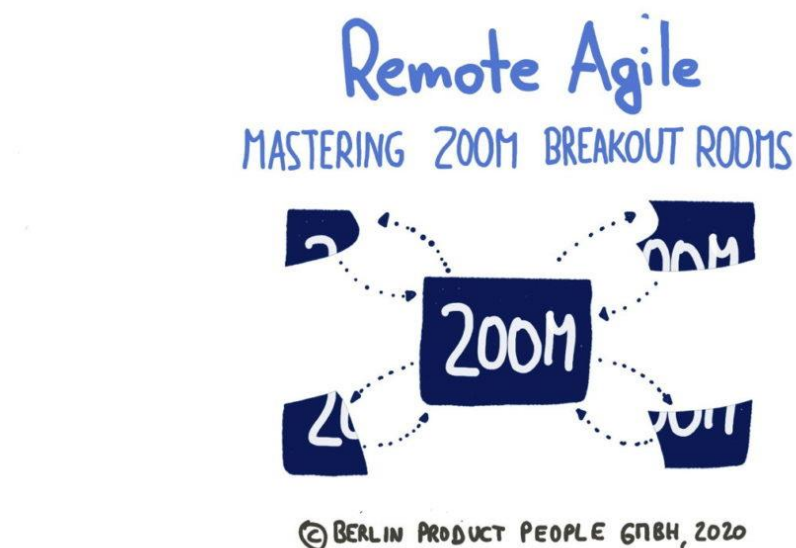


## Technology for Virtual Events

### Mastering Zoom

#### Introduction

In this third post of the Remote Agile series, we address the tool at the heart of working with a distributed team as a Scrum Master, Agile Coach, or Product Owner: Zoom. While Zoom is an excellent video conference application—particularly for larger groups of twelve or more attendees—by all standards, its killer feature is breakout rooms. Turning a more or less passive audience into engaged collaborators where everyone is included and has a voice makes the difference between a successful facilitator and someone who fails to adapt to a new situation. Let's have a look at what mastering Zoom is all about.



Read the previous articles: [Remote Agile \(Part 2\): Virtual Liberating Structures](#) and [Remote Agile \(Part 1\): Practices & Tools for Scrum Masters, Agile Coaches, and Product Owners](#).

#### Zoom Basics

Zoom has an excellent [support section](#) with a library of video tutorials that explain everything in detail. The primary application is reasonably simple to use if you consider a few ground rules:

# The Remote Agile Guide



- Give attendees the possibility to join by phone. (While this is by no means perfect, it probably is better than missing out altogether. If participants are expected to dial from outside our country, please provide them with the respective dial-in information, such as local phone numbers to call.)
- If you run public meetings, enforcing a registration might be useful, as is the use of the password. (For internal meetings, it may be safe to opt to include the password in the sign-up link.)
- Enabling the chat is always useful; it is a vital communication channel for facilitators. (Allow private chat messages, too, but be careful to notice when you are answering a private message instead of posting something for everyone. Please note that ‘private messages’ are exported in the chat.txt file.)
- Make it the standard to turn off video for everyone upon entry. Also, mute everyone.
- If you allow participants to join the meeting before you do, it probably causes more trouble than it is worth. As a facilitator, instead, make it your habit of being there up & running 15 minutes early and welcome every new attendee personally. (It is also a good way to build rapport with everyone.)
- Maybe, you want to consider using the waiting room as it means that you have to admit every participant to the session manually. (Please keep in mind that participants more often than not tend to be late to the session. So, while you already started the session, some others wait for you to admit them. On the other side, the waiting room is suitable means to keep unwanted participants out of your meeting.)
- While you could start recording every meeting automatically, it may look a bit suspicious to your teammates. (Hence when I have to record a session, I first explain my need and ask if there are any objections.)
- Allow screen sharing for everyone but limit the sharing override—a new screen can be shared despite someone else is sharing his or her screen—to the host. (First of all, as a facilitator, you must be able to prevent any of the participants from not stopping sharing a screen. Secondly, you do not want just anyone to hijack the session by [zoombombing](#) [NYT] it. (In this regard, being able to put unruly participants on hold is helpful, too.)

Of course, for customizing aficionados, Zoom offers plenty of opportunities to tweak basically everything one way or another. Again, check out [Zoom’s help center](#).

# The Remote Agile Guide



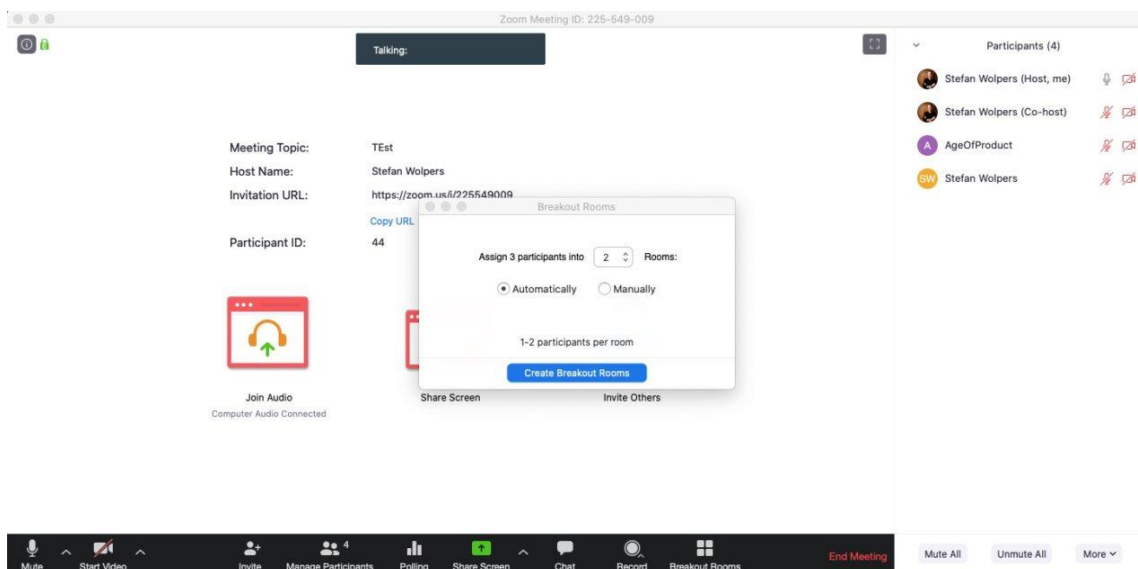
## Breakout Rooms

In the previous paragraph, we scratched the surface of Zoom’s customization capabilities by aggregating a few ground rules to make a facilitator’s life easier. Now, let us have a closer look at the breakout room feature and why it is essential for mastering Zoom.

### Breakout Rooms: What Works

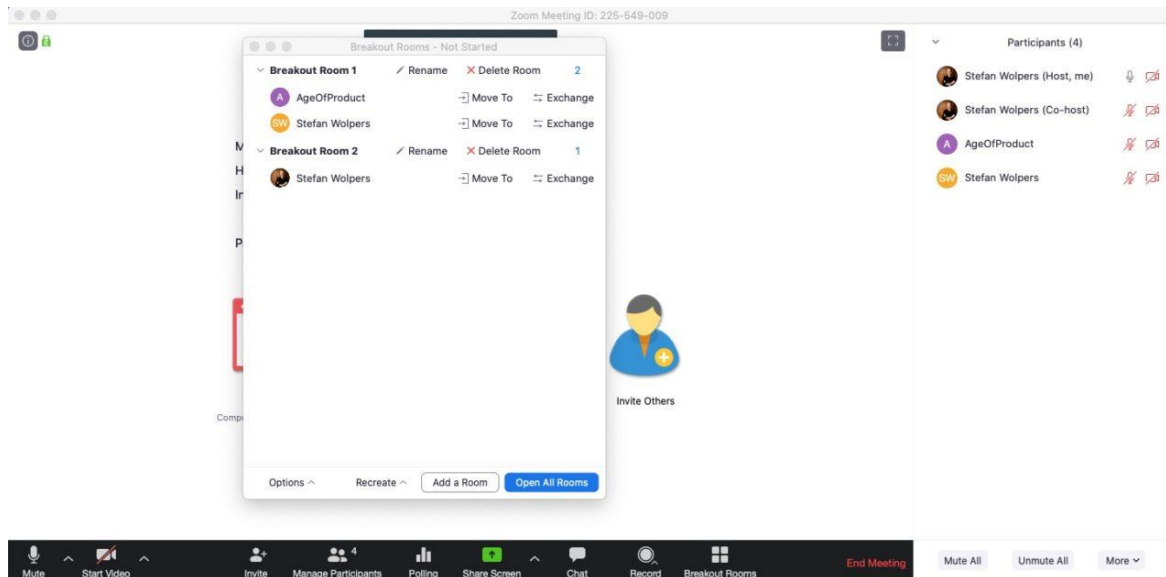
I doubt that Zoom designed the breakout room feature with Liberating Structures or ‘Training from the Back of Room’ principles in mind, which would explain a lot of the missing functionality. However, even at the current state, breakout rooms are essential to mastering Zoom as a facilitator working with distributed agile teams:

- Zoom creates breakout rooms automatically by distributing all participants equally across the number of chosen breakout rooms, or the host creates them manually.



- You can reassign members from one room to another room by drag-and-drop or choosing from the pick-list. This comes handy, for example, if you run an [Impromptu Networking](#) with many participants and some people end up with no partner because those failed to enter the respective breakout room. In this case, reassign participants by hand to another half-populated room to form new pairs.

# The Remote Agile Guide



- You can exchange participants between breakout rooms.
- You can delete suggested breakout rooms before creating them. (However, this won't reassign participants in these rooms to other rooms automatically.)
- If you restart a breakout session, you can decide to preserve the previous room composition or start over with new groups. In the case of Impromptu Networking, you want to start over with new pairs. In other cases, during virtual classes, for example, you want to preserve a group composition for several exercises to help create familiarity and trust among the member of the breakout room group.
- As the host, you can terminate all breakout sessions simultaneously, thus reassigning all participants back to the main session.

## ***Breakout Rooms: What Needs to Improve***

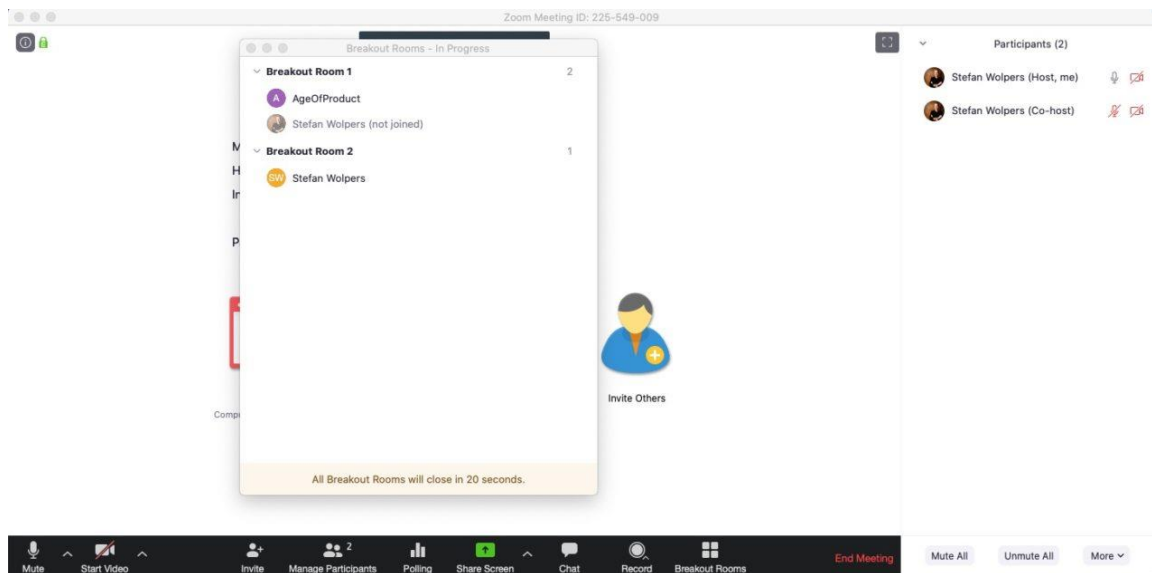
Where there is light, there is a shadow, too. Of course, Zoom's breakout rooms principally support many of the practices that Scrum Masters, agile coaches, or Product Owner need if facilitating remote agile events. However, there is quite some room for improvement:

- There is no audio or chat broadcast across all breakout rooms; all you can do is send every room a text message in the form of the pop-up window.
- As a facilitator, you cannot listen to discussions in breakout rooms. All you can do is enter a breakout room and participate regularly.

# The Remote Agile Guide



- Zoom cannot create breakout rooms of different sizes automatically. If you groups of different sizes you start with rooms of equal size, followed by manually assigning members from one breakout room to another.
- There is no way to “resend” the original room assignment. If a participant ignores the notification or discards it, you need to manually reassign that person to a different room first and then move that user back to the room he or she is supposed to enter. This way, the invitation is reproduced.
- There is no merger function of breakout rooms. For example, if you want to run a [1-2-4-All](#) via breakout rooms, you cannot automatically merge two pairs into a quartet. Again, you need to reassign one pair to another breakout room manually.
- Once triggered, you cannot abort the termination process for all breakout sessions. The only remedy here is to start another breakout session in the same room configuration as before.



As usual, there are some bugs, too. For example, as a host, I recently exited a breakout room to return to the main room, when I noticed that the breakout session administration panel was gone, while six sessions were happily proceeding. There was no way to regain access to the panel, so I had to end the Zoom meeting for all and restart it. (We lost about ten minutes that way.) Generally, the Zoom breakout room feature is quite stable.

## Conclusions

Despite all of its shortcomings, Zoom is—because of the breakout rooms—currently the best tool around to facilitate Liberating Structures in distributed teams, including everyone in the process and giving each of them a voice. Given the considerable increase in agile practitioners joining



# The Remote Agile Guide



Zoom at the moment, I am optimistic that we will see Zoom's breakout room feature improve further during the coming moments.

# The Remote Agile Guide

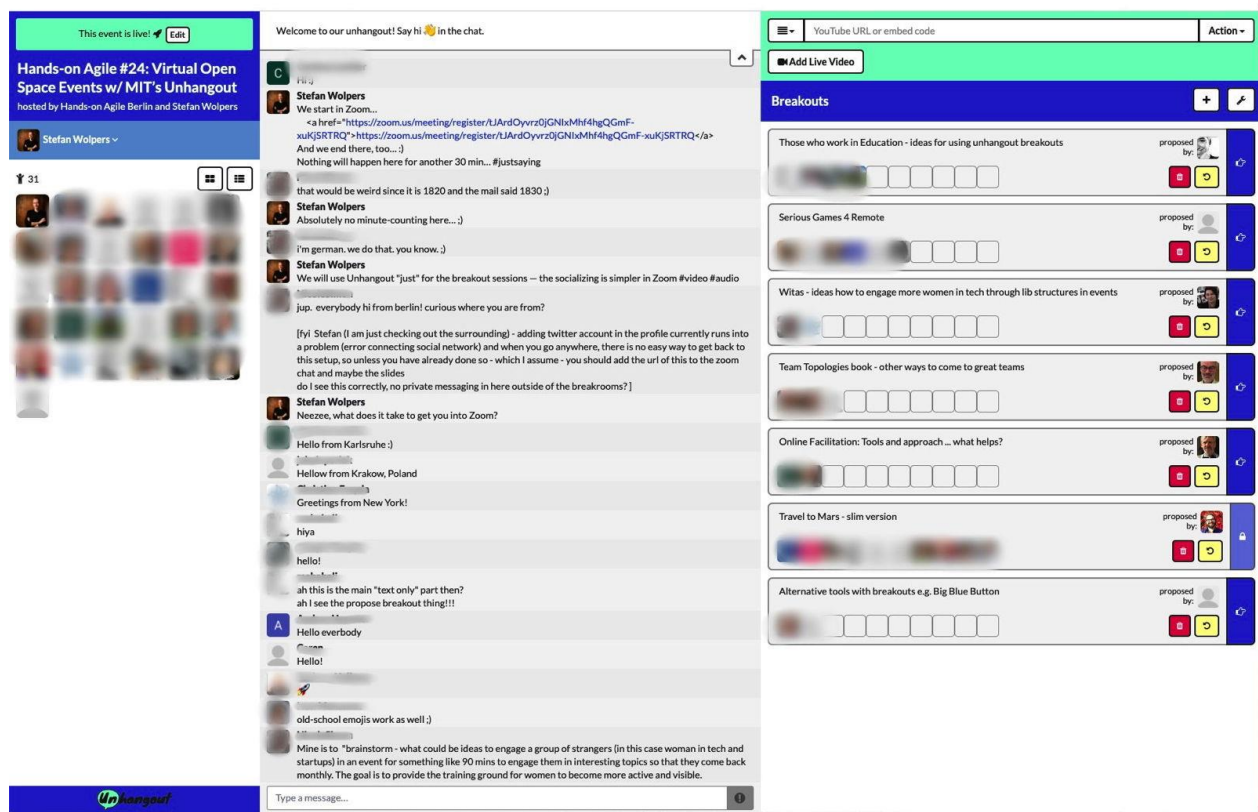


## Using Unhangout to Host a Virtual Barcamp or Open Space

### Introduction

Last week, 30-plus attendees of the 24th Hands-on Agile meetup ran a virtual Barcamp experiment w/ MIT's Unhangout, an open-source platform for organizing attendee-driven virtual open space events.

Read on and learn whether Unhangout is a suitable solution to remote collaboration challenges.



### Open Space Technology

BarCamps, unconferences — we know open space events under different labels. They have become very popular in the last years, either organized by an independent community or within organizations:

*When people must tackle a common complex challenge, you can release their inherent creativity and leadership as well as their capacity to self-organize. Open Space makes it possible to include everybody in constructing agendas and addressing issues that are important to them. Having co-created the agenda and free to follow their passion, people will take responsibility very quickly*

# The Remote Agile Guide



*for solving problems and moving into action. Letting go of central control (i.e., the agenda and assignments) and putting it in the hands of all the participants generates commitment, action, innovation, and follow-through. You can use Open Space with groups as large as a couple of thousand people!*

**Source:** [Liberating Structures: Open Space Technology](#).

How do open space events work in practice? The Agile Camp Berlin describes the magic of self-organization at work:

*Usually, people who share a common interest meet and work on topics. Meaning: Everybody can present a session, even is encouraged to do so. If there's a topic you want to present, discuss, try out or you just want to ask the community for help: the pitching session in the morning of the BarCamp gives you the opportunity to propose your topic. After the initial pitching of sessions, all proposed sessions will be mapped to the spots and rooms. After that, the law of the two feet will apply. Law of the two feet means: if you decide that a session might be valuable for you, you show this by attending it. Sometimes there's the situation that there are more proposed sessions than slots. In that case, we vote collectively, and the most popular sessions get a slot.*

**Source:** [Agile Camp Berlin: What is Barcamp?](#)

## Unhangout's Promise

Unhangout promises to solve a problem we are facing if we consider Zoom to run a virtual Barcamp: How do we ensure that the law-of-two-feet still applies to a virtual Barcamp? By making everyone a co-host in Zoom? Probably, that is not the best idea in some situations. Alternatively, Manually assigning people to sessions in breakout rooms would not just create a massive administrative overhead. It would also introduce a dependency that might threaten what makes open space events so successful: autonomy, self-organization, and serendipity.

This is where Unhangout comes into play to help organize a virtual Barcamp:

*Unhangout is an open-source platform for running large-scale, participant-driven events online. Each event has a landing page, which we call the lobby. When participants arrive, they can see who else is there and chat with each other. Hosts can welcome their community and do introductions in a video window that gets streamed into the lobby. Participants can then join breakouts, which are small group video chats, for in-depth conversations, peer-to-peer learning, and collaboration on projects.*

**Source:** [About Unhangout](#).

In other words: Unhangout's unique value proposition is the free movement of attendees of the virtual Barcamp between breakout sessions. And we put it to the test.

# The Remote Agile Guide

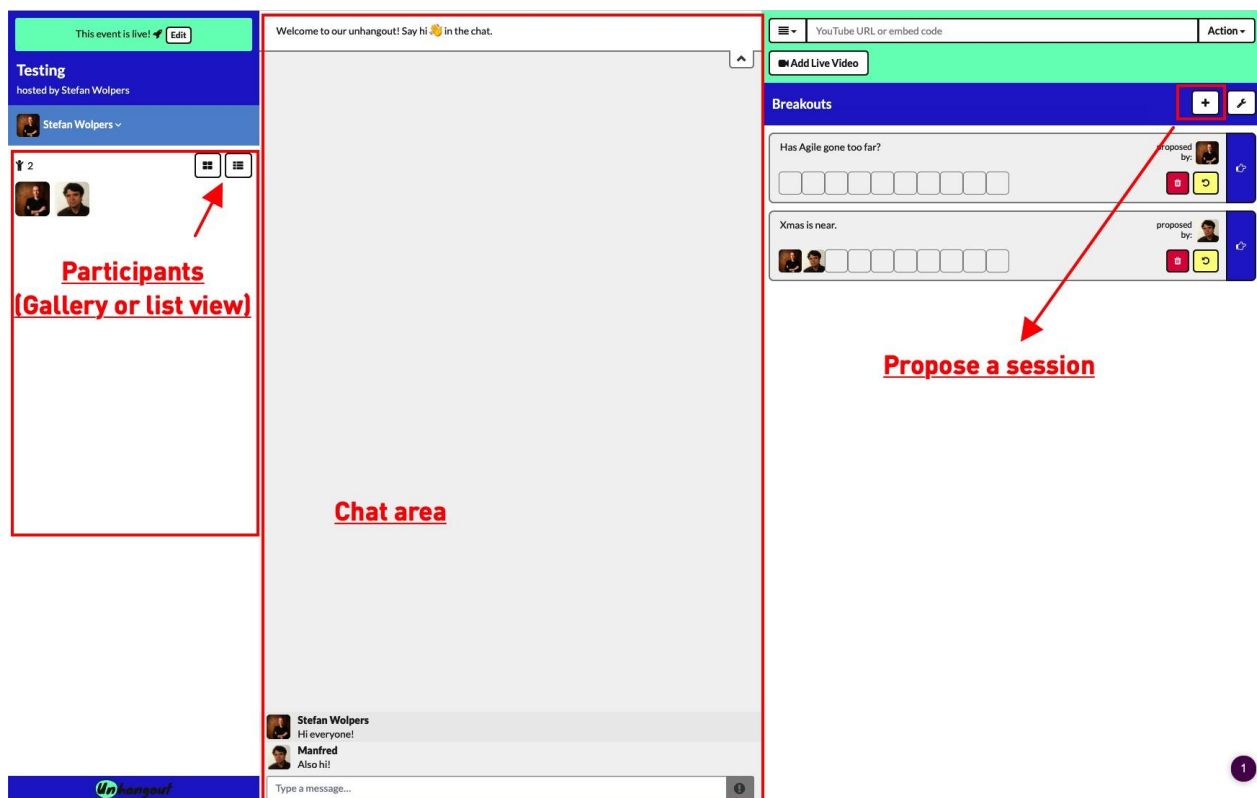


## Lessons Learned from Using Unhangout to Host a Virtual Barcamp at the 24th Hands-on Agile Meetup

### Unhangout's Features — An Overview

Unhangout comprises of two main areas: a) the lobby of the event and b) the breakout room. The following screenshots were taken on an exploratory session.

### Unhangout's Lobby



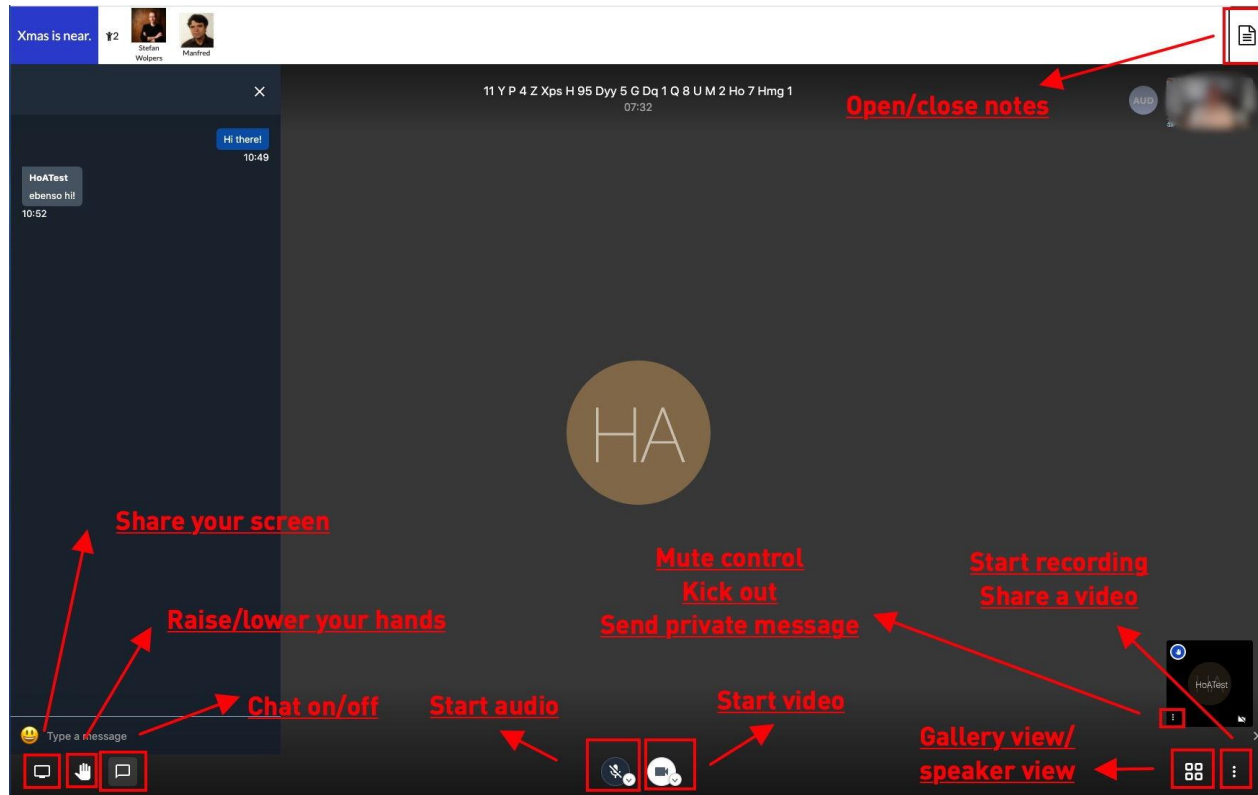
Unhangout's lobby is the space where all participants of the virtual Barcamp meet:

- On the left hand, there is a list of participants.
- In the middle of the window, you find the chat area. (No video chat, though.)
- On the right hand, there is the session planning area. (Here, we use the mode of participants to suggest breakout sessions.)

# The Remote Agile Guide



## Unhangout's Breakout Rooms



Unhangout's breakout rooms are where the action happens. If you know Zoom or Google Meet, you will feel familiar with the functionality on offer:

- You can share your screen, raise and lower your hands, and join the chat within the breakout session.
- You control audio and video, muting, and removing people from the session.
- Lastly, you can record sessions and share videos.
- Additionally, there is a notepad that the attendees of a breakout session can use.

## Unhangout's Shortcomings

There are several shortcomings of Unhangout that come to mind regarding the organization of a larger virtual Barcamp:

1. There is no video chat available in the lobby.
2. The number of participants of breakout sessions is limited to ten; that's hardly "supporting" the law-of-two-feet."
3. Consecutive sessions are not supported.
4. There is little control over breakout sessions on the side of the admin.
5. Live audio pitching of sessions before a vote is not supported.

# The Remote Agile Guide



6. The note pad does not seem to be session-specific.

While we could not address the #2 and #3, we decided to remedy the lack of a video chat in the lobby by moving the lobby to Zoom. We hence used Zoom as a wrapper application for the virtual Barcamp: We started with the kick-off in Zoom, then moved to Unhangout for the session planning and the breakout sessions, and closed the event in the Zoom lobby. However, engaging Zoom for this purpose caused audio and webcam issues in some configuration—it was either Zoom or Unhangout, but not both at the same time.

## The Script of the 24th Hands-on Agile Meetup

Organization-wise, we followed these subsequent steps—directly taken from the skript—to accomplish the test:

1. *Share the Unhangout link, share your Unhangout dashboard via Zoom:* Please join Unhangout via the following link.
2. *a) Round 1: Adding breakout sessions:* Please add your breakout session proposals by clicking the “plus” in the upper right corner. (3 min)
3. *b) Round 2: Introducing the breakout sessions:* If you have suggested a breakout session, please introduce your session to the whole group via Zoom. ( We go down the list from top to bottom. It is one minute per pitch.)
4. *c) Round 3: Voting:* If you haven’t yet voted in Unhangout, please do so now. You have one vote per proposal; clicking twice will remove your vote again.)
5. *Accept and open breakout sessions in Unhangout:* I will now “open” the breakout sessions, and you can join your preferred one by clicking on it. (The breakout session will open in a new browser tab.) Please remember: A breakout session can host between 2 and 10 participants.
6. *Run the breakout sessions:* Join your breakout session of choice; the sessions will last 30 minutes. Alternatively, stay in the lobby — our Zoom session — for a chat at the virtual coffee machine. Session hosts: decide on how to run your session. Please ask in advance for permission of the attendees if you intend to record your session.
7. *Close breakout sessions after 30 minutes:* Please rejoin the whole group in Zoom to collect feedback and discuss the outcome.
8. *Rate the experience w/ Unhangout:* How would you rate the experience with Unhangout? Please add your feedback to the Google slide deck. (5 min)
9. *Discussion:* Let’s discuss the findings. (10 min)

The whole exercise with a short debriefing/retrospective took about 2.5 hours, including a break. The number of participants peaked about half an hour into our virtual strategy session. (38 out of 102 who RSVPed.)

## A Virtual Barcamp w/ Unhangout: Discussion, Feedback, and Lessons Learned

These are the lessons learned from running a virtual Barcamp with Unhangout:

# The Remote Agile Guide



## *What Works Well in Unhangout*

The following [quotes regarding the positive sides of Unhangout](#) are from the participants:

1. Easy to use
2. Jitsi was working fine for me
3. Jitsi worked for me in this tool. (I had problems in the past.)
4. Worked well for me, no issues with Jitsi
5. Liked the overview of the sessions and the visibility of how many are in a session
6. Easy to set up the market place
7. Relative easy to use
8. Quality communications with new people with similar interests.

## *What Needs Improvement in Unhangout*

The following [quotes regarding the negative sides of Unhangout](#) are from the participants:

1. Jitsi can sometimes be a problem
2. First time joiners will have a hard time figuring everything out and will miss out on things  
→ make new testing for everybody in advance
3. Two video conference tools in parallel are hardware challenged
4. Audio was awkward in the first minute in Unhangout session until everyone closed their Zoom window
5. Need time to figure out how all things are working
6. Look and feel is quite clumsy
7. I didn't get that I needed to close Zoom, after that it worked
8. Difficult to maintain Zoom and another tool at the same time
9. A few people I could not understand during the Unhangout session, but I don't know why the audio was bad for them
10. In retrospect, I would ask participants to "leave computer audio" in the Zoom call and stop their video before joining the Unhangout session. The echo and webcam blocked in another application was an issue for us.

## Conclusions

Does Unhangout support a virtual Barcamp organization, a remote unconference, and a distributed open space event? Yes, it does as long as the group is small; 30 people can be well supported, particularly if they already know each other. Otherwise, a different set-up may prove to be more useful. The idea of utilizing Zoom as a wrapper for Unhangout worked okayish for the majority of participants. However, there are configurations where Zoom and Unhangout are colliding in their respective attempt to gain control over microphones and webcams.

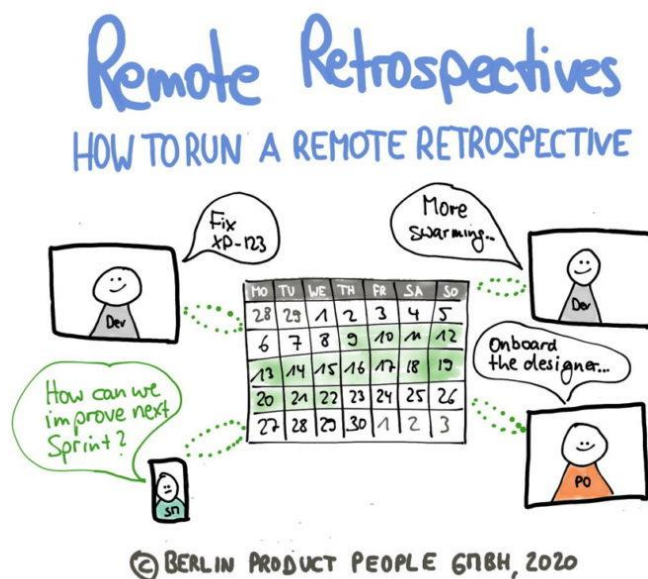


## Remote Scrum Events

### Remote Retrospectives

#### Introduction

We started this series on remote agile with looking into practices and tools, followed by exploring virtual Liberating Structures, how to master Zoom as well as common remote agile anti-patterns. This article now dives into organizing a remote Retrospective with a distributed team: practices, tools, and lessons learned.



### Remote Retrospectives with virtual Liberating Structures and Zoom Breakout Rooms

The following suggestions on how to handle a remote Retrospective as the host, facilitator, or Scrum Master are based on two assumptions: a) we use Zoom as a video application as we need to work in breakout rooms, and b) our Retrospectives are modeled around Liberating Structures strings. (The latter does not rule out utilizing other techniques from the agile toolbox, and there are plenty of those available on [Retromat](#) or [Tasty Cupcakes](#).)

### The Design Elements of Virtual Liberating Structures

Virtual Liberating Structures share a set of common design principles:



# The Remote Agile Guide



- **Breakout rooms** are used to divide the whole group of participants into smaller workgroups, starting with pairing up two participants. (I am using Zoom for that purpose.)
- **Muting/unmuting** is used — beyond the purpose of reducing noise — to mark different states of participants. For example, in the Conversation Café exercise during rounds 1, 2, and 4, everyone is muted except the individual that is sharing his or her thoughts.
- **Video on/off** is used to distinguish between roles, for example, between the inner circle and the outer circle of the User Experience fishbowl. Here, the outer circle members turn their video off as well as mute themselves.
- A **shared workspace** is needed to aggregate findings, for example, as the result of a 1-2-4-All session. This can be a simple Google slide or a [FunRetro.io](https://funretro.io) board.
- **Workbooks** are useful to provide participants with instructions when working in breakout rooms; for example, a detailed description of how an individual Liberating Structures works.
- A **chat channel** is used to facilitate communication within the whole group.

The following LS microstructures refer to these basic patterns of virtual Liberating Structures.

## The 5 Stages of a Retrospective

I modeled the following design of a remote Retrospective is after the five stages of Esther Derby and Diana Larsen’s book “[Agile Retrospectives](#).”

### *1. Setting the Stage*

- [Impromptu Networking](#) is a simple application of breakout rooms; just make sure that after each round, the pairs are created a new. Provide the invitation and the three questions in the workbook in advance.
- Organizing a [Mad Tea](#) in the virtual realm requires a different approach. Of course, we cannot recreate two concentric circles of attendees facing each other. However, what we can do is use the prompts—the half-sentences that the attendees shall complete—and the chat channel to create a quick and comprehensive picture of the team’s sentiment. As the moderator, prepare a few prompts in advance regarding the topic of the remote Retrospective, for example, “when I think of our recent Sprint, I...” Then post that prompt to the chat and ask the participants to add their answers but not to hit enter. That is done simultaneously by all attendees when the host asks for it. The result is a bunch of answers on the same topic in the chat. Allow the team to scan the answers and then move on to the second prompt.

# The Remote Agile Guide



- Use check-ins with emojis. (Industrial Logic has created a set of [SprintMojis](#) for that purpose.)
- Choose from a growing list of online icebreakers. (For example, see [Online Warm Ups & Energizers](#) or [10 Fun Virtual Icebreakers to Take Remote Working to the Next Level](#).)
- Consider crafting a working agreement for the upcoming meeting or workshop if the team has yet done so.

## II. Gathering Data

There are numerous ways of gathering data for an upcoming Retrospective. Probably, you want to track quantitative metrics like cycle-time or the number of bugs that escaped to production. Or you might be interested in qualitative metrics such as team-health or the sentiment of the team members. The point is that concerning the data, it does not matter whether the Scrum team runs the analysis in a face-to-face or remote setting: Both environments provide access to the same data. Typical practices of gathering data for Retrospectives are:

- Not only is **Impromptu Networking** an excellent way to create a sense of togetherness among the participants, but it is also a useful exercise to gather data if the invitation is crafted in the right way.
- **Anonymous surveys** provide an option to collect data during the Retrospective as well as in advance. Those surveys can be Sprint-specific, penciled-in between the Sprint Review and the Retrospective. Alternatively, they can be open-ended surveys such as a permanent suggestion box. Or, they are conducted at regular intervals to track progress in areas of interest. Suitable applications for this purpose are, for example, [Google Forms](#), [Typeform](#), or [SurveyMonkey](#). (I use to run anonymous surveys with Scrum teams after each Sprint Review, asking for the perceived value created during the recent Sprint, the state of technical debt, the employer NPS, and finally, the personal sentiment: are you happy, or are you looking for a new job?)
- A subset of the anonymous survey is the ‘**Team Radar**.’ It is a great way to create transparency about important team matters and track their development as time passes. (One team radar I regularly run with all Scrum teams, for example, is the Scrum Values radar.)
- Finally, you can **derive metrics** from supportive applications, for example, your **ticket system**. (Be careful, though, with the reports that are available out of the box. Often, those do not provide useful metrics specific to your situation.)

## III. Generate Insight from the Data

After collecting the data, making sense of it is next. The following three LS microstructures have proven to be useful, also in a remote setting:

# The Remote Agile Guide



- [What, So What, Now What?](#) is a sequence of individual work and group work based on breakout rooms, aggregating findings in shared workspaces to be shared with the whole group in the end.
- Again, [TRIZ](#) is a combination of basic elements of virtual Liberating Structures: breakout rooms, embedded 1-2-4-All, joined workspaces, Shift & Share when several groups are working on the problem. Consider time-keeping via the breakout room broadcasting function, as participants are likely to be highly engaged and may lose track of time.
- Use the [Conversation Café](#) by creating groups with the breakout room function, and identify a host for time-keeping. During rounds 1, 2, and 4, where one participant is talking while the others are listening, use mute for the listeners. Once the timebox has expired, the previously talking participant “hands over” the microphone by calling out the next one in line and then muting him- or herself. As the facilitator, also consider providing a matrix — rounds by speakers with checkboxes — to the hosts to ensure that everyone has a fair share of airtime.)

## IV. Deciding What to Do

The next step of the remote Retrospective is to agree on improvement items that will allow the team to grow and become more mature. Four Liberating Structures microstructures well-suited for this purpose:

- [15% Solutions](#): We use a similar procedure as with TRIZ. Consider aggregating all suggestions in the whole group’s shared workspace for clustering and ranking by voting. (I like to use a FunRetro.io board for that purpose: it is simple and does not need much explaining.)
- [25/10 Crowd Sourcing](#): This microstructure belongs to those that are hard to replicate online with the currently available tools. The following prototype is not yet satisfactory but pointing in the right direction: Use a form application to collect both suggestions from the team members on the subject in questions as well as their names. Once all participants have filled out the form, export the answers as a CSV-file and import this file into a FunRetro.io board. (The board has the voting disabled, and the number of votes is hidden.) As the facilitator, distribute the answers in packs of five to new columns and allocate the “name tags” of the participants randomly to each column in an even distribution. Then activate the voting and ask all participants to vote on the answers in the column they have been assigned to before. (Hence the “name tags.”) Set the number of available votes so high that every answer in a column can be awarded from 1 to 5 votes. Once the voting has ended, move all answers to one column and activate the “vote count.” Finally, sort that column by votes. (There are many issues with that process. For example, you have nine answers, not ten. Also, the attendees will be assigned to a column, which reduces the randomness of the voting.)

# The Remote Agile Guide



- [Lean Coffee](#) is an excellent example of a workaround for virtual Liberating Structures. Gather all the input in the usual way, for example, engaging in 1-2-4-All, and gather those on a FunRetro.io board while voting is turned off. (Use several columns if the whole group is large to speed up the gathering process.) Then ask the whole group to cluster similar topics, then turn on the voting and order the remaining entries by votes. For here, you continue with a whole group discussion, or you engage smaller groups with breakout rooms.
- [Ecocycle Planning](#): Principally, we apply the techniques as before, from breakout rooms to shared workspaces. Speaking of which, given the large number of “stickies” that you usually create during Ecocycle planning, you may want to consider a specialized online board application such as Miro or Mural. (Please note that both tools are not self-explanatory and require a prep session with participants to avoid frustrating them.)

## V. Closing the Retrospective

The last step of the Retrospective sequence is the closing or check-out. Basically, it is a mini-retrospective within the “large” remote Retrospective focused on reflecting on what has happened as well as providing feedback: Was the time well-spent or do we need to change our approach to running a remote Retrospective next time? To entice this kind of feedback, keeping the process short and simple is paramount. Although we do not pass a door while leaving the meeting room, there are many ways to collecting the feedback of the team members:

- We can replicate the door sticky practice with the annotation tool of the video application on a prepared graphic. All at once, attendees leave a symbol on a scale from □ to □ to □.
- Then some applications allow for gathering live feedback, such as [Poll Everywhere](#).
- Alternatively, run a [Fist of Five voting](#). (Make sure, though, that everyone knows the scale in advance. For example, include the scale in the working agreement if you use this voting practice regularly.)

## Supporting Liberating Structures

Two Liberating Structures can support a remote Retrospective with a larger team: [1-2-4-All](#) and [Shift & Share](#):

- To cover 1-2-4-All, we need breakout rooms and a place to aggregate the findings. We start with everyone in the whole group for a minute in silence; then, we split the whole group into pairs using Zoom’s breakroom feature for 2 minutes. After that round, we merge two pairs into a group of four for five minutes—this has to be done manually by the host—and the group aggregates its findings, for example, on a Google sheet prepared for each group in advance. We can introduce each group’s findings to the whole group by

# The Remote Agile Guide



screen sharing in a Shift & Share.

- Shift & Share: Each workgroup presents its findings to the whole group by screen sharing. Alternatively, if the shared workspace has been created in advance, for example, Google Slides with a slide per workgroup, the moderator can share his or her screen while someone from the team is explaining the findings to the whole group. This reduces the stress of switching screen sharing on and off among several groups.

## Applications to Run a Remote Retrospective

Of course, instead of tailoring a string of Liberating Structures to host a remote Retrospective, there are other options:

- There are specialized Retrospective applications for distributed teams. Well-known providers are [Retrium](#), FunRetro, or TeamRetro.
- Then there are digital workspace applications, namely digital whiteboards like Miro and Mural, that often have prefabricated templates for Retrospectives. Another example would be Atlassian's Confluence that also provides a simple template that aligns with the Scrum Guide's purpose of a Retrospective.
- Finally, workarounds based on general-purpose collaboration tools such as Google Docs or Office 365 also allow running at least rudimentary remote Retrospectives.

## Good Practices for a Remote Retrospective

From the list of all practices that generally apply to remote agile events, see [Remote Agile \(Part 1\): Practices & Tools for Scrum Masters, Agile Coaches, and Product Owners](#), I want to point at three practices that make hosting significantly easier:

- Create a script with the probable time-line of the remote Retrospective in advance, including all the required documents to be shared with the participants and all the copy you need to provide to the chat during the session.
- Document the outcome of the remote Retrospective so team members can revisit them at a later stage. Restrict access to sensitive information by limiting access privileges strictly to team members. (Be prepared to explain the necessity of this procedure to curious or demanding line managers.)
- Keep good track of action items. Without a prominent placement in the team room, improvement items tend to be forgotten. ("When you put problem in a computer, box hide answer. Problem must be visible!" [Hideshi Yokoi](#), former President of the Toyota Production System Support Center in Erlanger, Kentucky, USA.)

# The Remote Agile Guide



## Antipatterns of Remote Agile Retrospectives

There are plenty of [Retrospective anti-patterns](#) in general. However, I want to point at a few anti-patterns of Scrum Masters that are particularly relevant for a remote Retrospective:

- **Waste of time:** The Retrospective provides a poor return on investment. (Probably, it is insufficiently prepared, or the team lacks the required skills to have a meaningful remote Retrospective.)
- **Prisoners:** Some team members only participate because they are forced to join. (Don't pressure anyone to take part in a retrospective. Instead, make it worth their time. The drive to continuously improve as a team needs to be fueled by intrinsic motivation, neither by fear nor by force. My tip: Retromat's "[Why are you here?](#)" exercise is a good opener for a retrospective from time to time, see above.)
- **No psychological safety/bullying:** A few participants dominate the Retrospective, while the more introverted team members pull back, and the host/Scrum Master does confront this misbehavior. (Working remotely requires the host to become more assertive in some situations; a laissez-faire Scrum Master is usually not up to that job.)
- **Groundhog day:** A useful routine has been turned into a mindless ritual. (If you run the same-style Retrospective format over and over again, do not be surprised if a) the team is no longer improving its way of working, and b) the mood is turning sour. The problem for the host is that this effect happens in a remote setting significantly faster by comparison to a face-to-face Retrospective—remote Agile speeds up the revelation of collaboration issues.)

**Read more:** [21 Sprint Retrospective Anti-Patterns Impeding Scrum Teams.](#)

## Conclusions

Working as a distributed agile team is, in many aspects, more difficult than being co-located: 'Reading the room' is significantly more complicated, for example, and communication is taking a toll as the beloved informal meeting at the coffee machine is no longer happening. However, being suddenly distributed does not mean that we cannot have useful critical events anymore. On the contrary: The necessity to invest more preparatory work upfront may provide a chance to improving the meaning of events, and I would consider the remote Retrospective to be a prime candidate for that purpose.



## Remote Sprint Planning

### Introduction

We started this series on remote agile with looking into practices and tools, followed by exploring virtual Liberating Structures, and how to master Zoom. We had a look at common remote agile anti-patterns, and we analyzed remote Retrospectives based on Liberating Structures. This article now dives into organizing a remote Sprint Planning with a distributed team: practices, virtual Liberating Structures, and lessons learned.



### The Purpose of the Sprint Planning

The purpose of the Sprint Planning is to align the Development Team and the Product Owner on what to build next, delivering the highest possible value to customers. The Product Owner introduces the business objective the upcoming Sprint is supposed to meet, the Scrum Team collaboratively creates a Sprint Goal, and the Development Team forecasts the work required to achieve the goal by picking the appropriate items from the Product Backlog, transferring them to the Sprint Backlog. Also, the Development Team needs to come up with a plan on how to accomplish its forecast as well as pick at least one high priority improvement issue from the previous Sprint Retrospective.

According to the Scrum Guide, the Sprint Planning answers two questions:

1. *What can be delivered in the Increment resulting from the upcoming Sprint?*
2. *How will the work needed to deliver the Increment be achieved?*

# The Remote Agile Guide



Given the relative distance between the members of a distributed Scrum Team, the task of aligning everyone becomes even more crucial for a remote Sprint Planning as is the determination of the Sprint Goal:

*During Sprint Planning the Scrum Team also crafts a Sprint Goal. The Sprint Goal is an objective that will be met within the Sprint through the implementation of the Product Backlog, and it provides guidance to the Development Team on why it is building the Increment.*

By comparison to a co-located Scrum Team, the preparatory work, for example, regular Product Backlog refinements, keeping track of technical debt, and preserving a high-quality Definition of ‘Done’ is paramount to securing a successful outcome of the Sprint.

Source: [Scrum Guide, 2017](#).

## Virtual Liberating Structures for a Remote Sprint Planning

To ensure that the physical barrier is not impeding a remote Sprint Planning, and similar to the [remote Retrospective](#), we once again turn to virtual Liberating Structures to include and engage all Scrum Team members, ensuring that everyone has a voice.

For a remote Sprint Planning, the following Liberating Structures microstructures have proven to be supportive:

- The [Celebrity Interview](#) is a useful way for the Product Owner to introduce the upcoming Sprint’s business objective. Working in the whole group, use mute and video off to distinguish between the Product Owner, the interviewer—a member of the Development Team—and the remaining members of the Scrum Team. Gather additional questions through the chat channel from the listeners as the interview proceeds. The interviewer should pass on these new questions in due time. (During the discussion, the Product Owner should not read these new questions at the same time.)
- Use a [Conversation Café](#) to discuss the purpose of the upcoming Sprint: Create groups with the breakout room function, and identify a host for time-keeping. During rounds 1, 2, and 4, where one participant is talking while the others are listening, use mute for the listeners. Once the timebox has expired, the previously talking participant “hands over” the microphone by calling out the next one in line and then muting him- or herself. As the facilitator, also consider providing a matrix — rounds by speakers with checkboxes — to the hosts to ensure that everyone has a fair share of airtime.)
- Consider running a [Mad Tea](#) to identify a Sprint Goal based on the business objective and the introduction of the Product Owner. Of course, we cannot recreate two concentric circles of attendees facing each other. However, what we can do is use a prompt—a half-sentence that the team members shall complete—and the chat channel to create a quick picture of the team’s sentiment on the Sprint Goal. As the moderator, prepare a few



# The Remote Agile Guide



prompts in advance regarding the topic of the upcoming Sprint, for example, “I think the Sprint Goal should be...” Then post that prompt to the chat and ask the participants to add their answer(s) but not to hit enter. That is done simultaneously by all attendees when the host asks for it. The result is a bunch of suggestions on the Sprint Goal. Repeat the exercise if no suggestion finds unanimous support.

- Alternatively, a [25/10 Crowd Sourcing](#) can also support the creation of a Sprint Goal: This microstructure belongs to those that are hard to replicate online with the currently available tools. The following prototype is not yet satisfactory but pointing in the right direction: Use a form application to collect both suggestions from the team members on the subject in questions as well as their names. Once all participants have filled out the form, export the answers as a CSV-file and import this file into a FunRetro.io board. (The board has the voting disabled, and the number of votes is hidden.) As the facilitator, distribute the answers in packs of five to new columns and allocate the “name tags” of the participants randomly to each column in an even distribution. Then activate the voting and ask all participants to vote on the answers in the column they have been assigned to before. (Hence the “name tags.”) Set the number of available votes so high that every answer in a column can be awarded from 1 to 5 votes. Once the voting has ended, move all answers to one column and activate the “vote count.” Finally, sort that column by votes. (There are many issues with that process. For example, you have eight suggestions for the next Sprint Goal, not ten.)
- [Min Specs](#): Min Specs is an excellent exercise to focus the whole team on the essential work to accomplish the next Sprint Goal. In a remote setting, it is a sequence of individual work and group work based on breakout rooms, aggregating findings in shared workspaces to be shared with the whole group in the end. Min Specs work well with embedded 1-2-4-All and Shift & Share, see below.
- If you need to balance the demands of the Product Owner for new features with the necessity of the Development Team to keep technical debt at bay, probably already a latent conflict, consider running an [Integrated~Autonomy](#) session and ask “How is it that we can be more integrated and more autonomous at the same time?” As Integrated~Autonomy builds on a sequence of small group work based on 1-2-4-All, breakout rooms, and a shared workspace to visualize the outcome, it is also well-suited for a remote Sprint Planning.

## Supporting Liberating Structures

Three Liberating Structures can support a remote Sprint Planning with a larger team: [1-2-4-All](#), [Shift & Share](#), and [Lean Coffee](#):

- To cover **1-2-4-All**, we need breakout rooms and a place to aggregate the findings. We start with everyone in the whole group for a minute in silence; then, we split the whole group into pairs using Zoom’s breakout room feature for 2 minutes. After that round, we

# The Remote Agile Guide



merge two pairs into a group of four for five minutes—this has to be done manually by the host—and the group aggregates its findings, for example, on a Google sheet prepared for each group in advance. We can introduce each group’s findings to the whole group by screen sharing in a Shift & Share.

- **Shift & Share:** Each workgroup presents its findings to the whole group by screen sharing. Alternatively, if the shared workspace has been created in advance, for example, Google Slides with a slide per workgroup, the moderator can share his or her screen while someone from the team is explaining the findings to the whole group. This reduces the stress of switching screen sharing on and off among several groups.
- **Lean Coffee** is an excellent example of a workaround for virtual Liberating Structures. Gather all the input in the usual way, for example, engaging in 1-2-4-All, and gather those on a FunRetro.io board while voting is turned off. (Use several columns if the whole group is large to speed up the gathering process.) Then ask the whole group to cluster similar topics, then turn on the voting and order the remaining entries by votes. For here, you continue with a whole group discussion, or you engage smaller groups with breakout rooms.

## Remote Sprint Planning Antipatterns

There are plenty of [Sprint Planning anti-patterns](#) in general. However, I want to point at a few anti-patterns of Scrum Masters that are particularly relevant for a remote Sprint Planning:

- **What are we fighting for?** The Product Owner cannot align the business objective of the upcoming Sprint with the overall product vision. (Alignment is a challenging endeavor in the best of times; at a remote Sprint Planning, it is essential to preserve cohesion among the team members. A serious goal answers the “What are we fighting for?” question. To a certain extent, it is also a negotiation between the Product Owner and the Development Team. It is focused and measurable, as the Sprint goal—based on the business objective—and Development Team’s forecast go hand in hand.)
- **The remote Sprint Planning is ignored:** The Development Team is not participating collectively in the Sprint Planning. Instead, two team members, for example, the “tech lead” and “UX lead,” represent the team so the others can “work.” (The remote Sprint Planning is a critical opportunity for alignment between all team members, the product vision, and the business, and therefore mandatory. As far as the idea of one or two “leading” teammates in a Scrum Team is concerned, there are none, see above. And unless you are using Nexus or LeSS—no pun intended—where teams are represented in the overall Sprint Planning, the whole Scrum Team needs to participate. It is a team effort, and everyone’s voice hence needs to be heard. Otherwise, transparency suffers, and flawed decisions might be made, reducing value creation and increasing risk.)

# The Remote Agile Guide



- **Planning too detailed:** During the Sprint Planning, the Development Team plans every single task of the upcoming Sprint. (This might be triggered by a subconscious fear of loss of control due to the remote nature of the work. However, as in real life, there is no need to become too granular. One-quarter of the tasks are more than sufficient to not just start with the Sprint, but also start learning. The Sprint Backlog is emergent, and doing too much planning upfront might result in waste. Just embrace the Prime Directive.)
- **Too little planning:** The Development Team is skipping planning altogether, as working as a distributed team led to everything being now well-documented in the Product Backlog. (Skipping planning is unfortunate, as it is also an excellent opportunity to talk about how to spread knowledge within the Development Team, where the architecture is heading, or whether tools are adequate. For example, the team might also consider who will be pairing with whom on what task — which is even more essential in a remote working environment. The Development Team planning part is also well-suited to consider how to reduce technical debt.)
- **Over-commitment:** The team takes on way too many tasks for a distributed team and moves unfinished work directly to the next Sprint. (Over-commitment is problematic for co-located teams, but it becomes even more pressing for distributed Scrum teams, as the additional overhead needs to be considered. If two or three items spill over to the next Sprint while the Development Team meets the Sprint Goal, so be it. If regularly 30 to 40 percent of the original forecast is not delivered during the Sprint, the Scrum Team may have created a kind of ‘time-boxed Kanban.’ Maybe, this is the right moment to ask the Scrum Team whether moving to Kanban might be an alternative. If the team considered Scrum still to be its choice, I would recommend to put more energy into Product Backlog refinement and creating meaningful Sprint Goals.)

**Read more:** [21 Sprint Retrospective Anti-Patterns Impeding Scrum Teams.](#)

## Conclusion

Garbage in, garbage out: The task of aligning everyone becomes even more crucial for a remote Sprint Planning by comparison to the planing of a co-located Scrum team. The good news is that virtual Liberating Structures can help to overcome this communication disadvantage of a distributed agile team by visualizing issues and giving every member of the Scrum team a fair share of airtime.



## Remote Sprint Review

### Introduction

We started this series on remote agile with looking into practices and tools; we explored virtual Liberating Structures, and how to master Zoom. We had a look at common remote agile anti-patterns, and we analyzed remote Retrospectives and Sprint Plannings based on Liberating Structures. This seventh article now looks into organizing a remote Sprint Review with a distributed team: How to practice the review with virtual Liberating Structures, including and giving a voice to team members, stakeholders, and customers.



### The Purpose of the Sprint Review

Before we get into recreating the Sprint Review online in a virtual setting, let us first revisit the purpose of the Sprint Review. The Sprint Review is Empiricism at work: inspect the Product Increment and adapt the Product Backlog. The Development Team, the Product Owner, and the stakeholders need to figure out whether they are still on track delivering value to customers. It is the best moment to create or reaffirm the shared understanding among all participants whether the Product Backlog is still reflecting the best use of the Scrum Team's resources, thus maximizing the value delivered to customers. It is also because of this context that calling the Sprint Review a "sprint demo" does not match its importance for the effectiveness of the Scrum Team.

The Sprint Review is thus an excellent opportunity to talk about the general progress of the product. The Sprint Review's importance is hence the reason to address Sprint Review anti-patterns as a Scrum Master as soon as possible.



## What Does the Scrum Guide Say about the Sprint Review?

In contrast to other Scrum events, the Scrum Guide goes into detail regarding the Sprint Review. The Sprint Review includes the following elements (quote):

- *Attendees include the Scrum Team and key stakeholders invited by the Product Owner;*
- *The Product Owner explains what Product Backlog items have been “Done” and what has not been “Done;”*
- *The Development Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved;*
- *The Development Team demonstrates the work that it has “Done” and answers questions about the Increment;*
- *The Product Owner discusses the Product Backlog as it stands. He or she projects likely target and delivery dates based on progress to date (if needed);*
- *The entire group collaborates on what to do next so that the Sprint Review provides valuable input to subsequent Sprint Planning;*
- *Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next; and,*
- *Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated releases of functionality or capability of the product.*

*“The result of the Sprint Review is a revised Product Backlog that defines the probable Product Backlog items for the next Sprint. The Product Backlog may also be adjusted overall to meet new opportunities.”*

Source: [Scrum Guide 2017](#).

## Virtual Liberating Structures for a Remote Sprint Planning

The Sprint Planning indeed features a level of bringing everyone onto the same page, although this should not be confusing with a reporting session. Nevertheless, it seems to be tempting to many teams to handle Sprint Reviews precisely like that: As a team, show that you have been a useful entity of the organization by delivering the feature requests to the stakeholders that are paying for the team. This attitude tends to prevail in less “agile” organizations that are still in the early phases of a change, often fostered by a traditional management style rooted in the industrial paradigm, thus focused on output and the utility of the individual.

While this scenario is problematic when working co-located, its inherent lack of trust is seriously impeding a Scrum team’s ability to create value for customers in a remote scenario. Scrum is an excellent probe for organizational dysfunctions in normal, more or less co-located corporate life. Moving everything into the virtual realm is like putting the organization’s culture under a microscope: Remote Scrum reveals all shortcomings, problems, and issues at a much faster pace.

# The Remote Agile Guide



Therefore, to mitigate the risk of running a remote Sprint Review, for many Scrum teams applying Liberating Structures is sound business advice. Including everyone in the process, giving everyone a voice in the remote Sprint Review builds the trust to cope not just with the competition of the markets but also with the challenges of working remotely. Hence the following virtual Liberating Structures microstructures offer a useful start to learn, inspect, and adapt how to run a remote Sprint Review.

## Virtual Sprint Review Part 1: What Has Scrum Team Accomplished during the Sprint?

Consider the following microstructures for the first phase of the remote Sprint Review:

- [Shift & Share](#) — the science fair approach: Members of the Scrum Team present different elements of the Product Increment to the whole group by screen sharing. (While Shift & Share properly includes the developers, the stakeholders might remain passive. To address that issue, see below: Simple Ethnography.)
- [Simple Ethnography](#): Why present the outcome to the audience, when the stakeholders can explore the Product Increment on their own while the Scrum Team is carefully observing what is happening? (If the stakeholders are thinking loud during the exploration, it will improve the Scrum Team's capability to understand whether the Product Increment is meeting expectations and where there is room for improvement. Simple Ethnography uses screen sharing and session recording for later analysis and is hence simple to move to a remote work environment.)
- [User Experience Fishbowl](#): The UX Fishbowl is an effective follow-up to Simple Ethnography. Working in the whole group, use mute and video off to distinguish between the inner circle — the stakeholders — and the outer circle — the Scrum Team — of the User Experience fishbowl. Here, the outer circle members turn their video off as well as mute themselves. Gather additional questions through the chat channel from the out circle members. A facilitator should pass on these new questions in due time. (While discussing the topic at hand, the inner circle members should try not to read these new chat messages at the same time.) Use W3 to debrief in the whole group.

## Virtual Sprint Review Part 2: Is Our Product Backlog Prepared for the next Sprint?

Consider the following microstructures for the second phase of the remote Sprint Review:

- The [Celebrity Interview](#) is a useful way for the Product Owner to repeat the upcoming Sprint's business objective as well as the longterm objectives, thus encouraging reflection on whether the Product Backlog is still up to its task. Working in the whole group, use mute and video off to distinguish between the Product Owner, the interviewer—a member

# The Remote Agile Guide



of the Development Team or one of the stakeholders—and the remaining participants. Gather additional questions through the chat channel from the listeners as the interview proceeds. The interviewer should pass on these new questions in due time. (During the discussion, the interviewee should not read these new questions at the same time. The Celebrity Interview is also an excellent format to interview, for example, the CEO, when a shift in the organization’s product strategy has occurred or is imminent.)

- [Min Specs](#): Min Specs is an excellent exercise to focus the whole team on the essential work to accomplish the next Sprint Goal. In a remote setting, it is a sequence of individual work and group work based on breakout rooms, aggregating findings in shared workspaces to be shared with the whole group in the end. Min Specs work well with embedded 1-2-4-All and Shift & Share, see below.
- If you need to balance the demands of the Product Owner for new features with the necessity of the Development Team to keep technical debt at bay, probably already a latent conflict, consider running an [Integrated~Autonomy](#) session and ask “How is it that we can be more integrated and more autonomous at the same time?” As Integrated~Autonomy builds on a sequence of small group work based on 1-2-4-All, breakout rooms, and a shared workspace to visualize the outcome, it is also well-suited for a remote Sprint Planning.

## Virtual Sprint Review Part 3: Closing

To close the remote Sprint Review, consider running a [virtual Mad Tea](#) to identify areas of improvement for the upcoming Sprint Retrospective. Of course, we cannot recreate two concentric circles of attendees facing each other. However, what we can do is use a prompt—a half-sentence that the team members shall complete—and the chat channel to create a quick picture of the team’s sentiment. As the moderator, prepare a few prompts in advance regarding the upcoming Sprint Review, for example, “I think the next Sprint Review should be...” Then post that prompt to the chat and ask the participants to add their answer(s) but not to hit enter. That is done simultaneously by all attendees when the host asks for it. The result is a bunch of suggestions from the stakeholders and the Scrum Team members regarding the upcoming Sprint Review, serving as data for the next Sprint Retrospective.

**Learn more about remote Liberating Structures:** [Remote Agile \(Part 2\): Virtual Liberating Structures](#).

## Supporting Liberating Structures

Two Liberating Structures can support a remote Sprint Review with a larger team: [1-2-4-All](#), and [Lean Coffee](#):

- To cover **1-2-4-All**, we need breakout rooms and a place to aggregate the findings. We start with everyone in the whole group for a minute in silence; then, we split the whole

# The Remote Agile Guide



group into pairs using Zoom’s breakroom feature for 2 minutes. After that round, we merge two pairs into a group of four for five minutes—this has to be done manually by the host—and the group aggregates its findings, for example, on a Google sheet prepared for each group in advance. We can introduce each group’s findings to the whole group by screen sharing in a Shift & Share.

- **Lean Coffee** is an excellent example of a workaround for virtual Liberating Structures. Gather all the input in the usual way, for example, engaging in 1-2-4-All, and gather those on a FunRetro.io board while voting is turned off. (Use several columns if the whole group is large to speed up the gathering process.) Then ask the whole group to cluster similar topics, then turn on the voting and order the remaining entries by votes. For here, you continue with a whole group discussion, or you engage smaller groups with breakout rooms.

## Remote Sprint Review Antipatterns

There are plenty of [Sprint Planning anti-patterns](#) in general. However, I want to point at a few anti-patterns that are particularly relevant for a remote Sprint Review. It is all about embracing the new reality and trusting in people:

- **Death by PowerPoint:** Participants are bored to death by PowerPoint presentations. (The foundation of a successful Sprint Review is “show, don’t tell,” or even better: let the stakeholders drive the discovery.)
- **Scrum à la stage-gate®:** The Sprint Review is a kind of stage-gate® approval process where stakeholders sign off features. (This Sprint Review anti-pattern is typical for organizations that use an “agile”-waterfall hybrid. However, it is the prerogative of the Product Owner to decide what to ship when.)
- **Cheating:** The Development Team shows items that are not “done.” (There might be a good reason to show unfinished work on some occasions. Selling partially finished work as good to go, however, violates the concept of “Done,” one of Scrum’s first principles.)
- **No customers:** External stakeholders—also known as customers—do not attend the remote Sprint Review. (Break out of your organization’s echo chamber, and invite some paying users to your Sprint Review. Inviting customers is even more important concerning the challenges that remote product discovery imposes on the value creation process.)
- **No stakeholders:** Stakeholders do not attend the Sprint Review. (In the remote realm, this is the ultimate dysfunction. Of course, there are several reasons why stakeholders do not participate in the Sprint Review: they do not see any value in the event, or it is conflicting with another important meeting. They do not understand the importance of the Sprint Review event. No sponsor is participating in the Sprint Review, for example, from the C-level. To my experience, you need to “sell” the event within the organization, at least at



# The Remote Agile Guide



the beginning of using Scrum. When working in a distributed environment, consider doubling down on fixing this issue.)

**Read more:** [15 Sprint Review Anti-Patterns Holding Back Scrum Teams.](#)

## Conclusion

Sprint Reviews are challenging at the best of times, and moving them into the virtual world compounds the difficulties. Given that the flow of information and the discovery of opportunities is likely to drop at the beginning of a Scrum Team's journey to remote work, the importance of the remote Sprint Review has just increased significantly. Scrum Team, it is your turn now to overcome these impediments.

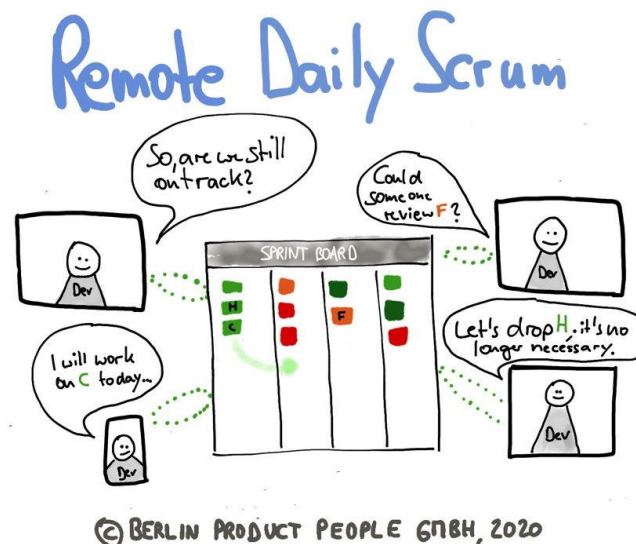
# The Remote Agile Guide



## Remote Daily Scrum

### Introduction

We started this series on remote agile with looking into practices and tools; we explored virtual Liberating Structures, and how to master Zoom. We had a look at common remote agile anti-patterns; we analyzed remote Retrospectives, Sprint Plannings as well as remote Sprint Reviews based on Liberating Structures. This eighth article now looks into supporting a distributed Development Team organizing a remote Daily Scrum.



### The Purpose of the Daily Scrum

The purpose of the Daily Scrum is clearly described in the Scrum Guide — no guessing is necessary:

*The Daily Scrum is a 15-minute time-boxed event for the Development Team. The Daily Scrum is held every day of the Sprint. At it, the Development Team plans work for the next 24 hours. This optimizes team collaboration and performance by inspecting the work since the last Daily Scrum and forecasting upcoming Sprint work. The Daily Scrum is held at the same time and place each day to reduce complexity.*

*The Daily Scrum is an internal meeting for the Development Team. If others are present, the Scrum Master ensures that they do not disrupt the meeting.*

# The Remote Agile Guide



*Daily Scrums improve communications, eliminate other meetings, identify impediments to development for removal, highlight and promote quick decision-making, and improve the Development Team's level of knowledge. This is a key inspect and adapt meeting.*

**Source:** [Scrum Guide 2017](#).

The Daily Scrum is an essential event for inspection and adaptation, run by the Development Team, and guiding it for the next 24 hours on its path to achieving the Sprint Goal. The Daily Scrum is hence the shortest planning horizon in Scrum and thus mandatory.

## Virtual Liberating Structures for a Remote Daily Scrum

Contrary to popular belief, the Daily Scrum's 15-minute timebox is not intended to solve all the issues addressed during the event. It is about creating transparency, thus triggering the inspection. If an adaptation of the Sprint plan or the Sprint Backlog, for example, is required, the Development Team is free to handle the resulting issues at any time. In my experience, most Daily Scrum issues result from a misunderstanding of this core principle. We hence look at the Daily Scrum's two parts separately.

### *Virtual Daily Scrum Part 1: Are We Still on the Right Track to Accomplish the Sprint Goal?*

This is the 15-minute time-box where the members of the Development Team inspect the progress towards achieving the Sprint Goal since the previous remote Daily Scrum. To structure this part of the remote Daily Scrum, the Scrum Guide suggests a non-mandatory information-sharing pattern — the three questions —, that pretty well translates to a remote set-up.

One way of applying this pattern to a remote Daily Scrum is to run a [virtual Mad Tea](#) to collect and distribute information from all developers rapidly among the team members. Of course, we cannot recreate two concentric circles of attendees facing each other. However, what we can do is use the set of questions to create a prompt—a half-sentence that the team members shall complete—and the chat channel to create a quick picture of the team's sentiment.

As the moderator, prepare the necessary prompts in advance regarding the progress of the team towards the Sprint Goal, for example, “Today, I will support the Development Team by contributing...” Then post that prompt to the chat and ask the participants to add their answer(s) but not to hit enter. That is done simultaneously by all attendees when the moderator asks for it. This way, the Development Team can generate insight synchronously at a rapid pace, identifying potential areas where more collaboration is needed to ensure that the Scrum Team is meeting the Sprint Goal.

### *Virtual Daily Scrum Part 2: What I Need from You*

Let us move on to the part of the remote Daily Scrum, where the Development Team collaborates on solving issues that might impede its ability to achieve the Sprint Goal. Consider the following

# The Remote Agile Guide



microstructures for the second phase of the remote Daily Scrum, when Development Team members may need individual support, or where the whole team needs to decide how to adapt in the light of new learnings:

- [Troika Consulting](#) proves to be especially helpful at sourcing support at the individual level, for example, from other Development Team members on how to solve a technical issue. To run it in a distributed team, we start by creating breakout rooms for groups of three. Consultants and the consultee have the initial conversation; then, the consultee turns around on his or her chair for the consulting phase. Alternatively, both consultants stop broadcasting their video, so the consultee is just listening to what they have to say.
- [Min Specs](#) has proven to be a very effective virtual LS microstructure when it becomes apparent that the original plan to achieve the Sprint Goal is no longer valid and needs to be trimmed, probably in collaboration with the Product Owner. Like What, So What, Now What?, a remote Min Specs is a sequence of individual work and group work based on breakout rooms, aggregating findings in shared workspaces to be shared with the whole Development Team in the end.
- In an inexperienced, new Development Team in its early team-building phase or structures centered around component teams or challenging software architectures, '[What I Need From You \(WINFY\)](#)' could offer the right frame to articulate core needs to achieve the Sprint Goal and coordinate and involve everyone in the collective effort. Again, in a remote setting, we employ breakout rooms, the chat, and shared workspaces. Embedded 1-2-4-All or What, So What, Now What? may support the effort as well.
- [TRIZ](#) may prove to be useful when the team needs to reaffirm its plan to achieve the Sprint Goal, for example, because of outside pressure. Again, TRIZ is a combination of basic elements of virtual Liberating Structures: breakout rooms, embedded 1-2-4-All, and joined workspaces. As a Scrum Master, you may consider supporting the effort by time-keeping on behalf of the participants as they are likely to lose track of time.
- Another approach to solving the beforementioned challenge is the [Discovery and Action Dialog \(DAD\)](#) to “Discover, Invent, and Unleash Local Solutions to Chronic Problems.” A remote set-up may result in a heightened bias for action, particularly on the side of stakeholders and the management due to a perceived loss of control, impeding a Scrum Team on its way to achieve its Sprint Goal. DAD can support the Development Team to discover outside intervention and how to address the situation by identifying positive deviant (PD) behaviors and practices to provide solutions to real (and perceived) problems. Technically, a virtual DAD applies standard procedures like breakout rooms, embedded 1-2-4-All, joined workspaces, and probably a Shift & Share.

**Learn more about remote Liberating Structures:** [Remote Agile \(Part 2\): Virtual Liberating Structures](#).

# The Remote Agile Guide



## Supporting Liberating Structures

Two Liberating Structures microstructures can support a remote Daily Scrum and particularly its second phase: [1-2-4-All](#), and [Lean Coffee](#):

- To cover **1-2-4-All**, we need breakout rooms and a place to aggregate the findings. We start with everyone in the whole group for a minute in silence; then, we split the whole group into pairs using Zoom's breakroom feature for 2 minutes. After that round, we merge two pairs into a group of four for five minutes—this has to be done manually by the host—and the group aggregates its findings, for example, on a Google sheet prepared for each group in advance. We can introduce each group's findings to the whole group by screen sharing in a Shift & Share.
- **Lean Coffee** is an excellent example of a workaround for virtual Liberating Structures. Gather all the input in the usual way, for example, engaging in 1-2-4-All, and gather those on a FunRetro.io board while voting is turned off. (Use several columns if the whole group is large to speed up the gathering process.) Then ask the whole group to cluster similar topics, then turn on the voting and order the remaining entries by votes. For here, you continue with a whole group discussion, or you engage smaller groups with breakout rooms.

## Remote Daily Scrum Antipatterns

There are plenty of [Daily Scrum anti-patterns](#) in general. However, I want to point at a few anti-patterns that are particularly relevant for a remote Daily Scrum:

- **Orientation lost:** The Daily Scrum serves one purpose as it answers a simple question: Are we still on track to meet the Sprint Goal? Or do we need to adapt the plan or the Sprint Backlog or both? Often, the Development Team cannot answer that question immediately. (In that respect, tracking the progress towards the Sprint Goal by regularly updating Scrum boards or ticket systems is even more recommended for a distributed Scrum team.)
- **Cluelessness:** Team members do not prepare themselves for the Daily Scrum. ("I was doing some stuff, but I cannot remember what. It was important, though," is not the right attitude for successfully practicing remote Scrum. This kind of work requires becoming more organized by comparison.)
- **No use of work-item age:** A Development team member experiences difficulties in accomplishing an issue over several consecutive days and nobody is offering help. (Often, this result is a sign that people either may not trust each other or do not care for each other. Alternatively, the workload of the Development Team has reached an unproductive level as they no longer can support each other. Note: Of course, the Scrum Guide does not mention the 'work item age.' However, it has proven to be a useful practice.)

# The Remote Agile Guide



- **Status report:** The Daily Scrum turns into a status report meeting, driven by the fear of the management to ‘lose even more control now’ and Development Team members are waiting in line to ‘report’ progress to the Scrum Master, the Product Owner, a stakeholder, or the manager. (No comment needed; that situation is a case for the Scrum Master to step in and support the Development Team.)
- **Planning meeting:** The Development Team hijacks the Daily Scrum to discuss new requirements, to refine user stories, or to have a sort of (Sprint) Planning Meeting.
- **No routine:** The Daily Scrum does not happen at the same time every day. (While routine has the potential to ruin every Sprint Retrospective, it is helpful in the context of the remote Daily Scrum. Think of it as a spontaneous drill: don’t put too much thought into the Daily Scrum, just do it. Skipping them can turn out to be a slippery slope: if you skip one Daily Scrums or two, why not skip four out of five?)

**Read more:** [Daily Scrum Anti-Patterns: 20 Ways to Improve.](#)

## Conclusion

Daily Scrum events — remote or not — are essential to a team’s self-organization and, consequently, its ability to achieve the Sprint Goal. While the first phase, the 15-minute time-box, is relatively straight-forward, the second, the problem-solving phase requires more effort on the side of the moderators, the Scrum Master or Development Team members, in a remote setup. Nevertheless, given the right level of training and support, also the second phase can be fully embraced by a distributed Scrum team.



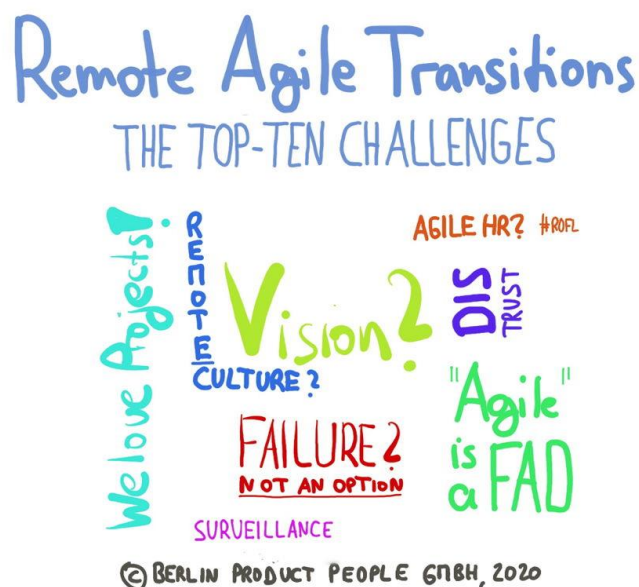
## Remote Agile Transitions

### Remote Agile Transitions — The Top-Ten Challenges

#### Introduction

We are used to saying the Scrum is a perfect probe for organizations, as it will reliably discover all dysfunctionalities. Since the pandemic has forced many of us to work remotely, this unique capability has been kicked into overdrive regarding remote agile transitions.

Here are my top-10 challenges of organizational change that remote Agile has made more urgent to address than ever before.



#### Why Remote Agile Accelerates Change

Three months into working with distributed teams, at least for the majority of us who are not working for one of the remote work pioneers like Automatic, Gitlab, or Buffer, operational issues at a tactical level have been addressed. Zoom has fixed many of the problems reported, we learned how to organize engaging remote events, and more people start embracing techniques like Liberating Structures or Training from the Back of the Room to get all sorts of work done. With a good outcome, as it seems that [remote work improves productivity](#), at least when [team members are engaged with their work](#).

# The Remote Agile Guide



Now that the dust is beginning to settle, I believe it is the right time to switch from survival mode with a focus on keeping operations running back to a more strategic way of handling change to become an agile organization. Here are my top-10 challenges that remote Agile transitions are facing:

1. **A lack of a vision:** Why is the organization pursuing to become agile? What are we fighting for—is it really about learning faster than the competition? If an organization is not transparent about vision and strategy, teams will find it significantly more challenging to become self-organizing and accept responsibility. (Or let us put it in a different way, Alice-in-Wonderland style: “If you don’t know where you are going, any road will get you there.”)
2. **Ignoring the trend:** The organization’s middle management, as well as individuals, do not embrace an agile mindset or abandon it quietly, believing ‘Agile’ is a pandemic-related management fad—like distributed teams and remote work—that will go away sooner or later. (Well, I think that Agile is a massive trend, not a mere fad. And remote work is here to stay, too.)
3. **Clinging to projects:** My budget, my feature: The organization tries to become agile without switching from a project-based to a product-based development approach. The process continues to be ruled by (annual) budgets, and risk-mitigation by committee is still the norm, becoming even more counter-productive during a remote agile transition.
4. **Culture I: No failure culture:** In moments of crisis and uncertainty, organizations and individuals tend to focus on short-term results. If your professional future depends on appearing useful, staying in your comfort zones becomes an incentive as it pays to play safe. However, shouldn’t rapid learning and being innovative now be the name of the game to preserve the organization’s future? (Failure is an inevitable part of that approach. During a remote agile transition, leaving the comfort zone, taking an acceptable risk should be rewarded instead.)
5. **Culture II: The culture is lagging behind developments:** The organization is not doubling down on transferring or preserving its culture in a remote agile environment. My favorite definition of company culture goes as follows: Culture is what happens when you are not looking. Now apply this model to a situation where there are no watercoolers and coffee machines that can act as waterholes for the tribe, and where having lunch is not an option. (Going remote requires more effort to cultivate your organization’s culture, particularly at the leadership level.)
6. **Distrust meets a bias for action:** A perceived loss of control at the management level accompanied by a bias for action both lead to doubling down on old-school Tayloristic micro-management during a remote agile transition. The management abandons self-organization the moment it is needed the most due to the complexity of the situation and resulting uncertainty. It also still believes in its traditional role: telling people what to do, how to do things, and making sure everyone is busy at all times. (A thoughtful dealing



# The Remote Agile Guide



with the new situation requires investing more in trusting people, their willingness to accept responsibility voluntarily, and self-organization.)

7. **Surveillance:** Trust won't be built by surveilling and micro-managing team members. Therefore, don't go rogue; the prime directive rules more than ever in a remote agile set-up. [Trust in people and do not spy on them](#) — no matter how tempting it might be. (Read more about the damaging effect of a [downward spiraling trust dynamic from Esther Derby](#).)
8. **A slow convoy of organizational silos:** The organization moves at the speed of the slowest to change functional silo. Pre-pandemic, it has not pursued efforts to establish a rapid build-test-learn culture, and thus departments are moving at different speed levels. (In a remote Agile world, the resulting friction is steadily increasing, probably to a breaking point.)
9. **Team-building issues and agile HR:** In a market where skilled people are made redundant, the people folks move back to defending the organization from the individual worker instead of focusing on supporting the transition to a team of teams by actively involving team members in solving personnel issues. (There are plenty of dark patterns of this kind, for example, moving people among teams upon short notice. Or, not staffing teams adequately, for example, leaving Scrum Master positions open or make working Scrum Masters redundant. Open positions are not filled, and teams are kept too small and hence not cross-functional and, therefore, not self-organizing. Or they do not involve teams in the recruiting process itself.)
10. **No training and tech is your problem:** The organization is not actively training team members in the technology, software, and practices that the distributed team relies on. To make matters worse, the organization does not provide support with technology or connectivity in the home-office. (Don't assume that people know how to use the technology and software involved or have an adequate Internet connection. Therefore, offer regular training classes to signal that not knowing how remote agile collaboration works can be resolved and provide any support necessary to get home-offices up to the standards. Failing in this respect at an organizational level qualifies for the category of penny-wise and pound-foolish.)

## Conclusions

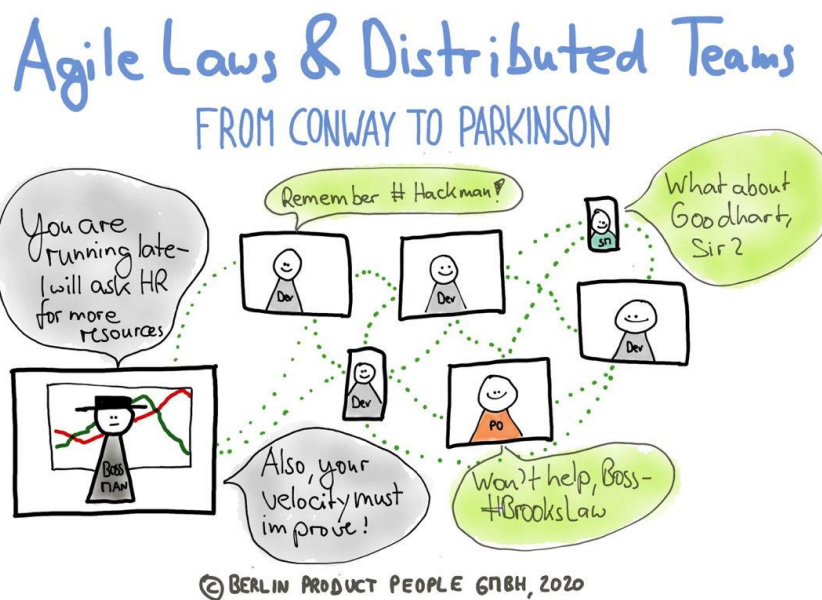
The added complexity, the increased uncertainty caused by this pandemic make working with cross-functional, self-organized teams in a decentralized fashion — collaborating as a team of teams — more valuable than ever. Now is the time to double down on becoming agile to reap the benefits of business agility: overtaking the competition while creating value for our customers; don't regress to the practices from the 1920ies if you want your remote agile transition to be successful.



## Agile Laws & Distributed Teams: From Conway to Goodhart to Parkinson

### Introduction

On many occasions in the recent past, working with distributed agile teams has amplified existing organizational, technical, and cultural challenges in many organizations. Starting changing, and I am not referring to the introduction of a new video conferencing tool, always requires the acceptance that there is a problem that needs attention. In that respect, the current issues that many distributed teams face may also act as accelerants to become more agile. The following article addresses some of the most current impediments to achieving agility by revisiting several agile laws that are particularly relevant to distributed agile teams.



### Agile Laws: Conway, Brooks, Hackman, Goodhart, Larman, and Parkinson

From the long list of observation, heuristics, and mental models in psychology, organizational design, or software engineering, I pick six “agile laws” that seem to be particularly relevant in this area of distributed agile teams:

#### Conway's Law

Mel Conway first postulated his thesis in a paper from 1968, stating that:

# The Remote Agile Guide



*“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization’s communication structure.” ([Source](#).)*

In other words: If two teams are building a part of an application separately, that system will probably have two components, introducing dependencies and additional communication overhead.

That has always been a challenge. When teams are co-located, at least you can negotiate issues informally over a coffee or the watercooler. With distributed teams, this approach has become less of an opportunity, given the additional communication overhead and the inherent formality to organize yet another a Zoom meeting.

One possible way to address the issue is the inverse Conway maneuver: “[...you may want to begin by breaking down silos that constrain the team’s ability to collaborate effectively.](#)” (See also: Torbjörn Gyllebring: [The Reverse Conway — Organizational Hacking for Techies](#).)

The idea has been around for several years: design the teams according to the product requirements and give them autonomy to create the best possible solution from both a value proposition and an organizational sustainability perspective.

(Free paper on Conway from HBS: [Exploring the Duality between Product and Organizational Architectures: A Test of the “Mirroring” Hypothesis](#).)

## **Brooks’s Law**

Frederick Brooks stated in his 1975 book [The Mythical Man-Month: Essays on Software Engineering](#) that “adding manpower to a late software project makes it later.”

The challenge that we face now is that the productivity of newly distributed teams is likely to suffer, see “[Conflicting Reports on Remote Worker Productivity and Contentment](#).”

The typical reaction of the middle management, driven by a bias for action to show initiative, fight the crisis, and overcome a perceived loss of control, is probably to assign more people to a problematic project, instead of allowing the teams to adjust and entrust them with more autonomy.

## **Hackman’s Law**

Adding more people to a team or project to accelerate delivery also contradicts another agile law, Hackman’s law: “[The larger a group, the more process problems members encounter in carrying out their collective work](#) [...] worse, the vulnerability of a group to such difficulties increases sharply as size increases.”

In a remote working situation, to make matters worse, there is a compound effect due to the increased communication overhead. Hence, an appropriate strategy to counter this effect would be employing small, agile teams and an organization designed as a team of teams, aligned yet autonomous.

# The Remote Agile Guide



## Goodhart's Law

Back in 1975, the British economist Charles Goodhart first published the idea that would carry his name, when he wrote about monetary policy. The anthropologist Marilyn Strathern later summarized it as: “[When a measure becomes a target, it ceases to be a good measure.](#)” (Doc Norton: “[And the target therefore no longer means what you think it does.](#)”)

Applying this to distributed agile teams, we need to come back to the middle management and its real or perceived pressures in the crisis, the organization are imposing. A perceived loss of control due to remote and often asynchronous communication, and the urge to ensure being visible in communication artifacts may result in a tendency to run a tighter ship: more reports, more metrics, and more meetings.

Again, reversing course in this manner in the middle of a massive, complex change — imposed from the outside — with an uncertain outcome is the opposite of the appropriate action. Exercising more command & control to counter complexity does not work as any experienced leader will note. (Eli Goldratt: “[Tell me how you measure me and I will tell you how I will behave.](#) If you measure me in an illogical way [...] do not complain about illogical behavior.”)

The alternative is creating room for autonomy and putting trust into people: “[Don't tell people how to do things, tell them what to do and let them surprise you with their results.](#)” (Curtis Carlson: “[In a world where so many people now have access to education and cheap tools of innovation.](#) Innovation that happens from the bottom up tends to be chaotic but smart. Innovation that happens from the top down tends to be orderly but dumb.”)

## Larman's Laws

You might now ask how come that organizations so often fail to create organizational structures that are flexible, yet resilient. Craig Larman formulated a reason for that:

*“Organizations are implicitly optimized to avoid changing the status quo middle- and first-level manager and “specialist” positions & power structures.”* ([Source.](#))

This observation reflects the systems-thinking approach to change: If you want the people's behavior to change, the system needs to change first. Attempting to change the culture of an organization without changing the underlying system will fail. The currently imposed need to change to respond to the crisis will thus have to target the system itself, not merely the operational or tactical procedures.

## Parkinson's Law

The reason why time-boxing is such a valued practice among agile teams is simple: “Work expands so as to fill the time available for its completion.” ([Parkinson's Law.](#))

# The Remote Agile Guide



When trying to create valuable, sustainable, and profitable products in complex environments, a rapid feedback loop is essential: Build, measure, learn. Waiting too long before shipping, or pursuing perfection, is not an option. Instead, Sprints, cycles, iterations as well as inspection and adaptation is the name of the game. We aim at iterating fast enough to keep in sync with the market, yet avoid too much overhead with Sprints that are too short.

The issue with distributed agile teams is that the routine of shipping sometimes tends to be valued higher than the learning part of the equation. Admittedly, “learning” is more difficult in a remote setting, but not impossible. But focusing on the shipping part to comfort our bias for action to tackle uncertainty, we are moving closer to becoming a feature factory—which is the opposite of creating a team of autonomous teams, solving problems on behalf of our customers.

## *Conclusion*

Working with distributed agile teams has amplified existing organizational, technical, and cultural challenges in many organizations. In that respect, revisiting ‘agile laws’ has proven helpful in addressing those impediments. Probably, you may even be able to use these issues to your advantage. As the saying goes: “Every problem is an opportunity.”

# The Remote Agile Guide



## About the Author



Stefan is a [Professional Scrum Trainer with Scrum.org](#), an Agile Coach, and Scrum Master.



He is specializing in coaching agile practices for change, for example, agile software development with Scrum, LeSS, Kanban, and Lean Startup, as well as product management.

He also serves as one of the XSCALE Alliance stewards and coaches organizations in business agility. Additionally, he is a licensed facilitator of the Agile Fluency™ Team Diagnostic.

He has served in senior leadership positions several times throughout his career. His agile coaching expertise focuses on scaling product delivery organizations of fast-growing, venture-capital funded startups, and transitioning existing product teams in established enterprise organizations.

Stefan is also curating the popular [‘Food for Agile Thought’ newsletter](#) for the global Agile community with 26,000-plus subscribers. He blogs about his experiences on [Age-of-Product.com](#) and hosts the most significant global Slack community of agile practitioners with more than 7,700 members.

His ebooks on agile topics have been downloaded more than 45,000 times. Lastly, Stefan is the organizer of the [Agile Camp Berlin](#), a Barcamp for 200-plus agile practitioners.

Read more about Stefan at [Scrum.org](#), and connect with him via [LinkedIn](#), or [Twitter](#), or privately via [email](#).