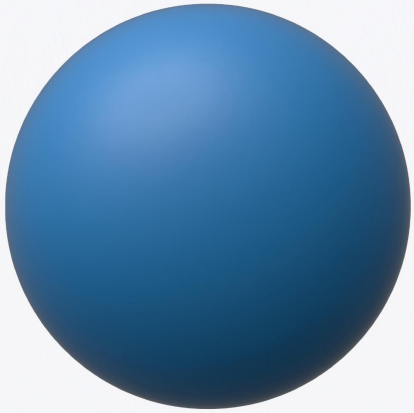
PragmaticBookshelf

simplicity

sustainable, humane, and effective software development



dave thomas

edited by
Susannah Davidson

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit https://www.pragprog.com.

Practice 9

Meetings, Bloody Meetings

At the risk of channeling Donald Rumsfeld, for any particular person, there are things:

1. they know that they know,
2. they know that they don't know and,
3. they don't know that they don't know.

The overall hope when running a meeting is that people in the first group will help enlighten those in groups two and three, and along the way, everyone will learn something.

The reality, though, is that meetings are often about those in groups one and three arguing about their personal realities, and those in group two catching up on their social media.

There are times when a meeting is an appropriate tool. Most of the time, though, the meetings we hold are at best a waste of time and at worst a source of misinformation, confusion, and resentment.

Clearly, there are many levels of meeting: two people discussing an idea might be one end of the spectrum, while an all-hands meeting is at the other. I have nothing at all against two or three people meeting to discuss something (as long as they are aware of the Practice 10, Decorum: If You Have to Have a Meeting, on page ? guidelines). But I have a strong aversion to large meetings, and I really dislike the prevalence of all-hands meetings in the so-called "agile" world.

But Meetings are "Agile"

Every "agile" method highlights the importance of communication within the team. Surely meetings are a great way to ensure that people are all on the same page. Meetings are the forum for leveling knowledge and sharing understanding, no?

I don't think so.

Idea 26 **Meetings are not "agile"**

And maybe, secretly, the big-A agile folks don't really believe it either given the recent trend to relabel meetings as "ceremonies." (Where did I put my ermine robe?)

Meetings are Inefficient

Let's start this section by reminding ourselves that five minutes spent in a meeting with twelve participants represents at least an hour of productive time.

Why "at least?"

- Most meetings start late, and many meetings run over.
- Meetings often wait while a small group has a friendly, off-topic chat.
- Often there's a participant who just won't stop talking, often about things that may be only tangentially relevant.
- Many people won't have prepared, so the rest of the group will wait while they are brought up to speed.
- And my perennial favorite: the projector doesn't want to connect to the presenter's device.

On top of that, the format of most meetings does not encourage efficiency. Typically, one person talks, then people chime in with comments, and then those comments spark other comments, and so on. Then another person talks and the comment cascade starts again. If this were an algorithm, it would be $O(n^2)$.

Idea 27

The negative impact of a meeting extends far beyond the meeting room

Meetings are Unfair

It is all too easy for a meeting to be dominated by the most senior, most knowledgeable, most aggressive, or just least polite person present.

Even if you don't have a verbal bully in the room, shy or junior people often hold back unless the meeting leader explicitly involves them.

The result: the same people espouse the same viewpoints. Knowledge isn't shared; it's imposed.

Meetings are Disruptive

Attending a meeting means stopping whatever you're doing at some appointed time. If you're in the middle of something, you have to stop. If you're in the flow, you have to stop. If you're in the middle of a conversation, you have to stop.

On its own, that can be disruptive. It's also time-wasting. If I know I have a meeting coming up, I'll typically not start something during the period of about 10 minutes before it starts. Sure, I'll try to fill in the time, but I won't be moving forward. Just as bad, after the meeting is over I have to come back and recapture the context of what I was doing—I probably waste 15 to 30 minutes coming back up to speed.

Meetings are Expensive

Let's look at Scrum, not because it is worse than the others but because it is the most common, and its meetings are well documented. In a typical two-week sprint, a team will have the following all-hands meetings:

Meeting	Number	Hours/Per	Total Hours
Sprint Planning	1	4	4
Backlog Refinement	1	2	2
Daily Scrum	10	.25	2.5
Sprint Review	1	2	2
Retrospective	1	3	3
TOTAL			13.5

For one sprint, that's 13½ hours out of the total of 80 hours. If the overhead cost of a developer is $200/hour, and you have a team of ten people, that's $27,000 every two weeks.

Is this time and money well spent?

Perhaps it is. But the only way to know is to measure, and the only way I can see to measure is to change what you do and see what the impact is.

So that's what this section is: a series of things that other teams have successfully adopted to overcome the problems with meetings. Let's start by establishing just why we have meetings.

(Oh, and if you were wondering about the title of this chapter, have a look at this marvelous John Cleese training video.)[1]

1. https://vimeo.com/709207228

Idea 28 **Meetings are inefficient, unfair, disruptive, and expensive**

Why Meet At All?

I asked people *why* they had so many meetings. Ignoring the "because that's what we do" answers, I heard:

- The daily standup gives people a chance to say what they've done, what they plan to do, and what problems they have. The intent is to help the team understand the overall context, and perhaps to give individuals the chance to offer advice on problems someone else is facing.
- Design sessions help to thrash out architecture, design, and implementation issues. Alternatives are discussed, new ideas are injected, and more junior team members learn about new languages, frameworks, and patterns.
- Planning meetings give the team context, helping them understand what work is to be done and what the priorities are. It also gives folks a sense of ownership of the schedule.
- The review meetings are an opportunity for the team to celebrate successes and to understand, mitigate, and perhaps forgive, failures.
- All-hands meetings are also felt to be team building events, where people get to know each other better and possibly trust each other more.

Are there other, more efficient ways, of getting the same benefits?

Developer Without Portfolio

Many of the successful teams I met with had created a new role. Some called it *project lead*, but to my mind that title has other baggage. Some called it *architect*, but again there's baggage. So I'm going to use *Developer Without Portfolio*, or *DWP*.

The DWP's sole responsibility is to keep the team running smoothly and effectively. If the team is an engine, the DWP is the oil.

Idea 29 **Developer Without Portfolio reduces team friction**

To do this, the DWP talks with people. Not by having weekly meetings or daily chats, but by actually engaging. I spoke with a team where the DWP spent a lot of their time pair programming with random team members. On another team, the DWP would carry their laptop into different areas and set up shop, listening in to what was happening and chatting where appropriate. There's no rule here: it's up to the personality of the individual. Whatever mode they choose, though, the outcome should be the same: they should know what everyone is doing, their pain points and misunderstandings, their successes and their frustrations.

Knowing all this, the DWP then works to remove impediments.

Replacing the Daily Standup Meeting

The DWP is constantly building a picture of the team's overall situation, sharing and updating it as they circulate through the team. They make a point of asking people what they're planning to do and what problems they're facing.

When people describe their plans, the DWP can slot that into the overall project context. They may discover that the planned work might not be needed, or they may notice that another team member had written some code that might help.

If the developer is having problems, the DWP will come up with ways to help. For small issues, this might be a simple as directly explaining something to the developer. For bigger things, though, the DWP's job is to put the developer in touch with other team members who can help.

During this process, the DWP is also monitoring progress.

Streamlining the Review Meetings

The DWP will already have a good picture of how a development has gone. They will have seen issues and misunderstandings arise, notice tasks where estimates were off, and seen people succeed or struggle.

So, rather that the typical "so let's go 'round the room" review, the DWP can circulate their observations ahead of the meeting, allowing the team to focus on resolutions.

Streamlining Design Meetings

The DWP is explicitly *not* the team architect, and should have no special authority when it comes to design decisions. However, of all the team's members, the DWP probably has the best overall picture of what is needed. Combined with the DWP's experience, this puts them into a good position to make suggestions and suggest alternatives when it comes to design.

At the same time, design is often best when done collaboratively, which probably calls for meetings. My strong suggestion: limit the attendance to three or four people, and try to follow the decorum guidelines at the end of the chapter.

You Expect Us to Add An Extra Team Member Who Isn't Coding?

I don't expect it, but I hope you'll try it.

If you need help justifying it, add up the developer hours currently spent in meetings. For the team of ten developers we talked about on page 5, that figure comes to about 135 hours per two-week sprint. If you eliminate those meetings by adding a DWP, that is equivalent to adding over one and a half extra developers to the team. Even if you only halve the time spent in meetings, you've freed 0.8 developers. The other advantages of a DWP will easily make up the extra 20%.

Try This

Discuss the concept of a Developer Without Portfolio, first with your team and later with managers. If everyone's on board, run an experiment.

Decide what metrics are important, and make sure you have them available for the three months before the experiment.

During this time, cast around for someone who can take on the DWP role. They should be fairly experienced, enjoy chatting with people, and capable of putting aside their own opinions.

Then take a couple of months migrating to the new way of working, and start collecting the metrics again. Six months later, compare metrics and decide whether to continue.

All I ask is that, alongside all the regular performance metrics, you also include some measures of developer sentiment.

Some Meetings Are Unavoidable

All-hands meetings are definitely appropriate when they address things that genuinely impact the entire team, such as project inception, large project windup, and company announcements. If you have any control over these, try to follow the meeting decorum guidelines that follow.

Other meetings will also be necessary: design sessions, management reporting, subteam coordination, and so on. Keep these small: you can always bring people in if needed. Follow meeting decorum.