Remote versus Co-located Work

There isn't a simple dichotomy of remote versus co-located work, instead there are several patterns of distribution for teams each of which has different trade-offs and effective techniques suitable for them. While it's impossible to determine conclusive evidence, my sense is that most groups are more productive working in a co-located manner. But you can build a more productive team by using a distributed working model, because it gives you access to a wider talent pool.

19 October 2015

Martin Fowler

Find **similar articles** to this by looking at these tags: agile · productivity · team environment · team organization · collaboration

Contents

One of the most profound consequences of the information age is the ability to do so many things while ignoring location. I no longer need to visit most shops, libraries, or travel agents. (I'm looking forward to the day I don't have to visit a dentist.) Most of the world has seen this, but it's particularly obvious to software developers, who are generally at the front of the digital transformation.

Yet when it comes to developing software, many developers do not take advantage of the communication possibilities of connected computers. Yahoo got a lot of press when it recently brought all its off-site workers back to a single site. Leading tech companies like Netflix and Google strongly prefer having their staff in a single site.

Such moves lead others in our profession to point and laugh. Some of the loudest are startups such as Etsy, Basecamp, and Github, many of whose employees have never worked in an office together. For such teams, remote work is the future, those who push against it are on the losing side of history.

While I've been involved in discussions about remote working many times in my years in the industry, I don't feel there is much of a conclusive set of factors I can talk about. The evidence of effects of remote working on software development resist being gathered together in any meaningful way.

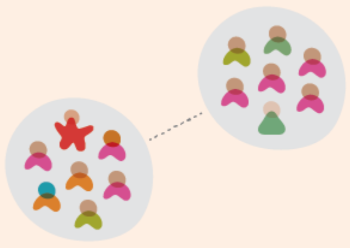
That said, I do talk to a lot of teams, and those conversations have led me to some tentative opinions that I'll share here.

The Many Shades of Remoteness

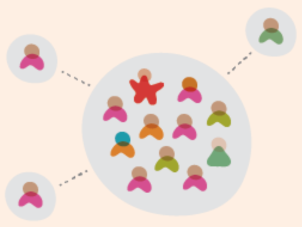
The first thing to get straight is that there isn't a simple dichotomy between co-located and remote teams. There's many different varieties, each with their own strengths and weaknesses. To make it easier, here are a few didactic landmarks.

A **single-site** team, is a team where everyone is co-located in the same physical location. Ideally this means you're all within a few steps of each other, able to quickly collaborate without having to arrange anything, and easily able to see what everyone else is up to. Many teams like a single team room to do this, as it maximizes the ease of communication. Even the presence of cubicles gets in the way - many agile coaches have stories involving screwdrivers.

A **multi-site** team has two or more co-located groups at separate locations within a larger team, perhaps with some formal sub-team boundaries and responsibilities. A good example of this is a development team split between Melbourne and Xi'an.



Satellite workers occur when you have most of the team co-located, but a few members working remotely, either from home or in another office.



A **remote-first** team is one where everyone works in a separate location, usually from home, and thus all communication occurs online. Most open-source projects are remote-first, and this experience has encouraged many startups to use it.

Remoteness varies by degree. It's often been observed that just splitting a team across two floors of the same building is enough to break the feel of co-location. Adding more distance and time-zones increases that remoteness, but many argue that the biggest difference occurs when you're no longer a short walk away from your collaborators. The key point is the point where you find it easier to send an email than walk over to talk.

Most people are more productive when co-located

As with so many topics in software development, arguments about process don't terminate because we cannot measure our output. I can't take 100 software development teams and analyze whether remoteness affects productivity in any quantitative way. People make anecdotal statements such as "I feel more productive in a co-located team" but that's not great evidence. But despite that it's not great, it's the only evidence we have.

Another factor is that there are so many other factors that make a team work well. If someone is saying they are more effective on a single-site team, that may be because other factors are in play compared to different teams. One way to reduce this problem is to pay special attention to teams that have changed their distribution pattern, such as splitting from single-site to multi-site. Other factors still intrude, particularly since changes in team distribution often mean people will leave or join a team, but I think this yields stronger evidence than comparing totally different teams.

Given this, all I (or anyone) can do is listen to lots of people and make the best judgement I can. I've heard a lot of experiences about teams and locations, including quite a few where teams have changed their distribution pattern (although I haven't heard very much to or from remote-first). The weight of anecdotes leads me to conclude that most teams are more productive when in a single-site model.

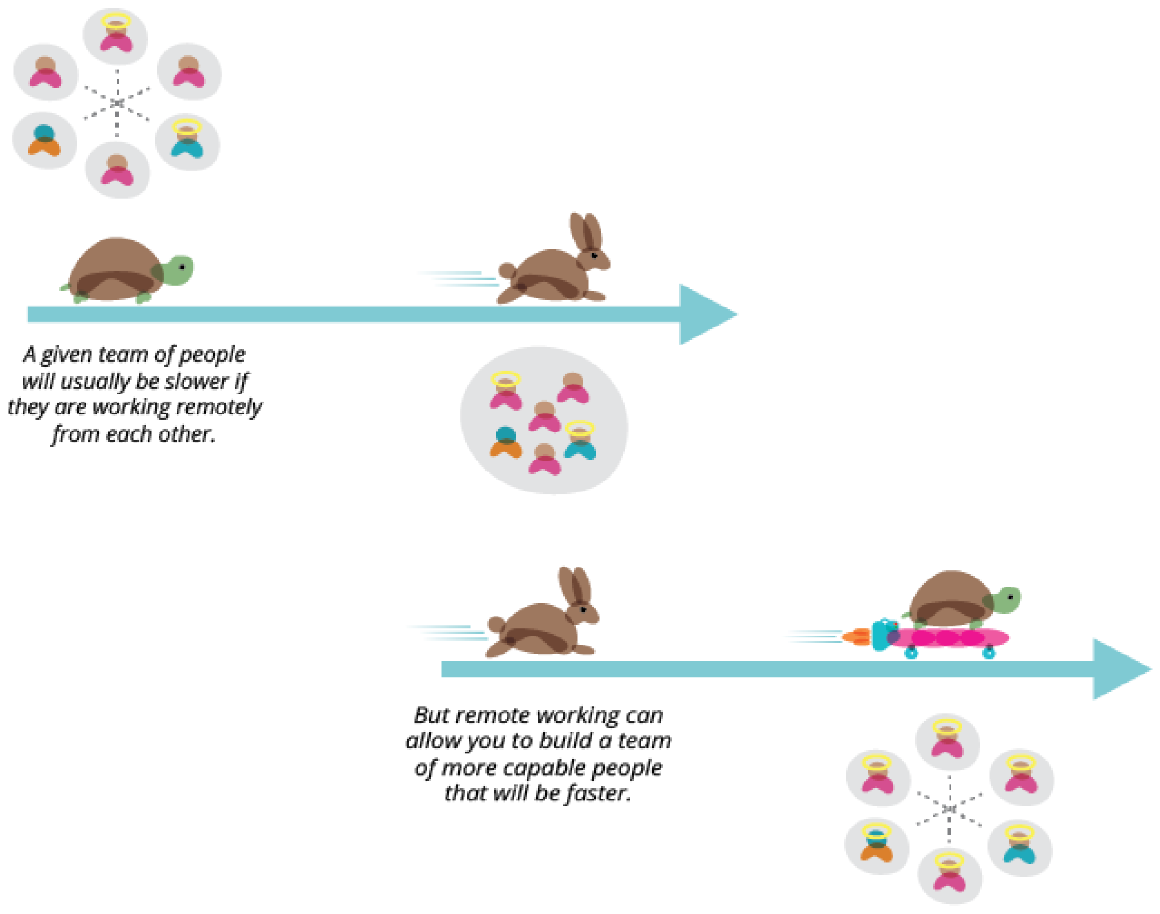
The reason for this is the ease of communication. While tools like (video) chat, screen sharing, and the like have done much to make remote work easier, there is still nothing as effective as being able to turn around, see the person you want to talk to, and just be able to speak. Co-location also introduces a huge amount of out-of-band conversations which improves personal relationships. The result is a virtuous cycle of improved relationships and communication. Since communication is such a central part of software development, this is a big impact on productivity.

But notice I said *most*. Human beings vary enormously, although one common feature seems to be a human tendency to think everyone acts the same. So I can easily believe that some people are more effective when working remotely. My sense is that this is a minority of people. (There may also be a generational factor here as younger people are more used to remote interaction.)

Remote teams are often more productive

So if we discount the remote-oriented minority, does the greater productivity of a single-site team mean we should favor the single-site model? As it turns out we often shouldn't.

Although I widely hear that a given team is more effective when co-located, a single-site puts a big constraint on who you can have in your team. Such a rule means you aren't able to hire the best person for the job, you can merely hire the best person who is prepared to relocate. By making a team remote, you can widen your range of people you can bring to the team. A remote team may be less productive than that same team if it were co-located, but may still be more productive than the best co-located team you can form.


A given team of people will usually be slower if they are working remotely from each other.
But remote working can allow you to build a team of more capable people that will be faster.

Not just does remoteness avoid questions of permanent relocation, it also provides a lot more options to individuals, particularly when working from home in a remote-first model. People value the fact that it's easy to pick up children from school, avoiding the waste of time and energy in a commute, and the pleasant environment. Offering that makes an employment package more alluring. Given that women often take on care-giving work that makes it harder to spend time in an office, it may also help improve diversity.

This effect is also a big factor across countries. As offshoring became popular, most people saw it as a way to reduce costs. At ThoughtWorks, we've seen it as more important in finding the best talent. For example our China offices have become particularly valuable supporting work in Australia because the size of the talent pool is so much bigger.

Pay attention to communication patterns

The way people communicate is central to effective software development. By introducing remote working, in any of its forms, you're introducing constraints on communication patterns. In particular we have to be conscious that co-located communication is much richer than online communications - at least for most people. Hence most people that are co-located will communicate better and have better personal relationships than occur between remote workers. This leads to a range of consequences that you need to be aware of.

Multi-site teams have a tendency to form an us-and-them attitude to the other sites. You can reduce this by using regular contact visits and ambassadors. Contact visits are short cross team visits. While these are good for some occasional deeper collaborations, often their best purpose is to build the human relationships. Organizations tend to forget the importance of the latter. So when doing contact visits, put more emphasis into social bonding (which means allocate time for activities that help relationship-building). Ambassadors are people who spend a few months at a different site. An ambassador can do a great deal to facilitate communication between their temporary remote team and their usual home team, both when remote and when they return home.

If you're going with a remote-first model, you need to go all-out with it. All communication should occur online, don't have co-located sub-groups in the same office. I've even heard of some teams forcing people working in the same office to working in solitary offices and mandating that any communication with the programmer next door occurs online.

With a multi-site team, divide the work by fully autonomous components. Each team should be full-stack and responsible for taking a component from idea right through to production. Don't divide by layer (frontend/backend/data) nor by activity (analysis/development/testing). Both layer and activity boundaries have rich communications across them. Remember the central importance of Conway's Law.

It's very difficult to get satellite workers to be effective. With most people co-located, most communication will happen within the co-located team. I hardly ever hear of this model without the satellite person getting increasingly detached. If their work is very autonomous, that will reduce the problem. It is also wise to ensure satellite people make regular visits to the onsite team, at least a couple of times a month. But in most cases it seems best as a temporary measure.

A particular area where the difficulties of remote communication kick in is mentoring junior staff. Some advocates of remote-first working argue that you should only take on experienced staff to remote-first teams. Like much of this, it's not impossible to mentor people remotely, but it is much harder. For multi-site teams, ensure each site has experienced mentors to guide new people. Avoid letting junior people be satellite workers. Be wary about having junior people join remote-first teams, certainly don't try it until the remote-first team is working smoothly and then only add junior people very slowly.

Remoteness and Agile

I've heard a few people argue that agile software development is incompatible with remote working. That's poppycock, or at least it is according to my understanding of agile thinking.

Certainly agile methods have encouraged a greater degree of co-location. Extreme Programming includes the practice “sit together” as one its primary practices: *“the more face time you have, the more humane and productive the project”*. The agile manifesto says *“The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”*

But all of this is just making the point that a given team usually collaborates better when co-located. It doesn't make any argument about getting a better team by supporting a remote working pattern. The first value of the agile manifesto is “Individuals and Interactions over Process and Tools”, which we should read as encouraging us to prioritize getting the best people we can on the team and helping them work well together. (Kent points out that “Sit Together” isn’t mandatory for XP.) While we recognize face-to-face communication is more effective, that recognition cannot act to override the importance of individuals and interactions.

Conclusions

As I hope is now obvious, there's not enough good evidence to form any strong conclusions about the efficacy of remote working. But based on these shaky foundations, these are my key thoughts:

- Never forget there are different distribution patterns for teams, not just a simple remote versus co-located dichotomy. The advantages, disadvantages, and effective techniques for multi-site teams will often differ from remote-first work.
- Most groups of people will be more effective when working co-located due to the richer communications they have. But don't let that make you forget that some people seem be more effective when in a remote-first model.
- Despite the fact that I think most teams would be more productive working co-located, you will often get a more effective team by embracing some form of distributed model because it will widen the talent pool of people you can get.
- When using a remote working pattern, pay attention to how the communication patterns form. Invest in improving communication, including travel and technology.

The fact that you can get a better team by supporting a remote working pattern has become increasingly important during my time in the software business and I expect its importance to keep growing. I sense a growing reluctance amongst the best developers to accept the location and commuting disadvantages of single-site work. This is increasingly true as people get more experienced, and thus more valuable. You can either try to ignore this and accept the best people who will relocate for you, or you can explore how to make remote working patterns more effective. I think that organizations that are able to make remote working patterns effective will have a significant and growing competitive advantage.

For articles on similar topics…

…take a look at the following tags:

agile productivity team environment team organization collaboration

Acknowledgements

Bill Kimmel, Fábio Santos, Gayatri Kalyanaraman, Hugo Corbucci, Jie Xiong, Ken McCormack, Kevin Yeung, Kyle Hodgson, Pete Hodgson, Peter Gillard-Moss, Rouan Wilsenach, Sriram Narayan, Tiago Griffo, and Valerie Roske commented on drafts of this article. Pete Staples helped me with the illustrations.

Further Reading

Remote working is a common issue that comes up in web articles and blogs and I haven't attempted to pull together a suggested reading list. I will, however, suggest an article I wrote a decade ago about using agile methods for offshore work, based on our experiences in India. The advice and conclusions in the article are still valid for multi-site teams, although there are a couple of technological tweaks I keep intending to add.

Significant Revisions

19 October 2015: First published