

# Remote versus Co-located Work

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

February 2015



Martin Fowler

## CONTENTS

[The Many Shades of Remoteness](#)

[Most people are more productive when co-located](#)

[Remote teams are often more productive](#)

[Pay attention to communication patterns](#)

[Remoteness and Agile](#)

[Conclusions](#)

[ACTIVITY](#)

[ENVIRONMENT](#)

[ORGANIZATION](#)

[COLLABORATION](#)

[REFERENCES](#)

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 restaurants. (I'm looking forward to the day I don't have to visit a dentist.) Most of the

as seen this, but it's particularly obvious to software developers, who are y at the front of the digital transformation.

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

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

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

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

## Many Shades of Remoteness

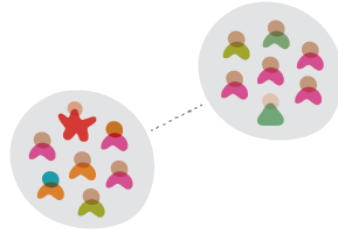
: thing to get straight is that there isn't a simple dichotomy between co-located ote teams. There's many different varieties, each with their own strengths and sses. To make it easier, here are a few didactic landmarks.

**Single-site** team, is a team where

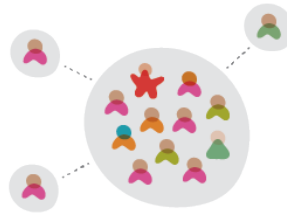
ryone is co-located in the same sical location. Ideally this means you're vithin a few steps of each other, able to kly collaborate without having to nge anything, and easily able to see it everyone else is up to. Many teams a single team room to do this, as it imizes the ease of communication. n the presence of cubicles gets in the ' - many agile coaches have stories olving screwdrivers.



**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 London and Xi'an.



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



**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 floors of the same building is enough to break the feel of co-location. Adding more floors 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. A tipping point is the point where you find it easier to send an email than walk over to

## people are more productive when co-located

so many topics in software development, arguments about process don't te because we cannot measure our output. I can't take 100 software ment teams and analyze whether remoteness affects productivity in any itive way. People make anecdotal statements such as "I feel more productive in ated team" but that's not great evidence. But despite that it's not great, it's the dence we have.

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

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

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

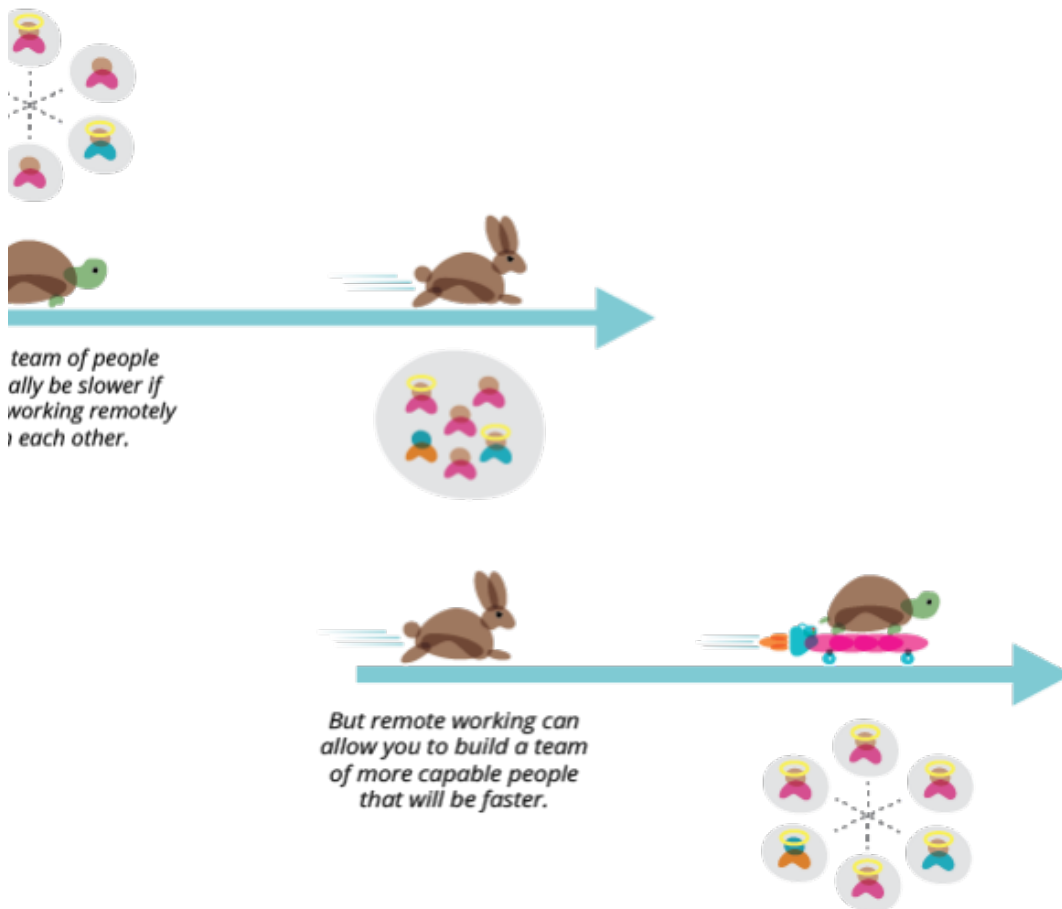
ce I said *most*. Human beings vary enormously, although one common feature o be a human tendency to think everyone acts the same. So I can easily believe ne people are more effective when working remotely. My sense is that this is a / of people. (There may also be a generational factor here as younger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

## ite teams are often more productive

discount the remote-oriented minority, does the greater productivity of a

ite team mean we should favor the single-site model? As it turns out we often 't.

h I widely hear that a given team is more effective when co-located, a single- s a big constraint on who you can have in your team. Such a rule means you ble to hire the best person for the job, you can merely hire the best person who red to relocate. By making a team remote, you can widen your range of people bring to the team. A remote team may be less productive than that same team e co-located, but may still be more productive than the best co-located team form.



does remoteness avoid questions of permanent relocation, it also provides a e options to individuals, particularly when working from home in a remote-first eople value the fact that it's easy to pick up children from school, avoiding the e time and energy in a commute, and the pleasant environment. Offering that n employment package more alluring. Given that women often take on care-ork that makes it harder to spend time in an office, it may also help improve /.

ect is also a big factor across countries. As offshoring became popular, most 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 early valuable supporting work in Australia because the size of the talent pool is much bigger.

## Attention to communication patterns

How people communicate is central to effective software development. By doing 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. Most people that are co-located will communicate better and have better working relationships than occur between remote workers. This leads to a range of differences that you need to be aware of.

Some teams have a tendency to form an us-and-them attitude to the other sites. To reduce this by using regular contact visits and ambassadors. Contact visits are more useful than occasional team visits. While these are good for some occasional deeper discussions, often their best purpose is to build the human relationships. Teams often tend to forget the importance of the latter. So when doing contact visits, put the emphasis into social bonding (which means allocate time for activities that are 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 a temporary remote team and their usual home team, both when remote and when they return home.

When 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 work in solitary offices and mandating that any communication with the person next door occurs online. Despite this, remote-first doesn't mean remote-first organization commonly do in-person meetups every few months to deal with tricky problems that benefit from co-location and to improve human relationships. (For example, Basecamp does one-week meetups, twice a year.)

In a multi-site team, divide the work by fully autonomous components. Each team is responsible 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 (design/development/testing). Both layer and activity boundaries have rich

communications across them. Remember the central importance of Conway's Law.

It's difficult to get satellite workers to be effective. With most people co-located, communication will happen within the co-located team. I hardly ever hear of this 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.

One particular area where the difficulties of remote communication kick in is mentoring and onboarding. Some advocates of remote-first working argue that you should only take experienced staff to remote-first teams. Like much of this, it's not impossible to do with 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 to join the remote-first team is working smoothly and then only add junior people very gradually.

## Agility and Agile

And 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 working.

Many agile methods have encouraged a greater degree of co-location. Extreme Programming includes the practice “sit together” as one of its primary practices: “*the more 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 amongst a development team is face-to-face conversation.*”

But 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 adopting 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.) We recognize face-to-face communication is more effective, that recognition should not act to override the importance of individuals and interactions.

## Conclusions

It 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 conclusions, these are my key thoughts:

Don't 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.

Some 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 teams seem to 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.

Realizing 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 this reluctance to keep growing. I sense a growing reluctance amongst the best talent 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 don't ever try to ignore this and accept the best people who will relocate for you, or rather 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.



## Acknowledgements

I'd like to thank Daniel, Fábio Santos, Gayatri Kalyanaraman, Hugo Corbucci, Jie Xiong, Ken McCormack, Kevin Yeung, Jason, Pete Hodgson, Peter Gillard-Moss, Rouan Wilsenach, Sriram Narayan, Tiago Griffio, and Valerie for their comments on drafts of this article. Pete Staples helped me with the illustrations.

## Further Reading

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



or offshore work, based on our experiences in India. The advice and conclusions in the article are for multi-site teams, although there are a couple of technological tweaks I keep intending to add.

man-Benson has led teams in both co-located and distributed contexts. James Shore did an job of capturing the lessons he learned from this. I'd particularly highlight his views that remote-als don't work well for large teams and are less effective with junior developers.

## ificant Revisions

oughtworks

Fowler | Privacy Policy | Disclosures

