Think globally, code locally: the secret to remote teams

Distributed teams and remote offices aren't going away. But can they be part of a thriving agile culture? We think so.

BY DAN RADIGAN

Agile development was originally imagined for clustered teams, or teams physically located together in the same office. In keeping with the idea that "the most efficient and effective method of conveying information to and within a development team is face-to-face conversation", early agile teams were meant to work together in close proximity.

But today most businesses have a few–or several–distributed teams. This isn't just a trend; it makes good sense. Distributed teams can work on projects around the clock, and strong talent can be found in less competitive markets. (Not to mention, talent is easily retained by not requiring an undesired relocation.) But the benefits of distributed teams aren't without some trade-offs. For many distributed teams, it's difficult to adopt the agile practice of face-to-face interactions.

Other challenges that arise for distributed software teams:

- Coordinating across time zones
- Building rapport when everyone is not in the same office
- Collaborating among different development cultures
- Scheduling meetings or informal conversations when both teams are online at the same time for only a few hours (or less)

These are real problems. But not un-solvable ones. Let's walk through some strategies to help bridge the distance gap between local and remote offices, and ideas to help mitigate other potential issues as well.

How to structure global teams

Good software architecture dictates modular design, so structure your teams the same way. Every office should be self-sufficient in developing a single piece of technology, which minimizes the amount of collaboration required with teams in other time zones and makes them generally autonomous. When a project does require teams in different locations to pitch in, they can focus on their integration points and APIs.

Code reviews also play an important role. Since people are online at different times, distributing knowledge of the code between offices makes support and maintenance much easier. If a production issue emerges when the team is not online, another office can easily step in to support and resolve the issue, thanks to the know-how they gained from cross-team or cross-location code reviews.

Building rapport

It's important in any program, especially agile programs, to have solid rapport across the team. Personal connection builds trust, minimizes missed expectations, eases self-organization, and boosts morale. Within your office, take time getting to know everyone on your team. And, as much as possible, do the same with the people you work with in remote offices. Personal connections are important. The stronger they become, the greater the chance of seeing these colleagues as any other, rather than distant coworkers from unfamiliar places without good relationships.

> PRO TIP:
>
> **At Atlassian, each new employee posts a "intro blog" on our internal Confluence instance, Atlassian's content collaboration tool. The blog introduces the new hire professionally as well as personally (hobbies, interests, family, etc.) which really helps bridge the gap between offices. The more we know each other as people, the stronger we are working together as teams.**

Above all, nothing replaces meeting face to face. Team members in each office will benefit from regular face time, and that includes video conferencing as well as visits to remote offices.

Video conferencing tools like Zoom help bridge the gap between teams, especially for distributed agile teams. However, teams that rely on Zoom should be aware of certain limitations.

- Video conferencing often allows for a very short window of communication, while working in the same office gives significant visibility into another's world: challenges, successes, and opportunities.
- Zoom has done well to address network hiccups. Still, there may be times when network issues occur between offices that can make video and audio choppy or difficult to understand.
- Most people still think of Zoom video conferencing as scheduled time. Creating a culture of using video chat for spontaneous casual conversation takes time. Also, use instant messaging tools like Slack for quick questions.

questions.

To help mitigate some video conferencing issues, encourage team members to have weekly 1:1 video chat sessions. These can be less formal, and help facilitate knowledge sharing in a casual way. Teammates can use these opportunities to build rapport and work better together.

Remember, tone, voice, and posture play a significant part in communication. In-person face time helps the team know their remote colleagues in higher fidelity, which, in turn, makes future video conferencing more effective.

Whether it's a house or a product, you need to define the vision and outline the strategic themes. Think of themes as organization-wide focus areas. What do you want to focus on over the next quarter, 6-months, year? Where do you want to spend time and resources? Performance, user experience, security, new competitive features (hot tub anyone?), or a combination of all these?

> HOW WE DO IT:
>
> **Secondments are temporary assignments in a new job role or location, ranging anywhere from a few weeks to a year. They're not only an effective way to build rapport and spread culture across the team, but it is also a great way for employees to experience a different culture.**

Build a united development culture

There are four simple ways teams can make working across geographies easier and share a common developer culture:

- Overcommunicate decisions across all geographies
- Minimize the friction in setting up the development environment
- Clearly define the definition of done
- Create guidelines for filing effective bug reports

Let's break that down.

First, when moving from a co-located office to a distributed culture, communication becomes significantly harder. The first challenge is training the team to understand that, when decisions are made, they need to be communicated. This may sound obvious, but it's easy to forget! Oftentimes important decisions are made in hallway conversations, informal local team meetings, or by individuals. Plus, it can be easy to dismiss small decisions as unimportant.

> When transitioning to a distributed culture, err on the side of over-

> communicating.

Communicate even minute details until both offices find a healthy groove.

When decisions are made, everyone in each office needs to understand the decision and ideally why it was made. Don't use email. It's too easy to lose important information. Use a content management system like a wiki where team members can easily browse for updates across the team (and get notified of updates via email or Slack group chat tool). You can also use Slack to create channels for individuals and teams to communicate and see updates. Delays caused by team members working on outdated information, hitting a roadblock, and then asking a question costs the team significantly more time than proactively sharing information.

Second, consistent development environments across the team make it easier to work together and track down issues. Spend the time creating a simple "Getting Started" guide and tame first-day friction by automating the setup as much as possible.

Third, when working between offices, clear standards around the definition of "complete" makes it easier to manage expectations and build rapport across teams. A firm definition of complete eliminates ambiguity in the work. For instance, when shipping a release that involves multiple teams, make it clear what it means to be complete: code written, pull request created, code reviewed, tested, and merged into the appropriate branch.

And finally, distributing development means that not everyone is online when problems come up. Having clear guidelines for bug reports and troubleshooting how-tos makes it easier for anyone on the team to track down an issue. Code review and good automated tests also share knowledge about the code base and empowers the affected team to make the fix and validate that the change doesn't have any unexpected side effects. Thus, no team becomes a blocker.

Maximize the golden hours

Every photographer knows "the golden hours" – just before and after sunrise and sunset–is the one of the most effective times to take great landscape photos. The golden hours for distributed software teams are when the local and remote teams are both in their respective offices at the same time. When all teams are in the office, this is a great time for stand-ups.

For teams that share work between time zones, stand-up is a great time to pass the baton so the team just coming online can pick up where the other team left off. And holding stand-up via video conference makes it easy to

ask questions and get up to speed so everyone is off and running as soon as the meeting is done.

Sometimes offices are so far apart that meetings will cause some form of pain for one team. (Get up at 5 a.m. for stand-up with the other team? Umm... no thanks.) Rotate the meeting time so it's a shared burden, rather than continually subjecting the remote team to the odd hours–a sure-fire way to destroy morale. Closely monitor the entire team's engagement at stand-up. If there is undue strain, or the team is not getting a lot out of it, team members will begin to disengage and stop listening or sharing. And stand-up doesn't absolutely have to be a daily meeting. Meet with the remote team a few times a week and use the other days for a local stand-up. Similarly, a stand-up doesn't have to be a morning routine, either. Whatever time of day is most convenient for everyone involved is the best time of day.

Every team is distributed

In a distributed organization, the reality is that every team is remote. All teams need to adapt and learn how to share work between offices, communicate effectively, and grow a consistent culture across geographies. The most effective teams don't just make the remote office conform to the headquarter's culture because they understand that every office can learn something from the others. They seek to find and share successful practices across all locations. They also embrace "we" rather than an "us vs. them" culture.

> Even teams within a single office can benefit by operating like a distributed team.

Because another reality is that they become distributed from time to time. Business travel takes members outside of the office, and working at home occasionally can help employees better manage a work/life balance. Teams that embrace both structure and transparency scale more efficiently. When your project scales beyond your office, the culture will be set up to do the right thing naturally.

DAN RADIGAN

Agile has had a huge impact on me both professionally and personally as I've learned the best experiences are agile, both in code and in life. You'll often find me at the intersection of technology, photography, and motorcycling.

TUTORIAL

Learn kanban with Jira Software

Step-by-step instructions on how to drive a kanban project, prioritize your work, visualize your workflow, and minimize work-in-progress with Jira Software

Try this tutorial →

ARTICLE

How to work with agile specialists

Sometimes skill sets needed for a project fall outside a team's collective abilities. Read best practices for working with an agile specialist.

Read this article →

Agile Topics

- Agile project management
- Scrum
- Kanban
- Design
- Software development
- Product management
- Teams
- Agile at scale
- DevOps

Sign up for more agile articles and tutorials.

Email

email@example.com

Subscribe

Up Next

Working with specialists →