

# Distributed Development

Geographically Distributed Agile Development

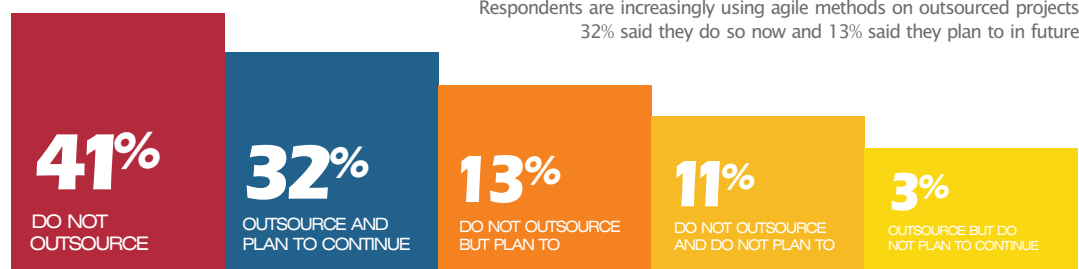
## Introduction

In a 2010 survey on the State of Agile Development<sup>1</sup> 90% of respondents said they worked in an organization that used agile development to some degree. From the same group of respondents 65% said they work with distributed development teams; up from 57% in 2009. Further more, 45% of those surveyed said they either currently use agile methods on outsource projects or plan to use agile on outsourced projects in the future; up from 41% in 2009.

It becomes clear then that distributed development is not only a fact of life for many teams but it is a growing segment of the agile community. This is despite the fact that the very idea of physically distributing teams seems to conflict with agile communication practices. Teams using agile tend to rely heavily on face-to-face communication as a replacement for lengthy and often tedious documentation. Being geographically separated will of course interfere with this form of communication.

## OUTSOURCED DEVELOPMENT PROJECTS

Respondents are increasingly using agile methods on outsourced projects; 32% said they do so now and 13% said they plan to in future.

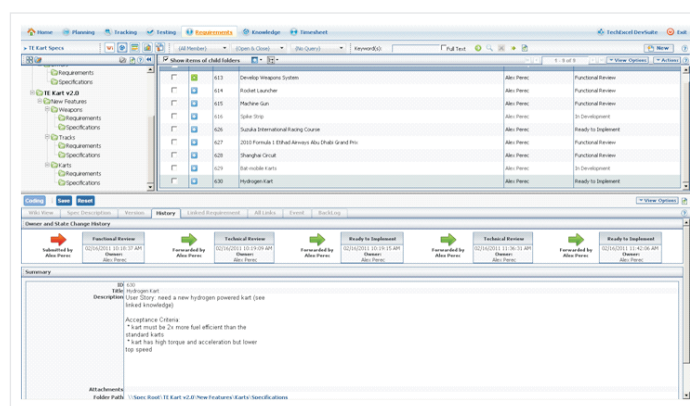


## Risks and Rewards of Team Distribution

So if distributed teams are such an impediment to communication why risk separating teams at all? There are certainly benefits to a distributed team and these of course need to be weighed against the hindrances to communication. The top 3 reasons for distributing are listed below:

- Expanding into global markets may require a local presence in the various locations
- Access to global talent
- Cost reduction through outsourcing

task and eventually tested can be extremely useful in ensuring that miscommunications do not lead to work having to constantly be redone. This is very important since few things will demoralize a team faster than constantly redoing the same work.



Conway's Law states that any organization that designs a system (in this case a piece of software) will produce a design whose structure is a copy of the organization's communication structure.<sup>2</sup> A distributed team will naturally tend to divide up work according to their distribution scheme. The end result of this is an architecture that reflects the distribution or has artifacts that specially accommodate the distribution. Unfortunately this creates knowledge silos and causes each team to think in terms of completing work on their piece of the puzzle rather than focusing on the user stories. The user stories need to remain the focus of development efforts since they are what the customer needs. The customer does not care about orchestrating a design to effectively accommodate a team's geographic distribution.

Effectively distributing agile teams is about minimizing the impact of distribution. The emphasis here is on the minimization. Clearly there is no substitute for teams having face-to-face discussions at a whiteboard or two programmers sitting together working on a piece of code. All we can do is take steps to improve the communication of our distributed teams.

If the decision is made to distribute a team and sacrifice communication we must take steps to make up for it in other ways. The most common way is to improve management of requirements and specifications.

Good requirements and specification management does not take place in spreadsheets. Invest a system that can offer your team a high degree of traceability and speed. Nothing will slow the adoption of a tool faster than a lagging interface. Other nice features to look for in a system are change management and base lining, notes and events, and a robust suite of reports.

Remember, short user stories work because they are supplemented by face-to-face conversations. If these conversations are no longer going to be taking place we need to add more detail to our specifications and task descriptions. Additionally, using a system that can ensure linking and enforce a proper workflow as specifications are turned into development

## Suggestions to Enhance Communication

Since communication is the area that takes the biggest hit when a team is distributed it is also the area we should work the hardest on improving.

- Minimize the overhead of setting up impromptu meetings. Teams should have easy access to a conference phone and projector. Web conferencing that shares the projected screen between teams is extremely useful. Video conferencing with whiteboards on both ends is another great option.
- Hands free headsets, web cams, IM clients, application sharing software and email are all great tools to aid communication. The key here is getting people into the habit of using them.
- The ways teams communicate will probably have to change to support distributed development. Informal verbal communication may need to be replaced by more formal written communication. An example of this would be using email or preferably a work item tracking tool to update status on a particular task, rather than waiting to be asked by another team member. This is particularly useful if a team is distributed across time zones since the information is available at all times.
- Be deliberate about involving remote members. Having them lead meetings from the phone can accomplish this.
- Adjust meeting formats as needed. In order to avoid daily standup meetings running to long and wasting time, use a parking lot system. Issues that come up during the meeting are parked and then parties involved can discuss after the main meeting ends.
- Paired programming can be done using desktop sharing software and should consciously be assigned regardless of geographic location. Frequent code reviews can replace or supplement paired programming as needed. Additionally, code should not be checked in until it has been reviewed and team members should take reviews seriously.
- Whiteboards are invaluable for design meetings. If possible, share whiteboards via webcam, or brainstorm via shared mind map. Pictures can also be taken and shared via a centralized wiki. Ideally a wiki system should be integrated with the tracking tool in use.
- There is no replacement for face-to-face communication. Especially at pivotal points in a project. Key points when temporary collocation should be considered are the first and last few iterations of a project. Also if the project is fairly lengthy, in person meetings at key midpoints can be very useful.
- Make the most of overlapping hours between distributed teams. If your teams are distributed offshore and the time zone distribution is so large that working days no longer overlap then new tactics will be required to allow the different teams to work together. Sub teams can each run standup meetings at the beginning of their respective days, with one sub team staying late or arriving early to accommodate this. Another option is having a designated representative or representatives trusted by both teams that works during the beginning and end of each team's shifts. The reps main purpose will be to poll both teams and communicate decisions that take place during meetings.



# TechExcel White Paper

- Asymmetric distributions can be the hardest to deal with. For example if one person is dialing in while everyone else is in the room they can miss out on a great deal without realizing it. To combat this have everyone dial into the meeting so that everyone is having the same experience. Alternatively have people who would normally be in the conference room dial in occasionally so they remember how difficult it is to be on the phone.



- Assign a coach who is committed to helping teams stick with practices for better communication. These practices are easy to drop since they're typically inconvenient. Remember the coach should not be a practice cop. Their main purpose is to help remind the team of the value of intensive communication even though distribution makes it more difficult.
- Avoid “function focused” remote teams. For example remote testing teams. This can lead to work being pushed back and forth between silos with great inefficiency and put teams at odds. Rather than focusing on delivering a story teams focus on completing their function. Agile teams usually function best with cross-functional members such as developers that code and test.
- If teams must be functional then assign a representative who is present at the beginning and end of each team's day. They will communicate development work completed to QA and vice versa at the beginning and end of each team's respective days.

- Do not distribute development work by location. Doing this will have negative effects over time. The architecture will begin to reflect the team's geographical distribution becoming increasingly segregated (according to Conway's Law<sup>④</sup>). Teams should think about their work as completing user stories not adding features to components. Tasks related to a single story should be consciously distributed across the whole team, regardless of geography. This is one of the most challenging areas for a distributed team as it requires the team to work closely across geographical boundaries but will lead to consistent user experiences and a reduction in functionality gaps between components.
  - Over specialization by geographical location and/or component is a sign that your team is succumbing to some of the communication challenges caused by distribution by changing the product to suit their development, not the customer's requirements.
- Patience remember building a good team, rapport, habits, and a rhythm take time.



## Give Your Team the Means to Succeed

*There is reason and sense behind valuing individuals and interactions over processes and tools. Processes and tools distill common experiences. Along comes an uncommon experience and processes and tools can't adapt. Only humans can. That's where the individuals and interactions come in figuring out how to make progress in unusual situations. A blind adherence to processes and tools in the face of novelty is as ineffective as persisting in variety, debate, negotiation, and creativity in the face of routine and repeating circumstances.*

— Kent Beck Tools for Agility

The decision to distribute teams needs to be coupled with the commitment to provide teams with the tools they need to maximize communication and the expectation that it will take them some time to optimize the work flows and communication processes around them.

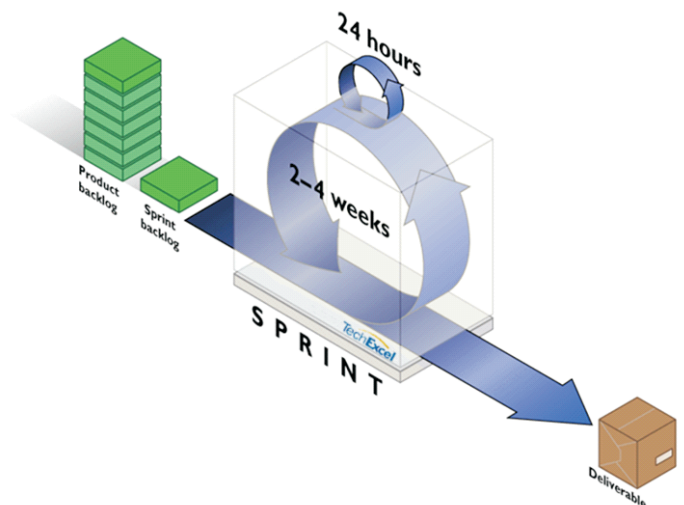
Though tools are not the primary focus of agile teams, or this paper, when selected properly they can certainly make teams more effective and efficient provided they are flexible and do not

## Conclusion

Distributed agile development makes it possible to tap into new global markets and obtain access to global talent while potentially reducing costs. Be sure to clearly understand the risks and rewards of team distribution before making the decision to proceed. One of the key factors in the success of an agile project is the high level of communication between team members. If your team is distributed they must make deliberate efforts to replace as much of the lost communication bandwidth with augmented processes and practices.

Bring teams together as often as possible, especially during pivotal points during projects. Use the points in time to nurture trust and strengthen working relationships. Agile development is difficult and requires a great deal of discipline, even more so when teams are distributed. Make sure you have someone with a

clear mandate to coach your teams and ensure they stay on track with communication practices. hinder a team's natural work flow. Distributed agile teams cannot rely on sticky notes, task boards, or burndown charts on a wall for project tracking. They require a robust system for sharing these items and metrics across multiple locations. Choose a tool with good performance across geographically distributed environments. Slow downs during simple tasks will cause frustrations that team members will seek to alleviate by cutting corners.



clear mandate to coach your teams and ensure they stay on track with communication practices.

Agile teams work on user stories not component features or implementation tasks. Organizing your team's work and distribution by component or discipline will make it extremely difficult to focus on what matters. That is, the stakeholders and their stories. Provide distributed teams with tools that will help them fulfill these user stories and work as effectively as possible with as few barriers to the distribution process as possible. Favor integrated systems that offer full traceability from specification design through development and testing. Finally, use an integrated wiki linked to tasks and stories to store notes from various design meetings and impromptu conversions.