

# THE DISTRIBUTED AGILE PARADOX: HOW TO OVERCOME THE CHALLENGES

**Aditya Shetye**  
*Sr. Manager, Neo Group*

**&**

**Hemant Puthli**  
*Partner & SVP, Neo Group*



**Beyond Advice to Outcomes**

Advisory | Governance Support Services | Real-time Risk Intelligence  
info@neogroup.com | www.neogroup.com

## Preface

By their very nature, the terms Distributed and Agile are seemingly incongruent as geographical and time-zone separation inhibit an agile way of working. So how do we find ourselves in this seemingly paradoxical dilemma of Distributed Agile? There are two forces at play. The first, driven by cost reduction and reduced time to market, is the widespread geographical distribution of software application development. Today, more than half of all software application development and maintenance is outsourced.\* The second, driven by goals of improved product quality and project performance, is the tremendous adoption of the Agile development methodology. Agile adoption rates across various industries are approaching 75%.\*\* At the intersection of these two forces, we find ourselves in the state of Distributed Agile.

The challenges that go along with practicing Distributed Agile are significant. So significant, in fact, that many companies are choosing instead to bring back in-house previously outsourced application development. In a previous Neo white paper, we labeled this trend “[Agile Insourcing](#)” and outlined the various advantages of this practice. But what if insourcing isn’t a viable option, can Agile overcome the challenges to succeed in an outsourced, multi-vendor, multi-location environment?

Surprisingly, the answer is yes. This white paper outlines best practices for successfully overcoming the challenges of Distributed Agile and includes an illustrative case study of a technology company’s successful distributed agile transformation journey.

## Why Distributed Agile is Difficult to Achieve

The Agile development methodology is a way of working that is focused on quick incremental developments through rapidly iterating cycles (as opposed to a compartmentalized and linear assembly line that moves in only one direction). This methodology requires a collaborative environment where cross-functional teams can produce, learn and improve. The ultimate aim is to meet user expectations with quick and continuous delivery of applications. In order to accomplish that, Agile focuses on releasing a ‘minimum viable product’ out to market as soon as possible and allows for changes to happen later in the cycle through each iteration.

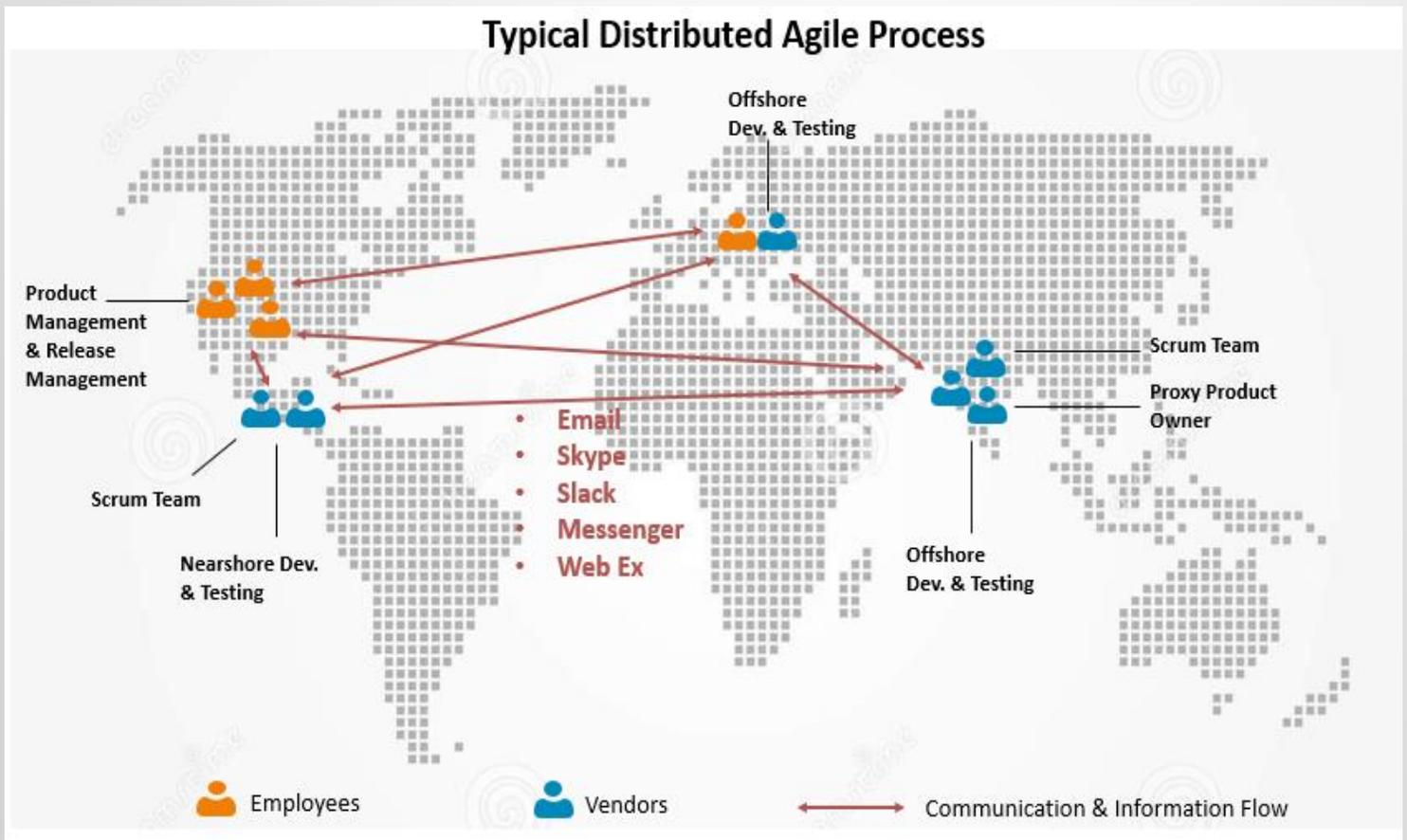
*\*The most commonly outsourced functions by IT leaders worldwide as of 2017 are software application (64%) and software application maintenance (51%) Source: [Statista](#)*

*\*\* As of 2018, the Agile adoption rates are as follows High Tech 74%, Discreet Manufacturing 74%, Banking & Finance 65%, and Retail 63%. Source: [Imarticus Learning](#)*

For Agile to work efficiently, the customer/business user needs to interact closely and frequently. The seamless exchange of information and knowledge critical to the success of Agile development are most easily achieved through co-located face-to-face meetings and interactions, and this is where the challenge with Distributed Agile lies.

In our previous [Neo Point of View: Agile Insourcing](#) we spoke in detail of the inhibitors of implementing Agile in an outsourced environment due to remotely distributed teams often operating in different time zones, including difficulties of:

- Synchronizing the coordination and communication between internal teams and remotely located teams
- Infrequent team interactions leading to a lack of focus, transparency, and trust
- Communication and goal cohesion challenges due to both locational and organizational cultural differences between distributed teams
- Teams gravitating to their comfort zones and defaulting to traditional practices of work pipelines between teams in a non-Agile, linear fashion
- Requirement of considerable overhead in the form of executive oversight and governance

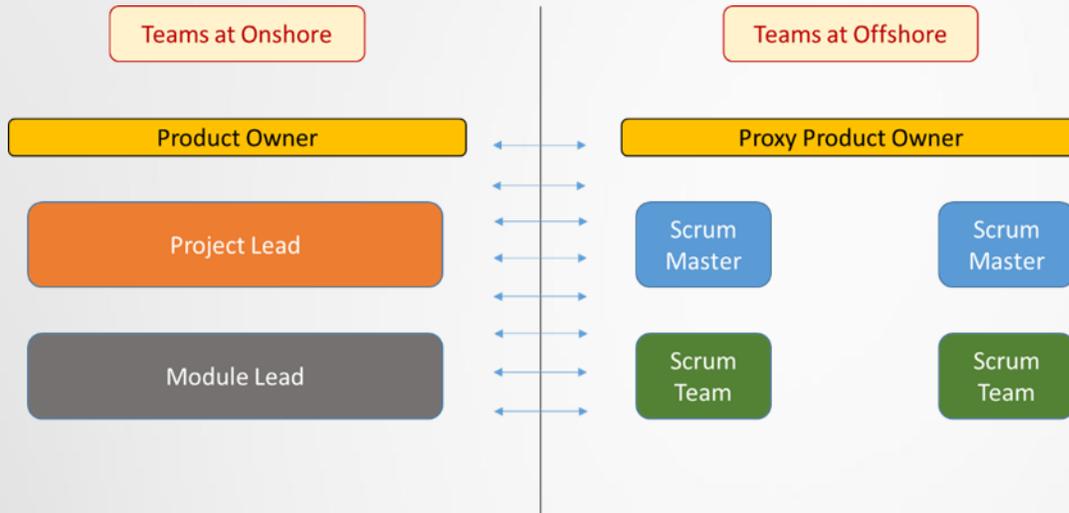


## How to Make Distributed Agile Work

In spite of the challenges, the following best practices can lead to the successful implementation of Distributed Agile:

- Transformation Approach – There are a number of steps that are critical to perform before starting a Distributed Agile transformation, including the vitally important establishment of a process roadmap. Perhaps for your organization a phased approach or starting with a pilot group presents the best opportunity for successful transformation. In order to make this decision, it's crucial to understanding the environment within which the transformation is taking place by conducting a health check and opportunity assessment. Effective metrics are also key to design beforehand to accurately measure progress along the transformation journey.
- Pick self-motivated individuals to form teams – With teams geographically distributed in different time zones, constant supervision is unrealistic. For Distributed Agile to work, teams need to be comprised of self-motivated individuals willing to take ownership of the work and see it through to completion with minimal oversight. Ideally each team should have at least one member with Agile experience. A best-case scenario is to assign a team member who has experience working on a successful Distributed Agile Team.
- Emphasize smaller teams – Relatively small teams are easier to manage and control. Larger teams can lead to greater communication issues and longer duration stand-up meetings resulting in a less Agile process. While the appropriate team size should be determined on a case-by-case basis, a good practice is to categorize them as small (6-8 members), medium (8-10 members), and large (10-12 members).
- Provide a sense of business impact – Structuring projects so that they have a meaningful outcome is a big motivator for the teams. Building trust by communicating to each team the impact of their specific tasks to the overall project will result in teams working more effectively and cohesively with the other teams towards achievement of ultimate project goal.

- Assign a proxy-product owner at each site – Although, it is ideal to have a product owner, who understands the technical, functional and business needs of the product at each site, in the case of distributed teams this may be impractical. As an alternative, enterprises can assign a proxy-product owner at secondary or offshore locations to coordinate and communicate with the onsite product owner at regular intervals to serve as the link between onsite and offshore teams.



- Sites with overlapping work hours – While the ‘follow the sun’ approach can help in reducing cycle times, waiting for the 8-12 hours to get answers from geographically dispersed teams will impede the agility of the process. It is a good practice to have some degree of overlap between distributed teams’ work schedules (ideally 3-4 hours) to facilitate effective and timely communication between teams.
- Invest in collaboration and communication tools/software – Given the distributed nature of teams, it is of paramount importance to invest in collaboration and communication tools/software to enable the real time communication and sharing of information Agile workflow requires. Tools/software for desktop sharing, teleconferencing and instant messaging will enable efficient daily stand-up meetings, showcasing iterations and demos, and providing real-time feedback.

- Daily stand-up meetings – Stand-up meetings are critical to the sprints required for the successful execution of Agile. These meetings ensure regular and effective communication as they play a critical role in discussing iterations and feedback. These meetings are one reason we recommend that distributed teams have overlapping hours as conducting these meetings during those hours is ideal.
- Establish robust feedback mechanism – One of the most important practices for successfully running a Distributed Agile model is to establish a robust feedback mechanism. One of the core principles of Agile is producing quick working prototypes, gathering customer feedback and making iterations to the prototype. Valuable feedback for co-located teams can happen outside of stand-up meetings and official feedback sessions in the form of casual interactions. With Distributed Agile fewer of these feedback opportunities exist. Therefore, globally distributed Agile teams need a structured and robust communication plan that fully utilizes collaboration tools and maximizes all opportunities (stand-up meetings, prototype discussions, iteration discussion and demos) to provide feedback.

## Case Study: Distributed Agile Transformation Journey

### Situation

The software development organization of a technology company wanted to develop a XaaS platform (henceforth referred to as product) and related set of services to its customers who then use it to provide services to their clients. Approximately 30-35 external contractors and consultants developed the product. The technology company acquired the product, and the newly built organization had to focus on knowledge transfer from external consultants and contractors to its employees. The development organization's teams and locations grew rapidly, and soon the company was operating from 5 sites in 3 countries. The company wanted to adopt Agile software development across its global dispersed locations.

### Motivation

The company was dissatisfied with its current way of working especially in software development space as there was a need for continuous deployment and adapting to rapidly changing customer needs. Like the majority of technology companies\*, Agile software development was becoming a part of the company's overall corporate strategy.

## Action Taken

The company decided on a three-phased approach to Agile transformation. Specifically, the managers used an experimental approach of small but critical changes in a stipulated time. They started with the introduction of Agile methodology by creating cross-functional and cross-component teams who were given the freedom to decide on Agile implementation under the guidance of Agile coaches.

A pilot team was formed which consisted of resources from two locations who had shown keen interest in Agile ways of working. While working with the pilot team, the company realized that it was also important to have someone in the team who understood the overall business flow. Thus, the team composition was modified to include a business-flow expert.

Once they got the pilot team rolling, the company decided to scale their Agile adoption. Unfortunately, this rollout did not go smoothly. Most of the teams ended up being cross-site and as a result, people who had never met each other were working on the same team. This led to lack of trust which affected the efficacy of the process and outcome.

To overcome this challenge, the organization decided to have a series of 'value workshops' to define a common purpose and value so that the sites were working together towards a common goal rather than competing against each other. The coaches and managers also created a future state organization to showcase what the organization would look like 2-3 years down the line after a successful Agile transformation.

The next step was to work towards continuous integration and deployment, but that step also had its share of challenges. They created new teams who focused on implementing CI and test automation. The team members were pulled from another product where Agile methods had been used successfully for some time. A future goal was to spread CI knowledge and mindset across the organization by organizing CI Road shows.

## Result

The company was able to sustain its large-scale Agile transformation by mitigating the risks and challenges through the adoption of some of the above listed best practices while taking an experimental and incremental approach to ensure that everyone was aware of and working towards a common vision and value.

## Conclusion: A Neo Point of View

Whether Distributed Agile is right for your organization requires weighing the pros and cons as well as many organizational factors, including strategic goals, geographical footprint and expansion strategy, in-house talent and capability, and budget constraints to name a few.

While Agile development is definitely simpler to practice with co-located teams, Distributed Agile can work if the organization aligns itself to the Distributed Agile best practices we've outlined in this white paper.

Neo Group has collaborated with many organizations guiding them through Distributed Agile solution design, successful Distributed Agile transformations, and vendor selection for Distributed Agile transformations. For more information on how Neo Group can help your organization with a Distributed Agile transformation from the design of a process roadmap and metrics to conducting a health check and opportunity assessment all the way through successful completion of the transformation process, please email [info@neogroup.com](mailto:info@neogroup.com).

## ABOUT THE AUTHORS

**Aditya Shetye**  
**Sr. Manager, Neo Group**

Aditya brings over 10 years of diverse experience across industries and functions and 5 years of experience in sourcing consulting. Aditya has been a part of large engagements while at Neo Group - with US based banking, publications and pharma multinationals, aimed at assisting them in the development of their outsourcing and transition strategies, and governance. The scope of work across included comprehensive portfolio assessment and transition planning, role rationalization, vendor negotiations and more.



**Hemant Puthli**  
**Partner & SVP, Neo Group**

Hemant brings over 30 years' experience in IT and Management Consulting Services. Over the last 12+ years, he has led Neo Group engagements with several clients across diverse industries, involving strategic transformation projects such as sourcing portfolio assessment, partner selection, contract negotiation, sourcing/ governance organization design, governance tool implementation and support services oversight, health-checks and readiness assessments.



Since 1999, Neo Group has advised and supported Global 2000 enterprises, empowering them to build new capabilities and generate rapid savings by leveraging global talent, digital technologies and analytics.