

The Agile Journal is an online monthly e-newsletter focused readers with the need-to-know resources they need to develop agile business.

HOME ARTICLES AND COLUMNS FORUMS ABOUT



Home ► Articles and Columns ► Articles ► Achieving Agility in Globally Distributed Software Development

Achieving Agility in Globally Distributed Software Development



IN THE

WRITTEN BY ADAM FINDEN
THURSDAY, 06 APRIL 2006

In today's business climate there exists an ever-increasing demand to achieve more from less. More return from less investment, faster time to market from shrinking resources, higher quality from collapsing timelines. The impact of these pressures on the software development industry has meant that organizations have had to look for new avenues such as offshore development to reduce costs yet still satisfy these increasing demands. Simply incorporating an offshore development strategy to realize lower costs is not a solution. Leveraging the lower cost and resource scalability that an offshore development strategy provides must also include the facilities to enable that capability to produce better results faster.

Agile Offshore

At first glance the terms agile and offshore seem to contradict each other; how can these two terms coexist? The term Agile has become synonymous with collaborative development where the business is meant to remain very close to the development effort, and where communication is meant to be fluid and personal. This is in stark contrast to the traditional picture of offshore development if for no other reason than geography thwarts the applicability of many practices that most agile methods are based on. This would be true if agile development was only about published practices such as pair programming. Instead agile development is about the principles that these practices are meant to embody. Performing a practice is useless if it does not result in the desired behavior. Instilling the principles of agile development is the key.

The Agile Landscape

At Valtech, we have found that picking an agile development methodology and applying it does not result in better software faster; in fact branding development as XP or Scrum, although perhaps lending focus to sets of practices, is not that important. Instead, what is more important is recognizing the principles. We have found the following underlying principles are what drive distributed agile development.

Communication

Communication can not happen too much or too early. There is a tendency in offshore development to bring the offshore teams into the picture when it is time to implement. The big picture is never acquired by the team. Most likely this happens because it is easier to communicate less than communicate more, and then only when it is absolutely necessary. Offshore development requires a commitment to constantly focus on increasing communication from all stakeholders.

Tackling
An Agile
Develop
Achieving
Develop
Agile at
Outsour
EPAM
Stretch
CASE S
FEATUR
Facilitat



COMIN

May - Ir
Develop
Jun - Me
Jul - Ser

Geographical issues must be addressed to enable collaboration. Rarely will a strong product result from the delivery of design artifacts from an onsite to offshore team for implementation. It is essential for the offshore team to be involved in as many analysis and design discussions and decisions as possible.

Visibility

Even with a heightened effort placed on communication, visibility may still not exist. This is because the first focus of communication is to communicate to the project team. Visibility focuses on communication being a two-way street and is achieved by taking part in the discussions in which the team hashes out issues and relays progress information amongst themselves. Visibility is about seeing the speed bumps as well as small victories the team has made.

Monitoring Progress

Facilities must exist to radiate the pulse of the development effort and identify trends. One of the traditional issues in offshore development is fear of the unknown. Fear that those requirements that were delivered 6 months ago have actually seen progress. The current state of development needs to be made visible to all interested. Publishing a project burn down and velocity trend helps to provide a picture of past, current and future state.

Feedback

Without feedback there can not be adaptation. By placing working code early and often in front of users, feedback becomes possible.

Seeing and Eliminating Waste

Recognizing what does not lend value to the development effort and removing it promotes a focus on productivity. This is somewhat interesting in distributed agile development. There typically will be more documentation and ceremony required. This is needed to help bridge the distance. What may well be considered waste in a small co-located agile team could be essential to a distributed team. Recognizing this fact and accepting this but also constantly re-evaluating potential waste is the goal.

Value-stream Mapping

If an interested party cannot be identified then the activity should not be done. This helps to identify waste but also helps in scheduling and prioritizing work. A common concern from organizations new to agile development, especially offshore, is the issue of not having all the requirements. Even though requirements are never complete, having the safety net that they are declared to be complete seems to allow people to proceed. By scheduling work based on the biggest problems facing the business that have yet to be solved, the greatest amount of value is delivered to the business more rapidly. Not only that, but due to their importance, these requirements are more often than not front and central and therefore more complete.

Early Risk Mitigation

Early risk mitigation means continuously addressing the hardest issues. In many cases, the most important business priorities are also the most architecturally significant but this is not always the case. So risk must be a constant consideration. Risk shouldn't be postponed because it only gets worse.

Stepping back from the details, the Agile Manifesto

Aug - Tr

Sep - Re

Oct - Ma



Usernan

Passwor

Ren

Login

Lost Pas

No acco

WHO'S

We have

RSS FE

crosses ten-and-one-half time zones, cultures and possibly languages is quite a different task. To tackle this task the first step is to pull back from the details and investigate the source.

The Agile Manifesto investigated

From The Agile Alliance (<http://www.agilealliance.org/>) the Agile Manifesto states:

- ▶ People and interactions over process and tools
- ▶ Working software over comprehensive documentation
- ▶ Customer collaboration over contract negotiation
- ▶ Responding to change over following a plan

The Agile Manifesto has typically been interpreted as favoring the left hand side (people and interactions, working software, collaboration and responding to change) so dramatically so that it paints a dark picture of the right hand side. Rather, the maximum value that fits an environment is going to exist somewhere along the continuum of each statement in the manifesto, similar to the way an equalizer is adjusted for ideal sound. The focus on left versus right should be adjusted to fit your environment, as illustrated in Figure 1.

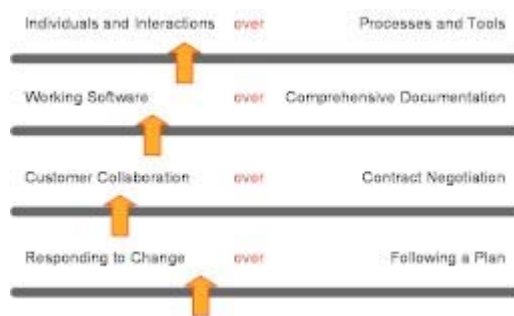


Figure 1. Agile Manifesto continuum

Practices and Techniques of distributed Agile Development

Accepting that a somewhat higher level of formality may be required to address the issues inherent to distributed development, we have found that basing our Agile development processes on the phases of the Unified Process to be the most effective. This allows for the inheritance of phase gate criteria, giving useful internal milestones to the teams as well as serving as a source of artifacts that could be used. Furthermore, we have found that the practices of Scrum fit nicely with the distributed nature of our teams, although some adaptations have had to have been made. With these two high-level guiding lights the following 11 practices have emerged as useful tools to consider:

1. *Communicate The Contract.* It is crucial to provide as much information as possible to the offshore team to ensure they are aware of contract details and restrictions as early as possible. Without this information decisions can not be made. If a team is to self-manage it needs the tools to make informed decisions.
2. *Release Information Early.* When asked, the majority of offshore teams will say they receive information too late and most likely they will also say they don't receive enough. It is extremely important to open communications channels early and keep them open. It is essential that when issues and changes arise they are immediately imparted to the offshore team.
3. *Onsite/Local Facilitator.* An onsite presence typically acts as a conduit between the client and the

stops. The conduit model simply places a façade of communication over the development effort. In contrast, a facilitator's job is to create and open sustainable communication channels.

4. *Iterative Development.* It is not surprising that iterative development is a must-have practice for distributed agile development. But, as it is such an important concept to agile development it can't be taken for granted. It must be understood that iterations have specific goals of fully-functional, tested deployable code, and that they must be monitored and tracked. Measures such as velocity should be collected and monitored so that trends in productivity and results can be understood and adaptations made.
5. *Client Collaborator Drives Iteration Demos.* An interesting practice that has emerged as very beneficial is having a customer or client run the iteration demos. This tends to lead to a much more collaborative effort between the client, customer representative and the offshore team.
6. *Global Scrum Meetings.* Global Scrum or stand-up meetings tend to be less fluid than co-located Scrum meetings, but they are still essential. This is due in part to language barriers and communication tool limitations. Furthermore, it is essential that the client be invited to every one of the Scrums. What has also emerged as a strong practice is the capturing of notes from the Scrum meetings and making them available. The point is not to spend time on the notes but rather to quickly capture the highlights and obstacles if identified.
7. *Knowledge Transfer Phase.* At the time the offshore team engages (typically late elaboration or early construction, if mapped to the UP phases), we have found there needs to be a two- to three- week dedicated knowledge transfer phase. This isn't an iteration because the goal of a functional increment is absent. Instead, it focuses on how the system or systems fit into the organization and on domain knowledge training. Furthermore, what has been found to be most effective is reverse travel in which a client representative travels to work with the offshore team.
8. *Globally Visible Backlog.* To assist in the realization of transparent development, visible project and iteration backlogs have shown themselves to be essential. They provide for a common focal point for the distributed team.
9. *On-demand Globally Visible Information Radiators.* It isn't enough to have a visible backlog. There should also be information radiators of project health and progress. Trends must be visible such as how much work is being burned, discovery rates, scope changes and velocity.
10. *Continuous Integration.* Continuous integration has emerged as an essential aspect to application development when geography and time zones are not part of the equation. With the added complexities that distributed development brings to the equation, continuous integration becomes nearly essential. The possibility of achieving around-the-clock development is next to implausible if working, tested software cannot be validated at all times.
11. *Leverage Testing Ability.* One aspect of offshore development that has been present for quite some time is the capacity for an offshore capability to facilitate testing. The opportunity present is to leverage the testing knowledge and capacity into a cross-functional and agile development team. Testing experience exists offshore. Leveraging that expertise to include testing as a development activity has been found to be instrumental in lowering defect rates and increasing productivity.

Mapping the Practices to Principles

In the first section of this article, a set of principles were outlined as high level goals that, if achieved, can promote agility in distributed development. Following that were a set of practices that have been found to be effective in realizing these principles. However, it is still useful to map the various the practices to the principles to gain an understanding of how variations and adaptations of the practices can still promote the goal of agility (see Figure 2).

	Communication	Collaboration	Visibility	Monitoring Progress	Feedback	Seeing and Elimination Waste	Value-stream Mapping	Early Risk Mitigation
Communicate the Contract	■	■				■	■	■
Release Information Early	■	■				■	■	■
Onsite / Local Facilitator	■	■		■		■	■	
Iterative development	■	■	■	■	■	■	■	■
Client Collaborator demos		■		■	■			
Global Scrum Meetings	■		■	■				■
Knowledge Transfer Phase	■					■	■	■
Globally Visible Backlog	■	■	■	■			■	■
Continuous Integration	■	■	■	■	■			
Leverage Testing capability		■				■	■	
Information radiators	■		■	■	■			

Figure 2. Mapping Practices to Principles

As the chart illustrates, no single practice stands alone in realizing a specific principle. This means that although each practice has emerged as beneficial, dropping, adapting, augmenting or modifying a practice or set of practices does not leave a gap in the quest for agility.

Summary

The terms agile and offshore, at first seemingly contradictory, can in fact co-exist. The question is not whether offshore development can be agile but rather *how* offshore development can be agile. This question is hard to answer if the focus is placed on the various agile practices, but becomes understandable and answerable if instead it is the principles of agile development that are investigated. Once the principles are identified, focus can then be placed on how to achieve these principles given the complex environment of distributed offshore development.

About the author:

Adam Finden is the Director of Global Sourcing for Valtech, a global software development consultancy. In this role he is jointly responsible for the development and evolution of Valtech’s globally distributed Agile Offshore process. Additionally, Adam is also active as a consultant assisting clients in object modeling, enterprise application integration and process adoption.

[Discuss this article on the forums. \(0 posts\)](#)

[Next >](#)

[\[Back \]](#)

Agile Project Management

Agile Software Development Mgmt Track
requirements, tests, defects

Ads by Goooooogle

Agile Whitepapers

Download free whitepapers about agile and
iterative development.

Advertise on this site

© Copyright 2006 **CM Crossroads**. All rights reserved.

[CM Yellow Pages](#) | [ALM Expo](#) | [CM Today](#) | [Configuration Management Journal](#) | [CM Crossroads](#)