

## **Adapting Agility in Offshore Software Development**

Muhammad Sajid Rehman

Lecturer, Qurtuba University of Science & Information Technology, Peshawar, Pakistan

sajidrehmann@gmail.com

### **Abstract**

Software developers are trying to optimize their processes to better support the business in a way to reduce cost, improve quality, quick deliveries, gain credibility and improve time to market. In order to achieve these goals, many of the organizations moved to offshore outsourcing. Some of them moved to agile application development and a very few use both of them. At first look, offshore software development with agile practices seems to be a mismatch. Most of the people think that the blend of these two extremes is like inviting a disaster. On the other side, a few people after experiencing, now believe that blend of these two extremes methodologies can bring remarkable benefit to both ends.

This paper focuses on the intersection of the two extremes i.e. "offshore software development and agile software development". Furthermore, while highlighting the major challenges in blending these two extremes (i.e. offshore software development and agile software development) the paper attempt to identify areas where to intersect / avoid the blending.

### **Keywords**

Offshore Software Development, Agile Software Development, Outsourcing, Agile Principles, Project Management

## **1. Introduction**

### **1.1 Off-shoring:**

Offshore outsourcing (off-shoring) has gained much more popularity in the recent years. One of the major reasons of the popularity is the cost benefit. Off-shoring deals with handling overall/ major area of the software to a third party companies in low wage countries. One can better understand the cost benefit aspect when taking into consideration that the average annual salary for a US developer is \$70,000 in 2004, which is much greater than the average salary \$13,580 of an Indian developer, according to Electronic Business Magazine (Vogel & Connolly, 2005). In addition companies offshore their non-critical software and project management functions while they themselves focus on business critical initiatives. Other benefits include potential tax savings, achieving high quality through off shoring and 24 hours development.

Nothing is without tradeoff, when anyone thinks about the offshore software development, he/she must consider all the potential drawbacks that include less communication, geographical distance, cultural and

time zone differences, social and potential backgrounds, languages and technology differences (Sauer 2006).

## **1.2 Agile Alliance:**

In past few years there are different software development (iterative / incremental) process models that claim to be agile. Giving a formal definition to “Agility” the “Agile Alliance” and the publication of its manifesto appears in 2001 meeting (<http://www.agilealliance.org/principles.html>). The manifesto of the “agile alliance” is a solid definition of the values and goals of “Agile Software Development”. A number of different principles were explained in the manifesto including high priority to customer satisfaction with continuous and frequent delivery of working software, encourages teams work in corporation with business people, embracing changes, focus on simplicity, face to face communication, providing good and supportive environment to the team members and promoting sustainable development. Other principles includes continuous integration, testing and continuous attention to technical excellence (<http://www.agilemanifesto.org>).

## **2. Integration of agile and offshore software development:**

Keeping in view the benefits of agile and offshore software development companies are integrating agile development process and offshore outsourcing in order to get maximum benefits. In a glance offshore software development using agile principles seems to be a big mismatch. Agile development focuses on face to face communication, while communication is considered to be a big challenge in offshore software development (Nisar et al.,2004). Following section will show list of challenges and their solutions.

## **3. Challenges and their solutions:**

**3.1 Communication:** There are different challenges when applying agile methodologies to offshore software development but communication is the major one. Agile methodologies focus on “face to face communications” (<http://www.agilealliance.org/principles.html>) while same is very difficult to achieve in offshore software development. In offshore development teams are geographically distributed and hence face to face communication is difficult to achieve (Shahzad et al., 2011).

**Solution:** There are two major solutions for the said problem.

S1: The first solution is to provide dual-shore development that is on-site and offshore team i.e. on-site and off-shore team. Both sides will distribute the work. This dual shore development will bring multiple advantages.

- A. Face to Face communication will be achieved.
- B. With dual shore development now development will be performed almost 24 hours a day because of the distribution of work in different time zones i.e. when on-site development team like in US will finish their work, Pakistan, China or India’s team will then take start at that time.

S2: Use of technology like wiki, Skype, telephone, emails, virtual whiteboards make possible face-to-face communication. Beside this team members should adjust their working hours in order to have few hours of overlap among the teams.

**3.2 Short Iterations – Frequent Builds – Continuous Integration and Testing:** Agile software development rely mostly on short iterations, frequent builds, continuous integrations and testing. These challenges put the scalable configuration management and version management into problem (Phalnikar et al., 2009).

**Solutions:**

1. Common framework can be used among the team in order to avoid integration problems.
2. The core architecture of the application is solidified by a small and experienced on-site team during the first iteration. This way offshore developer can start up on an established code base and a solid architecture (Sauer 2006).
3. Automated testing, issue/backlog tracking, and cruise control can be used.
4. Additional tools like Maven, Check Style, CVS and Jira etc can be used that supports continuous integration.
5. Common infrastructure/configurations of hardware and software should be provided along both sides.

**3.3 Embrace Changes Even Late in Development:** One of the principles of agile software development is to embrace change even late in development. Customer requirements are not stable and they are changing very rapidly (Beck 1999). Agile software development handles this issue via frequent communication with customers/stack holders. Now injecting this practice to offshore software development creates a lot of problems because customers are residing far away from the offshore team.

**Solution:**

1. The very simple solution is to communicate with the customers/stack holders via audio/video visual aids.
2. We can replace the real customers with the “Proxy Customers”. Proxy Customers are well equipped and should be able to support business and technical project members.
3. Acceptance tests can also be used as ways of communicating requirements.

**3.4 Project Management:** Project Management again is one the major challenge when blending offshore and agile software development. When dealing in offshore projects are always divided into multiple teams. In such situations the divided tasks, their progress estimations create problems. Therefore offshore project are difficult to manage (Andersen 2009). In offshore software development the project manager is the person who divides the work and responsible for progress estimation. Here managers are following a traditional method of “Command and Control”.

Agile principles focus much more on the developers. In order to implement agile principles the developers need more autonomy and decision making powers. They should not follow “Command and Control” traditions. Agile principles state that rather than managers, developers are responsible for the division of work and efforts that are required for the future tasks.

#### **Solution:**

1. In order to give empowerment and decision making powers the offshore team should be provided more authority and responsibility
2. Special training should be provided to the employees to realize them that they have the autonomy of making decisions. Ultimately this autonomy will act as a motivator through which they will become more responsible and productive as well.
3. Project management should be handled through the risk assessment meeting among the major stake holders once a week.
4. Agile principles of frequent delivery should be implemented with care for the better measurement of project progress.

#### **4. When to Blend and When to Avoid:**

No doubt by injecting these two can bring remarkable benefits, however this is not necessary that they should be always blend. There are areas where they can be blend likewise there are situations where blending of these two will create problems.

1. Small projects should be avoided to developed offshore i.e. they should be developed on-shore using agile methodologies (Fowler 2006).
2. Avoid offshore development initially in the projects which are high complex in nature or more strategic. In such projects the on-shore team should provide the core architecture in the fist iteration and later on the offshore team should start their work on an established code base and solid architecture.
3. There is no need to blend offshore and agile software development, if offshore development is achieving quality, service and productivity benefits.
4. Those developers who are already using agile development and if they want to get more advantage and wants to develop software in less cost can blend and handover development responsibilities to lower wages countries.

#### **5. Conclusions:**

Offshore software development and agile software development were seemed to be the two extremes, but after studying and comparing them (in the challenges-solutions mentioned above), It is concluded that the proper blend of these two extremes can bring remarkable increase in the production

cost, quality, service and productivity. Moreover, the blend of these two is not for everyone/project. Proper care should be taken to get the fruit of the blending positively.

## References

Andersen, J. H. (2009). Project Management in Agile Software Development.

Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70-77.

Fowler, M. (2006). Using an agile software process with offshore development. *Capturado em* <http://martinfowler.com/articles/agileOffshore.html>.

Fowler, M. Using Agile Software Process with Offshore Development. (<http://martinfowler.com/articles/agileOffshore.html>)

<http://agilemanifesto.org/> (March. 2014)

<http://extremeprogramming.org/> (June. 2014)

[http://www.scrumalliance.org/view/scrum\\_framework](http://www.scrumalliance.org/view/scrum_framework) (as of June. 2014)

Nisar, M. F., & Hameed, T. (2004, December). Agile methods handling offshore software development issues. In *Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International*, 417-422. IEEE.

Phalnikar, R., Deshpande, V. S., & Joshi, S. D. (2009, January). Applying agile principles for distributed software development. In *Advanced Computer Control, 2009. ICACC'09. International Conference on*, 535-539. IEEE.

Sauer, J. (2006, August). Agile practices in offshore outsourcing—an analysis of published experiences. In *Proceedings of the 29th information systems research seminar in Scandinavia, IRIS*, 29, 12-15.

Shahzad, F., Ahmed, S., Sher, F., & Ullah, K. (2011). COMMUNICATION CHALLENGES IN SUCCESS OF OFFSHORE SOFTWARE DEVELOPMENT. *International Journal of Reviews in Computing*, 8.

Vogel, D. A., & Connelly, J. E. (2005). Best practices for dealing with offshore software development. *Handbook of Business Strategy*, 6(1), 281-286.