



Agile Record

The Magazine for Agile Developers and Agile Testers



October 2010

issue 4

www.agilerecord.com

free digital version

made in Germany

© Tyler Olson - Fotolia.com

Do You Need a Project Manager in an Agile Offshore Team?

by Raja Bavani



© Lida Saitan - Fotolia.com

Software Product Engineering in a distributed environment requires optimal utilization of teams as well as hardware, software and related resources in order to improve speed to market under budget constraints. In this context, there is a tendency to propose a reduction in management overheads in distributed models and form extended teams that report to managers working at remote locations. This may work for very small extended teams of 1 or 2 engineers working on production support or routine maintenance tasks. Does it work for larger teams as well? Many times, practitioners tend to embrace agile principles and recommend a self-directed team of offshore engineers that can work with an onsite manager. Here the compelling question is on the need of an offshore project manager. This gives rise to several related questions such as

- a) Do self-directed teams need a leader or a manager?
- b) When there is an onsite agile team that reports to an onsite Project Manager, why do we need an offshore Project Manager for an agile offshore team that is going to work with the onsite team?
- c) What is the role of a Scrum Master at the offshore location?
- d) Why can't an agile team report to a remote manager or Scrum Master?

Let us start this discussion with the assumption that we are implementing agile practices through a home grown methodology or an industry standard agile methodology such as Scrum in a distributed model.

Scrum prescribes 3 roles: Product Owner, Scrum Master, and Team Member. Typically, the Product Owner owns product specifications and provides the same to the Scrum Master and the rest of the team. On the other hand, the Scrum Master facilitates the process of software creation by working with the team and

enabling team members to find solutions to problems. In a way, the Scrum Master is responsible for hiring, employee development and grooming, too. The role of a Scrum Master comes with an adequate command to lead the team. However, it does not involve control. The team controls itself and gets the necessary coaching from the Scrum Master. Agile teams are self-directed teams. Team members work together, support each other and solve their problems. They reflect and improve. They inspect and adapt. The classical 'Project Manager' role is a loaded role. It combines both the 'What' and 'How' parts of Software Project Management, whereas the role of Scrum Master revolves around the 'How' part alone. This is because the Product Owner takes care of the 'What' part and is responsible for providing product specifications to the Scrum Master. A typical Scrum team has 7 to 9 team members. For every Scrum team, there is a Scrum Master and a Product Owner who are part of the team. In this team, a Scrum Master needs to be co-located, whereas the Product Owner can be from a remote location. Scrum practitioners strongly recommend this structure. This is because, without a co-located Scrum Master, the team will not have a coach or a mentor to go to. In fact, on a need basis, Scrum Masters mentor their teams in implementing Scrum. Also, the Scrum Master will not see the team in real time and understand when to intervene and support in order to remove impediments or resolve issues. According to Scrum practitioners, having a remote Scrum Master and leaving the team alone is the first step to ensure project failure.

This answer may not be very convincing when our context does not involve 'Scrum'. This leads to questions such as 'We do not use Scrum. We use a home grown agile methodology and our onsite Project Manager will provide all necessary details to the offshore team. Why do you need another manager at the offshore location?'

Fair enough. Let us explore this from the expectations we have on self-directed teams. In a self-directed team, everyone is responsible for asking questions, answering questions, owning up to

situations and resolving problems. However, it is very uncommon to see self-directed teams that go on a mission without a manager or a coach. The manager of a self-directed team manages the context or contextual situations. The role of the manager is not to micro-manage team members. This role involves real-time observations, interactions and assessments of situations for timely corrective actions. This role involves consolidation of observations and events in order to understand if there are any issues that may impact the project goals. It does not stop here. This is a critical role that binds the team together and enables conflict resolution when required. This role helps the team by means of influencing external teams, or supports teams that owe a timely response or output for the self-directed team to perform. This role involves identifying events that require a root cause analysis or reflection in order to incorporate continuous improvement. Also, this role involves initiation of appreciations and celebrations to complement team members.

With these thoughts, can you think of offshore agile teams that function without a co-located manager (or a Scrum Master if you practice Scrum)? Or have you seen successful products delivered with such an optimization? We have not seen this happening. Based on our interactions with industry experts, not having a co-located role such as Scrum Master or similar, is a sure recipe for disaster. This is why in our engagements we strongly recommend such a role for each offshore team.

According to Alistair Cockburn who is one of the co-founders of the Agile Manifesto, Software Development is a people intensive cooperative game. Every orchestra needs a real-time conductor. Every football game requires a real-time coach as well as a manager. Every space mission has a leader. This applies to Software Development as well.

During August 2010, Len Bass, Senior Member of Technical Staff, Software Engineering Institute of Carnegie Mellon University, in his key note address at the IEEE International Conference on Global Software Engineering (ICGSE 2010, Princeton, NJ) made a very good comparison of Software Architectures and Software Project Teams in distributed environments. Software Architecture has two primary things – structure and behavior. Software Architects define the structure of any architecture depending on its expected behavior. That is, the behavior drives the structure, and the structure needs to deliver behavioral expectations. The same things hold good for distributed software teams. When you are structuring any team, try to identify the qualities and results that you expect from the team. That will help you define the structure.

To summarize, let us revisit our question. Do you need a Project Manager in an Agile Offshore Team? Well, it depends on the expected behavior of the team. For very small teams of 1 or 2 engineers that do monotonous work, such as bug fixing or maintenance of end-of-life non-critical products, you may be able to manage with a remote Project Manager. However, in all other cases, you will need to structure that team in such a way that it gets adequate local leadership and managerial support to deliver the best. If you follow Scrum, you will need a local Scrum Master for every project. Else, you may need a 'Project Manager'

or a similar senior role to support your local team to deliver the desired behavior.

Eventually, defining the structure of distributed teams, so that engineers at any location are not treated as augmented team members reporting to a manager or a leader at a different location, is very critical for the success of distributed agile projects. ■

> About the author



Raja Bavani

heads delivery for MindTree's Software Product Engineering (SPE) group in Pune and also plays the role of SPE evangelist. He has more than 20 years of experience in the IT industry and has published papers at international conferences on topics related to Code Quality, Distributed Agile, Customer Value Management and Software Estimation. His Software Product Engineering experience started during the early 90s, when he was involved in porting a leading ERP product across various UNIX platforms. Later he moved onto products that involved Data Mining and Master Data Management. During early 2000, he worked with some of the niche Independent Software Vendors in the hospitality and finance domains. At MindTree, he worked with project teams that executed SPE services for some of the top vendors of Virtualization Platforms, Business Service Management solutions and Health Care products. His other areas of interests include Global Delivery Model, Requirement Engineering, Software Architecture, Software Reuse, Customer Value Management, Knowledge Management, and IT Outsourcing. He regularly interfaces with educational institutions to offer guest lectures and writes for technical conferences. His SPE blog is available at <http://www.mindtree.com/blogs/category/software-product-engineering>. He can be reached at raja_bavani@mindtree.com