

Introducing Agile In Non-Agile Environments

By Ralph Moore, PMP, PMI-ACP

So you want to be Agile? Well, read on then.....

Introducing Agile to your organization sounds great in the boardroom, but once out in the trenches of your office it may not seem the way it was discussed in the boardroom. I will share my insights and try to help you avoid some of the common pitfalls that I have seen during my journey in the Agile world. In my experience there can be a huge difference in the way Agile functions in different size organizations and especially virtual work. I will focus on small organizations that are under a thousand employees total with the development team co-located on site with common sense as my guiding force. In my experience, some of the more important items to get ironed out quickly will be addressed:

- **Organizational Functions**
 - Operations
 - Maintenance
 - Emergencies
- **Organizational Culture**
 - Organizational commitment
 - Change and Stakeholder mind set change
 - Siloes
- **The Project Team**
 - Role changes
 - Agile adds overhead
 - Time-boxes

Hey everyone, we are going Agile but what about running the business? Yes, your developers more than likely handle a wide variety of issues on a daily basis; let's take a look inside of the daily grind:

Operations are the key to producing your product, providing your service or just plain old operating your business for success. So how do you stop your developers from this key business function so they can be Agile? Good question. Well each organization will handle this differently. The best scenario is for the developers to work on their projects and release reliable software on schedule and within budget. Is this always possible? Maybe not. So now what? So, how did I handle this situation? Well, what worked best in my experience is that we assigned a day to each developer where they would do operations related tasks. This time would be calculated into the project that was being worked on so we still could provide a realistic view of the work to be completed in the sprint. Maybe not the best option, but one that worked for me in the past.

Maintenance is needed to keep all these nice applications in top working order but all of my developers are now Agile. Who will do all of this maintenance now? Well, in my experience, a well-defined preventive maintenance schedule works pretty well especially if the maintenance is performed in off business hours. I have the team self-organize and develop a schedule and assign out the maintenance tasks. This still lets your Agile team be Agile and spend time in the sprint performing project work. Depending if the Agile team is salary or hourly, it can be an opportunity for the developers to put a little more cash in their pocket which always improves morale.

Emergencies, HELP everything is breaking. So how do we handle these emergencies? Well in my experience an emergency is just that and rather hard to plan for. So usually any emergencies take precedence and will be handled by the appropriate amount of developers. I have always made sure that my departments have a disaster recovery plan in place. This can reduce stress and the amount of time it takes to recover an application. The emergency will be managed by a supervisor, management will be updated and upon resolution a retrospective and lessons learned will be conducted. Unfortunately this time cannot really be planned into the sprint and can result in some of the story points not being completed and remaining on the product backlog.

Ok, alright, we have learned a bit about all the organizational functions, let's be Agile now. Well, slow down there a bit. Let's look at the organizational culture and how that can derail your Agile experience before you ever even get it started.

Organizational commitment in my mind is the make or break item. If you do not have support from your organization at the executive level, well then any change will be difficult but implementing Agile may be doomed from the start. Why you ask, do I need organizational commitment? Well, getting the buy-in from each department will be a lot easier if the executives tell them that they will support this new business model.

Ok, so you have Organizational commitment but now we have to **Change** the way everyone has always worked, "hey that's how we have always done it", as old habits are hard to break. Stakeholder mindset change is a hard task to accomplish; in my experience there are a few ways to ease the stress for everyone when a big change is happening:

- An initial meeting to explain the change and why it is needed
 - Explain the basic principles of Agile
 - Take suggestions – employees will feel empowered
- Identify training needs
 - Supply required training
- Have a plan for implementing the change
 - Publish the plan to employees
 - Know that there will be problems, be patient and work through the issues
- Keep everyone updated on a regular basis

Everyone now understands the change, has training and we have a plan; hopefully your company is not operating in Siloes. A silo is where all departments kind of do their own thing and do not communicate well. This is in general a bad business practice but is devastating to implementing Agile. Silos are tough to overcome. If silos exist they have more than likely been in place for a long time and just because you want to be Agile, don't expect the silo walls to crumble. In my experience I have seen many reasons for the silos, such as:

- Department heads not getting along
- Years of miscommunication between departments
- Departments making promises and not delivering to other departments

The best advice I can provide if you find yourself in a siloed company is to go ahead with the Agile implementation, but keep it limited to the Information Technology Department. Work with management regarding interdepartmental communications and if necessary hire business consultants to handle the silos. As you begin the Agile implementation, be the buffer for your project team and let them adjust to their new way of working.

The Project Team is the key to implementing a successful Agile program since they are the ones that will be experiencing the most change and doing the work.

The project team will experience Role Changes. Odds are good that before Agile the team really was not a team and that the developers worked on their own projects. Adapting to a team setting is a big change. Now we are introducing product owner, scrum master as the core team development team and in addition, testers, designers and the stakeholders. Big change. The best way to handle the role changes is to guide the team but let them self-organize.

One of the big complaints I have heard is that **Agile adds overhead**. The standup meetings, the testing, sprint planning, sprint demo and retrospectives. In my experience, once the team gets some experience under their belt they usually see the benefits and enjoy the comradery that comes with being a close knit team and what was once considered overhead is now all in a day's work.

Time-boxes can make the team feel pressured. Not many people like to work under time deadlines. But in Agile, time-boxes are a core component of the project success. Time-boxes will require a change in thinking for the developers. The time-boxes should be thought of as a toll to stay focused and produce working software quickly. Time-boxes will help the team prioritize what is most important at the time and avoid gold plating. After some time, most of the teams I have worked with have adopted time-boxing and have seen the value in it and it actually has helped reduce pressure on the teams.

The items discussed are some of the big hurdles in starting an Agile program. There will be other stumbling points but if you address the items discussed, your Agile program will be off to a good start. Starting an Agile program requires hard work and determination,

but in the end it can be well worth it for your organization as there will be many benefits and as your organization refines and becomes more Agile the benefits of Agile will continue to fuel your organization. I wish you the best of luck.

About the Author



Ralph Moore

Wyoming, USA



Ralph Moore is an experienced professional with more than 20 years in agile and traditional project management as well as roles in IT, Engineering, Telecommunications, Military, Public Safety and Education. Ralph is currently employed by a Wireless Telecommunications Company as the IT Project Manager in the Rocky Mountain region.

Ralph has earned a Master of Information Systems Management (MISM) degree with a Concentration in Information Systems Tools, a Bachelor of Science Technical Management and an Associate of Science Electronic and Computer Technology.

Ralph has earned several industry certifications to include: PMI-Certified Project Management Professional (PMP), PMI Agile Certified Practitioner (PMI-ACP), CompTIA Certified Security+, CompTIA Certified Network+, Electronics Technician Association Certified Network Systems Technician (CNST), Federal Communications Commission (FCC), General Radio Telephone License (GROL).

Ralph is a veteran of the US Navy and served three tours in the Persian Gulf. He maintained a TOP-SECRET clearance while in the US Navy. Ralph resides in Fort Bridger, Wyoming, USA with his wife Tracy and their two dogs. To contact – moore1746@gmail.com