Whitepaper
Knowledge sharing for your personal growth

IPROFS

Agile in a distributed environment: Local teams delivering a global project

Overview

Key Words
- Agile
- Distributed Environment
- Lessons Learned
- Project Methodology
- Scrum

Agile is being adopted by a number of companies to improve the chance of successfully delivering an IT project. Agile methodologies are being adopted by many companies, an ever-increasing number of these companies operate internationally. In a number of cases, organisations fail to see the need to tweak the Agile way of working to meet the context of delivering with a distributed team.

The aim of this paper is to provide some lessons learned from working on projects following an agile methodology, and Scrum in particular, with an external development team working for an international or global client. I hope it provides an overview of the things to consider and some tips on how to improve the chance of success.

Agile – conditions for success

Agile methodology calls for a close working relationship between business stakeholders who provide the requirements and the project team who deliver them. This can be incredibly effective, however there are some conditions which need to be in place:

> Agile project proceed on more of a 'learn as you go' premise where small working teams include customers and developers who are co-located and spending 100 percent of their time dedicated to the project. The work is done in increments, and quick iterations are continually evaluated and modified.[1]

The challenge is to compensate for the fact that more often than not the team is not physically in the same room (co-located) and are not fully resourced to the project, particularly on the business stakeholder side.

Good project sense

There are some things that you should always do on a project. You cannot afford to ignore these just because you are working Agile.

Model the business / project organisation, create and distribute a single org chart showing all teams. Make sure everybody understands the goals and high-level timeline, as well as who is responsible for what. Assign names and dates to actions during the meeting.

Everyone is not in the same room

On a project where teams are sitting in different rooms in different cities which are often in different countries, an effort needs to be made to compensate for this distribution.

Don't drown (each other) in email

Pick up the phone, use instant messenger (IM), collaboration tools (a project site, wiki, blog) to get in touch with people. This is a good way to build and maintain relationships with the team; a lot of important (and unrelated) things quite often come out when you phone to get one questions answered.

Conference calls

The project may succeed or fail based on how well you mange conference calls. People will need to have the right expectations

---

[1] "An eye for value: What the Business Analyst brings to the agile team", Kitty B. Hass, 2008.

and material before the call, otherwise you will burn a lot of time getting this sorted on the call. Make sure everyone knows what the goal of the meeting is. Use collaborative tools to make things easier, there are a number of screen sharing options out there (Microsoft Office Communicator and www.gotomeeting.com to mention a couple, and Skype has a number of plug-ins also.).

As with all meetings, make sure you take time to prepare for the call. Ensure the objectives are clear, set the agenda beforehand, assign roles (facilitator, presenter, note-taker), capture actions, and distribute notes (an email with actions and decisions is enough).

Get video conferencing
Invest in a web cam and some big TVs so you can see the people you are talking to. There are a range of solutions out there, and if budgets are tight then Skype is an option. When making the business case, remember that the investment is in infrastructure that will be useful for more than the current project, and that it will need to be maintained also.

Don't forget to have physical meetings
Regardless of the collaboration tools that are now available, "The most efficient and effective and effective method of conveying information to and within a development team is face-to-face conversation"[2]. The key is working out which meetings need to be face-to-face to ensure the success of the project; sometimes the motivation here can be political than anything else. Stakeholder buy-in and project sponsorship are still needed.

Longer sprints / iterations
In my experience a 4-week sprint (time-boxed development period) is just not enough, and I recommend a sprint cycle of at least 6 weeks. In large organisations, especially when not everyone is fully resourced to the project, it is not easy to get all the people you need together at a time that suits the project. I've been in situations where we have had to postpone 2-hour long call for several weeks due to stakeholder agendas; in a 4-week sprint this can really hurt your project. Either way, expectation management needs to find an extra gear.

Flexibility
Time to sign the team up for some yoga classes as flexibility is very important. You may have to host the same demo or meeting a number of times to meet time differences, shift in working day to compensate stakeholders in other time zones.

---

[2] "Principles behind the agile manifesto", http://agilemanifesto.org/principles.html.

Get a tool to manage the Product backlog

A Product Backlog captures all of the features that could be developed. Get a, preferably online, tool to manage the product backlog and include an owner from the project team and a business sponsor for each backlog item. 9 times out of 10, a spreadsheet is not going to cut it.

It will also help if you agree a process for managing the backlog with the Product Owner, and share this with the project. For example, agree who can do what in the backlog.

Scrum discipline

A Scrum is a daily stand-up meeting, and are designed to be short and to the point. There are three standard questions that everyone has to:

- What did you do since the last scrum?
- What are you going to do until the next scrum?
- Is anything stopping you do your work?

I recommend adding a fourth; Did you learn anything that might be of interest to others? This question helps combat the communication gaps that can arise out of working in a distributed environment. The danger is that people can quickly go off-topic. As Scrum Master (chairperson) you need to set the discussion frame clearly, and guard it well.

These scrums should be at the same time every day, and take priority over other meetings, they are not a luxury or optional extra. Keep the scrums tight, note the issues and address these after the scrum. 15 minutes is a good timeframe to aim for.

Multiple / layered scrums

Each team (here, I see a team based in a unique location) needs to have a daily scrum, with issues being discussed in a scrum of scrums which is attended by all of the scrum masters which has the aim of managing any blocking issues, and co-ordinating activities.

If a discipline is split over locations, each discipline should have a scrum to ensure that everyone is working in a coordinated way. For example, all developers independent of location (daily), or all requirements analysts (twice a week).

Not Everyone is fully resourced

"Business people and developers must work together daily throughout the project."[3]

I am including all resources responsible for delivering the project as developers in this context.

Business Analyst and product owner

This already critical relationship becomes more so in a distributed environment. It needs daily contact if the people involved are not in the same room, and as much face-to-face time as the project can afford.

Clear ownership and escalation paths

Different stakeholders may be brought in as they are seen as Subject Matter Experts (SME) in certain aspects of the business, for example Human Resources, however this may not stop them commenting on all aspects of the solution that they are exposed to before and during the demo. The way you deal with these comments needs to be clear to all involved, and more often that not the Product Owner will need to decide whether or not to go back to other business stakeholders to see about any changes.

It helps to prepare a short overview of the project timeline and processes to all stakeholders who get involved with the project after it has started and / or for a limited period (1 or 2 sprints). I suggest you introduce the concept of the Road Map and what this means in this presentation, as well as a simple overview of what has been done, and what is left to do.

Business stakeholders disagree with each other, and are unclear on what they want until they see the product working. User stories / wireframes / Product Backlog Items / specifications may not be enough. The demo is key to ensuring client acceptance and satisfaction.

It is equally key to have an established way to deal with demo feedback, either plan your sprint planning until after the demo feedback has been processed or set hours aside in the sprint coming up.

Take time to look back

"At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly."[4]

---

[3] "Principles behind the agile manifesto", http://agilemanifesto.org/principles.html.
[4] "Principles behind the agile manifesto", http://agilemanifesto.org/principles.html.

Due to relatively short sprints (typically 4 weeks), there can be a lot of time pressure on Agile projects. It is essential that the project takes time to look back and see how the process and way the project is running can be improved; this will win much time in the long run. Any changes that result need to be clearly communicated to everyone involved.

Conclusions

Agile remains a very strong methodology for "learn as you go" projects where new technologies or business competencies are involved. However, some problems are universal to all projects and deciding to work Agile, like every other approach, does not make these magically go away. Your project will have issues, business and technical challenges, and may even result in significant changes to the business. Just be sure you have a good way to handle these things in the context of your project.

I am going to finish with a request; before you start an agile project, ask yourself (and others): does it make sense do run this project agile?

Some questions that might help are:

- Do we understand what the client's desired outcome is?
- Do we understand the Business Architecture?
- Have we done this before?
- Do we have a clear and stable set of requirements?
- Do we have resources that are experienced in this technology?

If the answer to these questions is "No", Agile might just be for you.

Author
Roland Hill

Roland Hill is a Business Analyst with IPROFS, and a member of the International Institute of Business Analysis. He is very experienced in complex internet, intranet, and portal projects in commercial and e-government markets.

In the past, he has been a product manager for a content management system for the UK legal market, and an application and process consultant for a market-leading ERP firm.