Insights into an Agile Adventure with Offshore Partners

Mark Summers
Conchango, CSP
mark.summers@conchango.com

Abstract

This paper is an experience report from my time working at CampusSoft in the UK. The focus is on their experiences using Agile in a multi-site software development environment, spread across the UK, Romania and India. We will start by looking at the motivations behind outsourcing some of their work to India and why the relationship with their partners in India led them to try using an Agile approach. We will then look at some of the approaches which were important for them to be Agile and the challenges that they faced, such as communication, working practices and culture.

1. Introduction

This paper covers the experiences of a company called CampusSoft, based in the UK and their move to Agile, specifically regarding the impact on their offshore suppliers in India and Romania. This paper covers the time period from August 2006 when we first decided to make the transformation to Agile until July 2007.

CampusSoft is a company of around 2000 people, largely created by the acquisition of smaller companies who were leaders in their particular domain. Sections of the company are involved with the production of IT products. I was involved in one such section and the product we were building was for the education sector and involved a team of approximately 20 onshore staff based in the same office, 20 offshore staff based in India and 4 offshore staff based in Romania. This white paper covers that area of the business and how it developed with the introduction of Agile, the challenges we faced and what we learnt along the way.

1.1 Why Outsourcing?

Working with offshore partners was not a new idea to the team at CampusSoft. They had worked with an IT shop in Romania for many years and in 2003 we started to build a relationship up with a 3rd party supplier in India.

There were a number of reasons that CampusSoft sought the services of a partner in India. Some of the main reasons included:

- **Flexible resources**, it was seen as desirable to have the ability to easily expand and contract the number of resources, based on the amount of work currently in the program.
- **Skills available**, we had good people locally but it was always difficult to recruit the right people with the right skills to supplement those we already had. However India with its plethora of skilled IT professionals appeared to solve this problem.
- **Cost savings**, the cost of having resources offshore was more cost effective than local resources.
- **Professionalism,** a CMM level 5 partner such as the one we chose appeared to offer the controlled environment that would allow us to produce software to a predictable standard.

Central to the relationship with the 3rd party supplier was that they were viewed as an extension to the capabilities onshore, rather than a supplier with whom we created fixed price and scope contracts.

1.2 The Situation before Agile

Before Agile we had a traditional approach to building software. The business would decide there was a need to build a particular feature or undertake a module of work, an analyst would then write a detailed specification, which would be handed over to

somebody else to develop during a particular release phase. Generally nothing could be tested until the very end because it was a huge task to get the system up and running on a test environment.

When we initially started working with our partner in India, they were tasked with either supporting existing systems or adding small enhancements to some of our minor modules. Two of our IT team were assigned the responsibility of liaising with them. This fitted well with our existing way of working because the requirements were simple, the technology was well understood and we had people available to answer their questions. We needed to build on that initial success and scale it up so that they could get involved with our new development areas. At this time most of the development effort was around migrating our legacy systems to .NET, while adding business value along the way.

The first approach we used was to try to scale this directly. If I were to give our approach a name it would be "over the wall". New modules of work were written in requirements documents then handed over to analysts in India to flesh out into functional and technical specifications. The specifications were then sent back for approval, and then once the green light was given the team in India would start to build the new module. At a high level the project would always seem on track because we would get status reports about how the team was doing against the plan, but we couldn't actually see anything until we got very near the end of the project. Most of the output from this approach never saw the light of day. However it is clear why people thought this approach would work, as it was similar to our onshore approach and we had had some success on a small scale. Our development group at this time didn't really understand the value of the internal communication and interaction which supported these documents and also allowed us to build solutions onshore.

It was about this time in mid 2005 that I became team leader of this product development group. Together with the management team I devised a new approach for the next release, which would take the next ten months and add two new modules to the .NET product suite, curriculum and exams. We realised that we needed to be able to give feedback much earlier, so we decided to break the development down into iterations or a number of milestones. Our general strategy was:

- The first two months we would build most of the main screens, with nothing much behind them. This was very easy with the architecture that we had built.
- The second two months we would build the CRUD functionality behind these screens and start some of the more ad-hoc screens.
- Then for the final 4 months of the build we would do everything else, including creating all the complicated business logic.
- Finally we would have about two months to test it all.

We also split the teams, so that there were developers for each work-package both onshore and offshore. This gave the India team a colleague with whom to work. The big draw back with this approach was that this role rested with the people who knew the product and the technology best and who had previously been the most productive. Now, they were reduced to co-coordinating work for the offshore people. However it was the only way that we could see to facilitate the development of the offshore capability and at this stage we were still keen to invest in the offshore people.

With this approach we were able to make sure we were roughly going along the right lines in what we were building. It now looked right however when you dug deeper into the functionality it did not do what we expected. We still ran into a lot of quality problems, and work was still always slipping later and later, so that all the way through the project we were de-scoping work.

1.3 Why Agile?

We were moving towards last chance saloon with our hopes for having success with our partners in India. Morale was low with the onshore team because their productivity had been slashed. Therefore we decided to try something different, Agile. The main reason we were prepared to look at Agile was that we needed a way of breaking the work up, so that the offshore teams could deliver small chunks which we could inspect and quickly feedback and get them to correct if it was wrong.

We chose Scrum, because other parts of the organisations had reported having some success with it and it seemed to fit the bill with its short iterations and focus on delivery.

2. The Transformation

We started by creating two onshore Scrum teams, each set up to work on different modules of the system. There was about five days from going through the concepts behind Scrum and starting our first Scrum Planning meeting. The team had very little experience with Agile. The Technical Architect who introduced us to the concepts of Scrum had experimented with his small architecture team.

The rest of the team started Scrum with no experience. I worked with our Product Manager, to get our existing requirements into our prioritized backlog. I then facilitated the planning meeting for the first two teams and they started their respective four week Sprints, with me as their ScrumMaster. Even with this limited experience we had some success, the business was really pleased with what had been produced in four weeks and the software already seemed a lot more stable. We were certainly learning how to work together in teams, rather than as a group of individuals.

For the next Sprint my Project Manager became ScrumMaster of one of the onshore teams, while I concentrated on getting the offshore team working in an Agile way, which was the reason for choosing Scrum in the first place. The curriculum and exams modules that we had worked on before going Agile were in place but were not yet in a state with which we were happy. The user experience didn't flow well, they didn't perform as we wanted and there were lots of system crashes. We solicited inputs from various parts of the organisation and put the most important things we knew about into a Product Backlog.

2.1 Assembling a Team

While we needed some of the people offshore to support our existing systems, the rest were going to be part of a Scrum team. The team became known as Team Delta. Team Delta would work towards making the modules to manage curriculum and exam data solid, this would allow us to move more of our customers across to our new .NET platform, and thus reduce the cost of still supporting the legacy system.

Most of the Delta team had previously been involved in the development of these modules, so they knew the area. The offshore team was made up of developers, one tester and the ScrumMaster, with me onshore to coach them and facilitate things from this end. The Scrum Team would also consist of a business analyst and a tester, who would be onshore. Both these people had many years of experience in the business domain and because of their skill sets, were an excellent conscience for the team. They were also there to provide quick answers to most business questions and because they were closely plugged in to what the business wanted it ensured the team wouldn't go far off track. There was one overall Product Owner for five Scrum Teams. He couldn't be present at all of the meetings; therefore we had to scale the Product Owner by having the business analysts take on this responsibility at the team level. To coordinate, they met regularly with the overall Product Owner, who ultimately still prioritized work and made the higher level decisions.

We communicated our desire to move to Agile and the offshore partner had specialist trainers come to help them understand what Scrum was all about. The Product Owner also made a number of trips out to visit the offshore team to help align them with his vision.

3. Agile practices with the offshore teams

3.1 The Product Backlog

Items at the top of the Product Backlog for Team Delta were fairly simple to start with, it was full of bugs and small changes that were needed to bring the product in line with what was needed. This certainly helped the team get going, as it made it very easy to have sensible conversations and made planning easier. This meant that the team members who were onshore spent most of their time either answering questions about the current Sprint or working ahead on the Product Backlog.

Working ahead on the Product Backlog to make sure it is just good enough for the planning meeting is important for all teams and their Product Owners. However we found that just good enough required smaller chunks with more detail than you would normally aspire to with an Agile team. This is what the Delta team themselves discovered at one of their retrospectives and acted on it to improve the situation.

At about the time we set up the offshore team, we started using User Stories, so any new requirements that we produced would be written in this way. This helped with the offshore team, because this format of requirement forces people to have conversations. We had for too long tried different ways of writing down requirements for them and each time they would be misinterpreted. The conversations also helped develop relationships between the business and the offshore team members.

3.2 The Planning Meeting

Planning is key for any Agile team, as it is the activity that sets up the whole iteration. We wanted to make it feel as much as possible like one team in one room planning. For most planning sessions we were able to use video conferencing facilities. Unfortunately there were times when we either couldn't get the right room at one end or we had problems with the technology. When this happened it really seemed to make planning much more of a struggle. It really helps if you can see the whites of somebody's eyes when they are trying to explain why something might be more difficult than everyone else thinks.

When estimating, having somebody facilitating at both ends helped. Typically I would lead by getting the Product Owner to talk through the story, and allow for the team to discuss and ask questions. We found it valuable to use planning poker as it helped to get people talking about what was required. The ScrumMaster at the other end would say when everyone was ready to vote and the team would turn over their cards at the same time. Having somebody facilitating at both ends helped the team get into a rhythm and allowed us to coach them through the process. After a while the team started to do this naturally anyway and the need for two facilitators diminished and I was able to let them get on with the planning on their own by the fourth or fifth Sprint.

In addition to that we always made sure we had a WebEx session or later we used GoToMeeting, so that we could share desktops. This would allow the Product Owner or anybody else to go through the product to help explain what was being described. It was also useful to share the tool, which we were using to manage the backlogs.

3.3 The Daily Scrum

As a ScrumMaster, as soon planning has finished I like to create a wiki page describing what the team have committed to, who is in the team, the logistics of the review meeting and when and where the Daily Scrum will be. Even more importantly when a team is distributed, the name of a contact person needs to be made obvious should anybody wish to attend the meeting as a chicken.

As the 15 minute Daily Scrum happens every day you don't want to be spending time setting up the video conference every day or organizing a meeting room. Therefore a Skype call was usually preferred, and we made sure everyone had a headset to make it easier. With a distributed team we found it better for everyone to remain in their seats where they could access the Sprint Backlog, which could help give focus to their discussions.

3.4 The Sprint Review

The Sprint Review was conducted in a meeting room by sharing somebody's desktop, so that the team could participate equally with the stakeholders. This would be projected onto the wall both onshore and offshore. There would also be a conference phone at both ends, to allow the two groups to talk to each other.

Team members at both ends would demonstrate functionality, however if possible this would be done by the team members onshore, as it made it a richer experience for the stakeholders and the Product Owner. The Indians in this team were very softly spoken and not very confident when presenting, so people naturally switched off when they were demonstrating.

3.5 The Sprint Retrospective

We never got the Sprint Retrospective right for the offshore teams. When the meeting happened, it was done over the phone at people's desks. The team would talk about what went well, what didn't go so well and what they wanted to change for the next Sprint, but would not really using engaging activities to help drive the discussion.

3.6 Working with Others

The way the onshore part of the team supported the offshore team members was working well, although it did make the two onshore people in this team feel somewhat isolated from the others in the development group as a whole.

There was a big change in how we encouraged the offshore team to communicate with other people; both inside development and with other departments such as support and professional services. These other departments faced customers all of the time so they were a great source of information. Before Agile others in development had acted as the interface between the Indian people and these other departments. More and more we introduced people, coaching them in the benefits of having a direct and open dialog. This helped reduce the information that was lost in

translation and allowed the rest of us to add value in more efficient ways.

3.7 Integration

Before we even started with Scrum, we had an enlightened Technical Architect who had a pet project to set up continuous integration, for which he also roped in some other equally enlightened people to help sort out an automated build. This made it very easy for us to start using Scrum, because anything that was built one day could be deployed onto the test environment the next. At first there was a lot of manual intervention needed on the build but over time we automated more and more of the process. Everyone doing new development including the offshore people checked in to the mainline branch and everything was built over night onshore, including our legacy systems. This included all five teams integrating their code at least daily. The build could then be deployed to the system test environment the next day, which all the teams would use for any manual acceptance testing that they needed to do.

As soon as a developer checked anything in it would trigger a build, and the developer would be notified immediately if he had broken anything. This helped focus the developers on building the quality in before they checked in. This worked with the offshore teams because everybody checked into the same branch in the source code repository, and anybody who broke the build would receive an email no matter where they were located.

3.8 Testing

We had a reasonable set up for integrating our code when we started with Agile. We struggled to get teams writing automated tests. The architecture didn't help because it had been developed pre-agile and we had spent a long time building an architecture platform before we started to actually build the application. I have kept in touch with some of the developers from CampusSoft and they have started in some teams to automate their acceptance and unit tests, even to the extent of writing the tests first. So they are still on the path of continual improvement.

When I talk about testing here, it was manual testing on the system test environments that we could deploy first thing in the morning. We started to take advantage of the fact that the team in India began work five hours before we did and so often they kicked off the deployment remotely on our environments. It was also an ambition to automate this deployment process, and it was always fairly near the top of the Product Backlog, but alas it never quite made it into a Sprint.

The great thing now was that developers and testers were on the same team and working together towards the same goal. This created a much better working relationship. I can remember previous occasions where I would have stand up fights with testers because they were saying something was a bug and I was arguing that it wasn't worth trying to work around the limitation of the technology. Anyway these days were in the past, we now had developers and testers talking through problems together, and then discussing the trade-off decisions with the Product Owner.

The offshore team worked in the same way as all of the other teams. If a problem was found with any of the User Stories that caused it to fail its acceptance tests in the current Sprint then this was viewed as something else that needed to be done to complete that User Story. If this was found by one of the onshore people, it would usually result in a Skype conversation as soon as possible to discuss the problem. Often a new task would be created against the Sprint Backlog item. If a problem was found outside the scope of the current Sprint, then often it would be fixed as the developer found it. However if it required a bit more thought then it would be discussed with the Product Owner and if it was important it would end up in the Product Backlog. With the offshore team, because of the extra overhead in the communication and extra cost in task switching, most things out of scope would go into the Product Backlog.

4. Challenges

4.1 Culture

We had a contrast of cultures in our development group; we had the onshore people, and people in both India and Romania. The staff in the UK and Romania were more closely aligned in their culture and generally adapted quite well to working in an Agile manner. We found it more difficult with most of the Indian people, and it probably didn't help that they were quite young. They were always willing to say yes to most requests, for example "given what you have accepted into the Sprint so far do you think you can do this next item as well?" A few worried looks later they would say, "Yes". Therefore we had to learn that if there was a pause between us asking a question and them saying yes, it usually meant there were some concerns. Rather than just accepting it and thinking

"great I can have that feature as well", we would drill into why they thought they could do it as well.

The Indian culture is a hierarchical one, and as we were more senior than them, it was very difficult to get them to challenge us. We really wanted them to be able to engage in some challenging discussions about how we approach problems and about what is achievable. We never managed to get this interaction working, which was one of the reasons we felt that we could not build anything complicated with them.

4.2 Communication

Communication was a big challenge. You can't have face-to-face communication with a distributed team all of the time, but you need to get as close to it as you possibly can. Tools like video conferencing were vital for us to achieve results in the planning meeting when high bandwidth communication was required. Using an Agile approach got us discussing our problems early, rather than storing them up. It took us a long time to improve this aspect, as the Indian team members didn't feel comfortable airing their dirty washing, but it got a bit easier as trust developed between the team members.

The biggest barrier to effective communication was the cultural differences, and we probably could have done more in this area. Before Agile we had invested in bringing offshore people onshore for a time, but whenever we did this they always seemed to move on shortly afterwards. Therefore we were reluctant to make similar investments. However, given experiences since this period I have found the best way to break down the cultural differences and build relationships with the team is to have them come onshore for the first few Sprints or alternatively have the onshore people including the business go offshore.

4.3 Working Practices

We tried to employ consistent working practices both onshore and offshore. We used the same tools, to manage the backlog, shared a wiki, had a bug tracking system that was shared and also used by front line support. Development environments were set up the same and we had quick guidelines for doing so on the Wiki. Most importantly we all checked code into the same code repository that was built in one place; so we were never that far away from the truth. One big consideration before selecting a global partner has to be ensuring that you have the bandwidth to support one development environment.

4.4 Single Vision

A single vision is difficult to establish especially with multiple stakeholders and where you have had to scale the Product Vision. At the start of each release our Product Owner would travel out to India to try to immerse the offshore teams in the vision for the release. Therefore it is very important that you have a Product Owner who can engage with the team, and take them along the journey with him. We had had many previous instances of the offshore people not understanding what the business wanted, so having this engagement upfront really started to help. The Product Owner and other stakeholders from the business have to be willing to then answer questions and engage throughout the project. This all requires a total change in ethos from the business perspective, and this can take time. It only happens for real when they start to see the benefits, or to put it another way, when the benefits become tangible to them.

5. Conclusion

The culture aspect was always the biggest challenge for CampusSoft and its outsourcing partners. The Indian team members were always reluctant to engage in constructive disagreement. We got around this in some aspects by chunking the work up small and having experienced domain experts communicating with them daily. There were a number of really key things that we worked on over time. How we communicated in the various meetings, and how the ad-hoc communication worked within the team. Having a single vision that both onshore and offshore people bought into and investing in communicating that vision to those offshore. We had common working practices across the team with identical development environments; everyone checked code into the same place, and there was only one build. Therefore, the earlier you can get these challenges addressed when working with offshore partners the better off you will be.