How to ORGANIZE OFFSHORE and nearshore collaboration

lessons learned in offshoring and nearshoring

BRIDGE

www.bridge-outsourcing.nl

HUGO MESSER

DARELL CULLEN | HERKE SCHUFFEL | ANDY JORDAN | ERWIN DE BONT | ANUJ KUMAR
ANDREAS BRILLING | JEAN-PAUL VAN WIERINGHEN BORSKI | ABHILASH CHANDRAN | HENK WOOLSCHOT

Table of Contents

Foreword

Sign up at http://bookoffshoring.bridge-outsourcing.com/ and be the first one to know when the next eBook on managing remote teams is released

This is the third eBook in a series of eBooks that will be published within a couple of months’ interval and later on into one printed book. These eBooks are being written through a crowdwriting project and the authors are experts from all over the world.

We welcome any suggestions or feedback for further improvement. If you are interested in the upcoming eBooks or are an experienced practitioner who would like to contribute with your knowledge, please e-mail : h.messer@bridge-outsourcing.nl

Introduction

How do you organize a remote collaboration? What process should you introduce? How do you communicate your requirements? How do you ensure that people are doing what they are supposed to be doing? These questions come up when managing teams that are geographically distributed. Nowadays, the technological infrastructure is in place and there are many companies working with globally distributed teams. At the same time, a lot of people look for a proven way to organize remote work. In this eBook, eight practitioners from different parts of the world and from different organizations share best practices based on their experiences.

In the first chapter, I discuss three ingredients for a successful remote collaboration – process, responsibilities, and performance. Thereby, the crucial questions such as 'What process works best? Should you use scrum, waterfall, or something else?' will be answered. As far as process is concerned, I describe the importance of creating clear responsibilities among the team members. Once the process is in place and people know what to do, you need to measure the outcome.

The second chapter is written by Darel Cullen who has a long and varied experience in the software industry in various sectors including nuclear, space, air traffic control, and other industries. He provides guidelines on knowledge transfer and IP protection. Many companies struggle with transferring (tacit) knowledge to their team members at a remote location. The remote team lacks the interaction that a collocated team has. Darel shares some best practices on reducing this knowledge gap.

The third chapter is written by Abhilash Chandran, an Agile Software Development Manager, coach, and practitioner, employed with Xerox Corporation. He shares his experiences on setting up a distributed Agile team in India. His expertise gives you practical insights on what roles should be performed onshore versus offshore, and what tools can best support the Agile collaboration.

How do you discuss, manage, and realize planning and deadlines? In chapter four, Andy Jordan introduces the steps to success in planning your offshore project. Andy is the President of Roffensian Consulting Inc., an Ontario, Canada-based management consulting firm with a strong emphasis on organizational transformation, portfolio management, and PMOs.

In chapter five, Erwin de Bont, who has over 20 years of experience in successfully managing many aspects of the Telecom and ICT Industry, shares his insights on good governance and multi-level KPIs in outsourcing. He uses the example of a large organization that wants to outsource many primary processes to help their customers.

In chapter six, Andreas Brilling and Anuj Kumar describe their concept on 'how to make the waterfall model work in a multi-shore setting'. While most organizations move to Agile and scrum, the authors describe a successful implementation of waterfall and how to add Agile elements to the waterfall process. Andreas Brilling is an Engagement Manager at Capgemini based in Stuttgart. He has more than 20 years of experience in software development projects in various international settings. Anuj Kumar is a Senior Manager with Capgemini, India, based out of Mumbai. He has been working with custom software development-based projects for the past 14 years. Most of his projects involve multi-shore teams.

In the seventh chapter, Jean-Paul van Wieringhen Borski and Herke Schuffel describe the differences between offshore and nearshore collaboration, and provide suggestions to deal with them. They have developed a framework for organizing nearshore collaboration differently from offshore. Jean-Paul has 17 years of experience in the IT domain. He has spent most of his time in an international environment where he managed several offshore delivery centers. Herke has been in the IT domain for almost 20 years now. He is currently responsible for a business unit delivering application support on custom-built software with extended resource teams in Serbia and Ukraine.

In the final chapter of this eBook, Henk Woolschot, Delivery Manager at HCL, shares his insights on partnership. He draws the line between two kinds of collaborations and gives us a clear picture of when one should prefer a ‘client-supplier’ relationship over a real partnership.

Hugo Messer

Chapter 1 - Working Together, Sitting Apart

By: Hugo Messer

Introduction

There are essentially two factors that determine whether your offshoring adventure is successful or not – people and process. Some people will claim that process is the most important factor, while others will argue that people are number one. When I started with nearshoring about 8 years ago, I believed in process. I believed that offshoring would work by building a factory-like process. Today, I am convinced that although process is crucial, people are primary. Use an excellent process with the wrong people and the outcome will disappoint you. Hire the best people available without any process and the outcome will surprise you. If you do not hire the right people, they may not even think about process and instead just 'get going'.

The aspect of 'people' will be covered in depth in one of our soon to be launched eBooks. In this article, I will share my experience in developing a rock-solid process for remote collaboration. As people sit apart in (several) remote locations, extra attention must be paid to articulating how people work together. At Bridge, we have developed a workshop that we always provide for new customers. The chapters in the work book that we have created are the key ingredients to making a remote collaboration effective:

1. **Process**: How do we work?
2. **Responsibilities**: Who does what?
3. **Project Management Too**l: How do we track projects?
4. **Communication**: How (often) do we communicate?
5. **Performance**: What and how do we measure performance and progress?
6. **Coding standard**: What is our standard?
7. **Time registration**: How do we track time?
8. **Creating 'Co-workers'**: How will we perform together and become friends?

I will discuss three of these ingredients – process, responsibilities, and performance. You can download a free copy of the workbook on our website.

1. **Process**

The natural inclination of most people is to 'just get going'. They select a partner in Eastern Europe or India (spend a lot of time on RFIs and RFPs) and then 'get to work', 'start the project'. Especially, if the outsourcing relationship is based on fixed prices, the expectation is that the partner knows how to handle the project. Requirements are made, sent to the supplier, and now we sit and wait till the results are delivered on the agreed date. But, will this work?

Of course, as an outsourcing 'buyer', you should expect the outsourcing partner to know how outsourcing works. The remote team must know how to serve you well and make your project a success. And it is likely that they know all this. But do you?

And do they know how to make it work specifically for your situation and your project? Many outsourcing vendors are by nature technology companies. They are good at building stuff. But that does not mean that they are able to make a complex collaboration across cultures, time zones, and languages work smoothly.

Before the offshore team starts executing the project, before they start analyzing your requirements, or writing the code, it is crucial that you sit together with the team to discuss how you are going to work. The starting point is the way you work right now and/or the way the vendor works. Hence, two approaches are meeting here – the way you work and the way the vendor works – which means that one of you has to adapt and change behavior.

Software development is a creative process. Just pouring requirements in to a standardized process to get a finished product, would be a dream come true for any customer. But this is not the way a creative process works. In the past, software development followed the construction approach and people developed the waterfall model. This model has defined the way of working for about two decades. Today, everything is about ‘agile’. A creative process is agile by default. In the rest of the article, and to keep it simple, I will use waterfall and scrum as the two extremes on a spectrum of processes.

How do you work today?

Are you following a method that resembles the waterfall model or are you maybe truly agile? Do you use scrum as most companies do? Do you apply extreme programming or Kanban perhaps? At Bridge, we build remote dedicated teams that supplement an internal IT department. I have learned that the most challenging customers are service providers (as opposed to companies building a product). Especially, if you are a small service provider like an internet agency, it is extremely challenging to make offshoring work. You are, in most cases, forced by customers to work for a fixed price. Probably you provide this fixed price by using a waterfall model. If you are a bigger service provider, it is quite another story. As a big service provider, you are basically building ‘products’ for your customers in long-term projects. Therefore, you have more time and flexibility to build a process that produces the desired outcome.

Now, please take some time to become clear about how you really work. I myself have experienced that scrum is based on some principles and not on rigid processes. This is good but it also reveals that you might be following scrum while you are actually using waterfall or Kanban or something similar. If you want to establish a smooth, integrated way of collaborating with your remote team, you must clearly tell them how you work. You have not written down or drawn the steps you take or the principles you use, or the artifacts of scrum you do or do not use. People have built a way of working which your current team understands subconsciously. However, the new remote team also must consciously adapt to the same way of working and routine. So commit it on paper, draw it, and make it explicit. If you use only verbal ways to communicate how you work, it is likely that you may not see or understand them the same way once you get going.

How do you estimate the work?

A crucial challenge in the waterfall model lies in the estimation of a project. Let’s look at your personal planning. How do you personally plan your week? If you look at the priorities you have for the week, are you able to estimate how much time each will take accurately? If you do that estimate your work for the upcoming month? For the upcoming quarter? To identify the accuracy of your estimation, compare your approximated time with the actual time taken to complete the tasks within the estimated time period. You are probably off track by 2-3 for the week and 5-10 for the quarter.

We expect software developers to come up with good estimates and we invent all kinds of methods to make them more accurate. But, do we even get close to reality? Normally, projects for your customers need to be estimated in advance. Therefore, a simplified calculation is often being used to speed up the bidding process. Given this fact, one must not overlook that after you have won the bid, at a certain point of time, the offshore team is supposed to estimated the workload too. . And it is very likely that they are off track from your estimate (and you were off track already anyway!).

In case you are working this way and cannot change the waterfall model, it is advisable to find an offshore partner who uses the same process – a partner who is used to working on a fixed price and whose process matches with your estimates. I recommend that you should change the process if possible. You can start by introducing some elements of scrum to your current process and changing the process step-by-step. Take the time to introduce agile principles and scrum, train your team on scrum practices, and get some of your employees certified as product owners or scrum masters. Implementing this change will take a few months, and can be established with the help of your offshore partner before starting projects with your remote team.

In chapter 6, the authors describe ways to make the waterfall model work with a remote team for bigger projects.

If you are already working with agile or scrum, that is great. You probably use planning poker (or another variant) to get an estimate of each user story that the whole team buys into. And if you need to estimate an entire project, you can roughly estimate the scope and use the flexibility of scrum for adjusting the workload per sprint to eventually fit the scope. The ideal case is of course to have the whole team, including the offshore team members, join the sprint planning and commit to the scope.

How do we define requirements?

One of the most challenging aspects of software development is to know what needs to be built, and to express this through clear requirements. Also add the distance along with different time zones, languages and cultural differences to this, and the challenge quadruples.

The waterfall model is based on the assumption that we can figure out upfront, what to build. I believe that if we spend a lot of time, indeed we can, but only with a probability of75-80 percent and thus deal with the risk of building something that is outdated by the time it is ready. If we choose this path in a remote collaboration, it is essential to establish clear guidelines for making requirements or specifications. Some questions that must be addressed are:

- Which details are essential for the specifications?
- Do we create separate technical specifications? If yes, which specifications are essential?
- How will the offshore team be involved in developing or extending the specifications?
- How are additions to the specifications discussed, agreed upon, and stored?

The answers to these questions must be committed on paper, so that each team member has a clear understanding of how the requirements are being set. As a supplement to these guidelines, it is helpful to attach an example of the ‘ideal specification’.

The most fundamental question is 'who creates the specifications?'. The natural inclination of people is to create the specifications onshore because it was closer to the customer. But think about this carefully. When committing something on paper, you give extensive thoughts to it, based on many discussions you have had with your customer before. The remote team, however, is being excluded from this interaction. So you are sending requirements that are clear to you, and a mystery to your remote team. The earlier you involve the team in developing the solution and the requirements, the easier it will be for them to understand the process and what needs to be built.

Instead of specifications, discussions and meetings are the starting point in a scrum or agile process.When it comes to specifications, not only one person, but the whole team is engaged in developing demands.

First, the product owner speaks to the client or user, translates the requirements into user stories, and then the scrum master and the development team (designers, coders, testers, architects) further specify the solution for each user story. It is important to clearly understand how this process works in your company.

I have observed that sometimes companies call it ‘scrum’, even though they are not really following this method. In this case, for example, the product owner (or a person who is both owner and scrum master) would talk to the customer, describe the user stories in detail, prescribe the solution-sometimes up to the level of the technical implementation - , and then send the ‘user stories’ to the development team. If you handle it like this, you have created a mini waterfall model. Although this approach can work, it is crucial that the remote team and you are aware of applying this model because it has an impact on the level of involvement and clarity in understanding your requirements.

How do we create a plan?

Since this eBook contains an entire chapter on planning, I will only highlight the important questions.

If you use a waterfall model, you have probably developed a standardized way of creating project plans using tools like Microsoft Project. Thinking about the way you make plans, helps you providing an explicitly written concept for your remote team. Are you going to involve the offshore team in the planning? Will they create the designs and get them approved by you? Will you create the plans for them to execute?

In scrum, are you performing the sprint planning carefully? Is the whole team involved in the planning? How do you calculate the hours available in the sprint? Are your people able to work productively on user stories 80% or at least 50% of the time? Who commits to the sprint?

Who’s testing what?

Testing is a grey area in software development that is never organized in the same way in every company. Most companies expect their software developers to test their own work (but, what is ‘acceptable’ in that case?), after which a project manager does some final testing before sending it to the end customer or user. Larger teams have a separate tester in their team or a testing department which evaluates the features or user stories once they have been completed by the developer. Some scrum teams follow peer testing where one programmer tests the work of the other. In addition, some work with group testing where the whole scrum team gathers at one PC on the final day of the sprint to test together.

The important part here is to document your current testing process in the organization. What do you expect from each individual? How are you going to adjust the way you test the new situation with remote team members? What are the acceptance criteria or the definition of 'done'? Write everything down, discuss it with the remote team, and get a clear mutual understanding on how testing has to be done.

2. Responsibilities

If you put a team together and let them ‘get started’, they may or may not somehow manage to complete your projects. In my view, it is crucial to discuss what each person is responsible for and write it down before the team starts working.

The best starting point is at the company level. What are the responsibilities on either side? The primary aspect, which is relevant, is the project responsibility. Who is accountable for planning and setting deadlines? Who does the project management? A clear demarcation line (also contractually) can be the key to answer all these questions. Some other factors are: Who takes care of people management? Who ensures that every 3-6 months the team members discuss their personal development with their manager? Who decides on and organizes training?

In scrum, the most important roles to ‘assign’ are the product owner and the scrum master. In the waterfall model, these questions are similar, although the roles are named differently.

The most efficient way to distribute the responsibilities in an offshore setting, is to have the product owner onshore, as close to the customer as possible. There can be a second product owner offshore, even though this may complicate things if your project is relatively small. The scrum master can be either offshore and/or onshore. The variants are:

A. The onshore product owner talks to the offshore scrum master.

B. The onshore product owner talks to the offshore product owner.

C. The onshore scrum master talks to the offshore scrum master.

D. The onshore scrum master talks to the offshore development team.

Ideally, the customer or end user talks to the whole team in the sprint planning sessions or even better, sits in the office ‘with’ the (virtual) scrum team (either onshore, offshore, or mixed). In most cases, however, the product owner talks to the customer and communicates all the user stories to the team. No matter what setting you choose, the whole team (including the product owner and scrum master) uses video conferencing for the sprint planning.

Some questions to be answered to create clarity in the responsibilities are:

- Who is responsible for describing the implementation of the user stories (functionally and/or technically)? Is it the onshore or the offshore scrum master?
- How much can the product owner describe?
- Who is responsible for making the decisions on what and when to build during a sprint? What if that person is not available because of time differences?
- Who is responsible for giving a demo?
- To whom should the demo be given?
- Who conducts tests on the functionality?

In order to be able to complete your list of responsibilities for each person and/or role, I will continue with similar questions:

One crucial role in an offshore collaboration is a 'process manager' (also called delivery manager, quality manager, or similar terms). This person's responsibility lies outside the project and his core accountability is to ensure a smooth communication between the onshore and offshore teams. In a larger setting, for instance, it is even good to have this role both on the onshore team and the offshore team. These process managers discuss on a weekly basis and evaluate the current performance. As they are not involved in the day to day details of the projects, they keep an outside view and can provide valuable feedback on the way the team communicates; they can review the process to check if you are still on track; they can safeguard contractual agreements on project responsibility, etc.

3. Performance

For large enterprise settings, the chapter of Erwin de Bont provides you with a framework for measuring performance and KPIs. Here, I will describe some simple instruments on the team level.

On the company level, there is basically only one question: Is this collaboration delivering the value we want? In Bridge, we ask this question every week back and forth. It is not a one-way street. The onshore and the offshore team must be comfortable with the collaboration. The goal is to create a partnership, to collaborate, to create synergies, and 1+1=3. After all, this is what it is all about. Of course, you can also measure response time, productivity, or other variants that are linked to your SLA (see chapter 5)

The same question can be asked on the team level where there are several teams working on different projects with different project leaders.

The most crucial performance to be evaluated is on the individual level. Teams consist of people, so we need to have someone to assess each individual's skills. It is best if the scrum master (either onshore or offshore) does this assessment once a month or quarter. In Bridge, we ask our customers to rate individual team members every month. We use a ranking of 1-10 to keep things clear. By default, we use the following set of questions (depending on the importance of a certain area for the project, we modify the questions):

Is the programmer:

- Meeting your expectations?
- Responding appropriately to the tasks you assign?
- Effective in making decisions on his own?
- Understanding the economic model of the product/project?
- Open to receiving feedback?
- Completing tasks within the agreed timelines?
- Apt for your future projects?

Along with these ‘formal’ performance evaluations, a meeting rhythm must be established. If you follow scrum, you can provide team feedback in the daily, demo, and retrospective meetings. We also have a weekly meeting with the process managers (see previous paragraph) in which each ‘shore’ can give a weekly rating on the collaboration as described above. Once a quarter, we meet personally with the individual, our HR manager, and the process manager. In some cases, our customers join or take over this quarterly personal development meeting.

Conclusion

To lay a solid foundation for a remote collaboration, it is important to create ‘think time’ before ‘doing’. Before you start your project, think about the way you work today and how you will modify this way of working with (part of) the team offshore (process). The topics that you could cover are:

- What steps do we follow?
- How do we arrive at estimates?
- How do we list our requirements?
- How do we plan?
- Who tests what?

You can discuss these questions in a session where both the onshore and offshore team members are present. During or after the meeting, commit the process on paper. Describe the **responsibilities** of each person involved, up to the last detail. And finally, invent a solid framework for providing feedback and rating to each other, the company, teams, and individuals.

And now, you are probably thinking: "Great! We have nailed all these things down, discussed them for hours, written them down, and they will end up in the (virtual) drawer. Nobody will ever look at them again and we will get going.” Yes, probably you will do so, because that is a natural tendency.

Nevertheless, the team has used their brains to think about ‘how’ to collaborate before actually doing so. This creates bonding and in itself creates a more effective way of working. Now, it is your responsibility to keep this document out of the drawer, use it in your meeting rhythm, and update it as you go, so that it becomes a living plan or practice!

About Hugo Messer

Hugo Messer has been building and managing teams around the world since 2005. His passion is to enable people that are spread across cultures, geographies, and time zones to collaborate. Whether it's offshoring or nearshoring, he knows what it takes to make a global collaboration work.

To know more about Hugo, check out his website http://www.hugomesser.com. You can also read the blog or watch videos at Youtube.

About Bridge Global IT Staffing

Bridge Global IT Staffing offers western software companies an opportunity to work with IT talents from their offices in India and Ukraine. The personal support, offered from the European offices in the Netherlands, Germany, Switzerland, Sweden, and Denmark, makes it easier for clients to manage their colleagues from a distance. Since there is both an offshore and a nearshore office, chances are high that Bridge has the talented IT employee you are looking for. If not, the perfect candidate will be found for you.

Website: http://www.bridge-outsourcing.nl

Chapter 2 - Knowledge Transfer and IP Protection

By: Darel Cullen

In the early phases of setting up a remote development team, there is a stage an organization goes through where existing on-site staff will have to share their knowledge with other individuals and teams. This is often known as a knowledge transfer, an activity performed before a remote team can become productive.

There is knowledge that your organization knows it has - explicit knowledge, and there is also knowledge your organization is not aware it has, but actually does have. This type of knowledge is known as tacit knowledge. Sometimes tacit knowledge can make it difficult for receiving teams to become fully productive. One way to handle this is to perform knowledge audits or create knowledge maps before knowledge transfers, to uncover areas that need to be investigated and transferred.

At the start of a knowledge transfer, there is an intensive period where people start learning about each other and where teams are located offshore; there may also be cultural differences and language difficulties that quickly become apparent. It is very important that you have employed competent staff at your receiving center, who are technically capable of learning the knowledge you wish for them to receive. Common language skills (both spoken and written) should be at a satisfactory level, and therefore, it is important to check these skills at the beginning.

At times, you may face resistance from the sending organization; so it is vital to highlight and agree on the reasons for the knowledge transfer, and provide enough motivation to the sending organization to share their knowledge. Very often, offshoring brings benefits such as transferring over “legacy” products, freeing up time and resources to work with new technologies and developing new products, or perhaps even opening up access to the local market. Be open and honest about the rationale behind the move; if your own staff does not support the knowledge transfer, it will more often than not fail.

It has been proven that during knowledge transfers, the overall productivity decreases, so be prepared and plan ahead for such a period. Productivity loss happens irrespective of the destination (offshore, nearshore, or onsite), and the time taken to completely transfer knowledge is often longer than one might expect. Depending on the domain, it can take many years before a remote group reaches the level of expertise and skill of the sending group. The other challenge is maintaining a good and consistent quality after the transfer, when individuals start applying the knowledge they have learnt.

Often the best way to share knowledge is giving simpler tasks to the receiving organization in order to stimulate asking the right questions and raising levels of confidence. Very frequently, receivers of knowledge don’t know what they need to know, and senders may also be unsure about what knowledge the receivers already have.

Some recent academic findings are that both organizations should receive cultural training early on in order to avoid misunderstandings and to build an awareness of each other's cultures.

A recurring theme in many articles in this eBook is that teams have to meet each other in person and build a rapport and trust. So ensure to budget time and resources for extended periods of travel (at a minimum, several months seems to be a reasonable time frame).

Something that's often missed during this phase is that the new staff can sometimes suggest better ways of working; so try to capture these ideas and act on them. This also boosts the motivation of employees who are new to the organization. Some cultures may disapprove of questioning authority, so try to capture these suggestions in team retrospectives or as a part of your organization's normal continuous improvement efforts. Ensure that the remote staff know they are an integral part of improving how the work is performed.

Finally, make sure that you have good non-disclosure agreements in place, especially if there is a risk that your staff may move on to competitors. If you have some core algorithms or functionality that is vital to the success of your business, it is best to grant access to these on a strict need-to-know basis. Be aware of this when opening up your source code and documentation to many sites.

In summary, here are my ten best tips for effective knowledge transfer and IP protection:

1. Plan the transfer carefully

Identify knowledge holders, plan knowledge transfer activities. Oftentimes, it is the undocumented knowledge which is the important knowledge to transfer. Perform knowledge audits and draw knowledge maps where necessary. One way to perform knowledge audits is through structured interviews. An example of a knowledge map is a competency map, which is a matrix that lists people against their level of knowledge in products, parts of products, or technologies.

2. Monitor and measure the transfer periodically

Ensure that progress according to plan is monitored often, and take corrective actions where needed. At times, it can be difficult to know how well the transfer is progressing, but the ultimate test of a successful knowledge transfer is the quality and timeliness of work performed by the receiving center. Make sure the sending team peer reviews early work to ensure that the transfer has been successful. Frequent communication and feedback from both sides of the transfer are essential.

3. Give early language and cultural training where needed

Where cultures are different, it is important to prepare individuals regarding these cultural differences; this will pay back huge dividends later. To ensure that staff can understand each other, common language skills must be at a level where people can clearly express themselves and be understood. Test the language skills of new staff.

4. Meet early and often

Schedule face-to-face meetings, and try to build up personal rapport and trust. Visits to both sending and receiving sites are needed, and often for long durations, to achieve an effective and well-planned knowledge transfer.

5. Work together at the same site to increase effectiveness

Actively consider the possibility of an extended visit to the onsite staff. This enables individuals to spend time working in tandem, increase effectiveness, and decrease time spent in knowledge transfer.

6. Take advantage of new perspectives

Try to capture new ideas and new ways of working that will invariably come up during this intensive period with fresh insight from the new staff.

7. Identify knowledge that is sensitive

Keep this knowledge separated with good processes around access on a need-to-know basis.

8. Ensure a good set of non-disclosure agreements are in place

Where there is a risk that employees may end up with competing companies, ensure that a good (but fair) set of NDAs have been signed.

9. Test and improve knowledge transfer by giving development tasks early on

Allocating simple and small development tasks to remote team members gives them a chance to check whether they have all the basic knowledge and tools required to perform the task. It also gives you the confidence that nothing was missed in the process.

10. Ensure good IT processes, infrastructure, and tools

Ensure that access rights are well managed, and that remote staff have good access to the systems they need. Reliable and fast internet connections are essential for enabling high quality video conferencing possibilities.

About Darel Cullen

Darel has a long and broad experience of the software industry in many different industry sectors including nuclear, space, air traffic control, and industrial. He currently serves as Director of R&D at VolvohandelnsUtvecklings in Gothenburg, Sweden, a company that develops software for the automotive industry. For many years, Darel has been responsible for global software development. He enjoys bridging cultures and creating effective software teams.

Chapter 3 - Guidelines for Setting up a Distributed Agile Team

By: Abhilash Chandran

Introduction

Agile is the latest mantra in the software community. While it introduces new principles in software development, agile also brings along its own set of problems. Each company has its unique culture; so the best practices successfully followed by one company may not be suitable for another. Whatever the agile format is, the basic principles and practices remain the same. Agile works on the principle of inspection and adaptation which result in continuous innovation.

The following article provides a general guideline for setting up a distributed agile team. I wrote this keeping in mind the challenges I faced while setting up agile teams in India. However, almost all of the points discussed below are valid for setting up agile teams in any part of the planet.

Infrastructure

Lack of proper planning in setting up the right infrastructure results in failure of many projects. Agile thrives in an environment which promotes collaboration and trust. The right infrastructure is the key enabler for this.

Trial Run

You have created a state-of-the-art facility for your team and you are raring to go, but there is always a chance that something may be amiss. It is a good idea to do a dress rehearsal of the facilities before the team starts using them. When our new facility was created, we asked few of our people from different teams to work from the new office for couple of days. We found a few issues such as:

- A few network ports were not open
- A couple of VOIP phones in scrum rooms were faulty
- The network speed was not sufficient to handle peak loads

We were able to iron out all these issues before the entire team moved to the new facility.

Network

Here are few tips to consider while setting up an efficient network:

- Setup a fast, reliable, and dedicated internet
- If possible, setup VPN to the onsite network. This is critical if there are other onsite development teams which have to interact with offshore teams
- Use deployable virtual machines (VMware or Microsoft VM). This will help the teams a lot when they have to quickly deploy or test applications in a client’s environment.
- Set up a good code repository like TFS or GIT

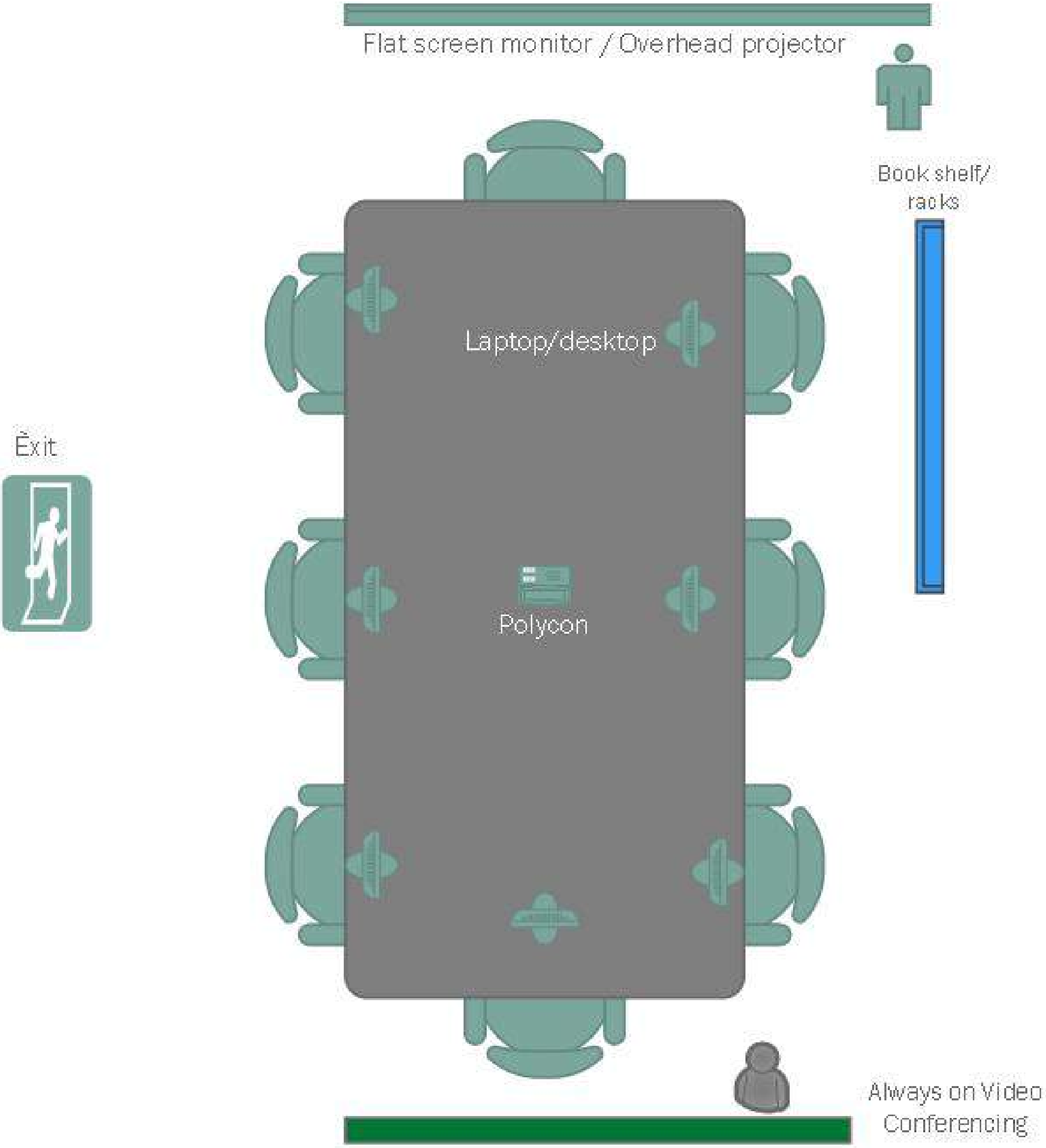
Agile/Scrum Rooms

How many times have we wondered why people send e-mails to the person sitting next to them when they could inform the same thing verbally? One of the key agile principle is “Individuals and interactions over processes and tools”. To facilitate more close interactions many agile teams are moving from cubicles to Scrum Rooms. The open room removes barriers between individual team members. As a consequence, this facilitates more face-to-face communication and better collaboration. This model can be followed by both onsite/local and offshore teams or any team which needs close interaction between its team members.

Following is my favourite scrum room model (Fig.1). An entire room is given to a team and team members are seated face to face.

Initially, when this layout was proposed, many in my team were not happy. They feared that they will lose their privacy and their “contact” with other teams. But after some time I got a lot of positive feedback from all my teams. Following are few of the feedbacks:

- Reduces outside interference to a minimum
- Improves face-to-face communication and collaboration
- Impromptu discussions – Since team is seated together they will engage in many short discussions/meetings. The entire team can participate without worrying about booking any external meeting room or disturbing others in nearby cubicles.
- Improves team bonding and trust.
- Productivity will increase because of the close interaction


Flat screen monitor / Overhead projector
Book shelf/ racks
Laptop/desktop
Èxit
Polycon
Always on Video Conferencing

Scrum Room Layout (Fig. 1)

Communication

The success of agile teams depends on the quality of communication. Misunderstanding among team members can be costly. Face-to-face communication is the best mode of communication. Although this is difficult to achieve with a distributed team, there are a few tools which can lessen this pain. Use communication devices/tools that are reliable and tested by such teams. The audio/video devices from Avaya and Polycon have given me favourable results in many locations. Irrespective of the location of team, a good communication strategy and high quality devices is critical for the project success.

- Always-on window

Set up a high quality, always-on video conference between the distributed teams. Use a large monitor with a high quality webcam (or video conference). The webcam should show all the team members. The monitor will act as a window between the distributed teams locally and in offshore locations. Through this 'window', people can talk to each other across geographic boundaries without setting up any special meeting.

As with normal communication, face-to-face communication works best when there is no barrier between people. The trick is to promote communication without any additional effort. For better results, place the device in a corner or near the wall from where people can view the screen from their seats. This is the best option if the bandwidth cost is not an issue.

- High quality video conference

If you don’t have an always-on window, install a high-quality video conference facility. This helps individuals put a face to the voice and helps them observe the non-verbal communication too. Personally, I insist on this for key meetings, such as planning, demo, or retrospectives.

- High quality phone

Initially, we had a lot of issues with the onsite product owners because of the low-quality phones. This even affected the quality of deliverables from the team because they could not understand all the directions from the onsite team. This situation has improved drastically when we installed high-quality phones for all the teams. The customer satisfaction improved along with quality of deliverables.

- Interactive digital whiteboard

Of course, not all the questions can be answered over phone or email. Sometimes, team will have to draw diagrams and connections to explain various relations. This is easy for collocated team. They can use normal paper or whiteboard. However, this is impossible when the team is distributed. To bridge this gap an interactive digital whiteboard can help.. The whiteboard contents can be controlled by computer, and any changes made to the whiteboard will also be reflected on the computer screen. This digital whiteboard is not very popular, but will be a good addition for these companies which need a lot of close interaction and complex discussion between distributed team members. There are some cheaper online alternatives too. Installing a big touch screen and using it along with an online collaboration tool will be cheaper.

- Webcams for individuals

In one of my scrum teams, a new team member has joined our distributed team in the USA. He has found it very difficult to remember the Indian names. Initial discussions over phone didn’t help either. To solve this problem, the team started talking with their American counterparts using webcams. This improved the quality of communication on both sides. Soon this became a standard operating procedure for distributed teams. This strategy helped people to remember faces and communicate with ease without the worry of offending others by wrongly pronouncing their names.

- Instant messengers

This is a must have tool for all software development teams whether they follow Agile or not. Use Microsoft Office Communicator, Skype or similar tools that enable this mode of communication. Such tools should be deployed for both, local and offshore teams. IMs provide an informal mode of communication. It is even better if the tool allows chat sessions to be saved; these saved chats text can be used as reference points for future discussions. (Microsoft Communicator and Skype have this option turned on by default)

- Wiki/SharePoint

Use SharePoint/Wiki to share common knowledge, best practices, etc. This will act as a repository for all the combined knowledge possessed by the company.

- E-mails

Provide web access and the option to synchronise e-mails to smartphones.

- Screen sharing

Use WebEx, LiveMeeting, or such other tools for screen sharing. Every team should have at least one account. Use a tool that has the facility to record both the screen and the audio during conference calls. Many knowledge transfer sessions will be done through this. Hence, such recorded sessions will soon become a video library for the team to refer to in the future. Camstudio.org has a pretty good free screen recording software.

You can store these videos in SharePoint or even in normal folders for future reference.

Build Trust

Agile is not very different from any other software development methodology. One thing which differentiates agile from others is that agile principles unveils all the hidden risks. Mutual trust is one of the key enablers for this. If the product owner and the team trust each other, then they will discuss openly without any fear.

Normally people do not trust strangers. To break this ice it is very important for the distributed team and the Product Owner to meet at the beginning of the project and spend time together at a common location. This will help to form an informal bond and get everyone accustomed to each other’s working style.

To continue and build this relationship of trust, it is a good practice to bring people together frequently. At every release cycle or during important phases, representatives from distributed teams must visit each other – travel to the distributed offices. Therefore, mutual trust will be enhanced and it will also help everyone to understand the different perspectives and other cultural barriers which are not visible when people communicate electronically.

Culture

Culture is a major stumbling block that often prevents successful adoption of agile principles. One should be aware of the fact that there is no silver bullet to solve the problems caused by cultural differences. This subject is so huge that I could devote an entire chapter just discussing this. Since I must also cover other topics, I will only give a coarse overview of some of the challenges that you will face while setting up a cross-cultural distributed agile team.

Change the approach

Don’t consider people as resources; consider them as normal people with emotions and ideas. This mindset has to start from the top management and percolate to all levels. Unless people are considered as human beings, management won’t understand much of the cultural problems brought out by the distributed teams.

Awareness

We had hired external coaches to embed this culture awareness in our European and North American teams. Many had a very different view about India and other offshore locations. A similar training has been given to all the people in offshore locations. Cross-culture trainings like these should be done periodically.

Trust and Bonding

Along with the cultural awareness trainings, management should provide opportunities to create personal bonding between team members.

- Plan team building activities outside office.
- Get volunteers from teams to accompany visiting team members from a different country. They can act as local guides for shopping and sightseeing. Personally, I have experienced a strong bonding after such outings. After such visits, the barriers of formal interaction are replaced with a touch of informality.
- Periodic communication through e-mails or other channels, about national events or celebrations in other locations, will help the team to understand their counterparts in a much better way. This, in turn, adds emotions and a feeling of a big family to the voice they hear every day.

Shared Project Vision

In many of the distributed teams in the US, UK and India which I have helped to set up or have worked with, the major area of concern of the top management was that people do an excellent job in executing what they have been told to do, however, they seldom come up with creative solutions to improve the project. Almost all of these teams did not know about the project vision or long-term objectives of the project. In most of the cases, they only work on a small part of the application and they would do a pretty good work in that. But they will miss the key principle of continuous improvement because they do not have any clear idea about how they fit into the larger scheme of things. This lack of knowledge will become an impediment that influences their practices and efforts towards the long-term objective. You will be surprised by the turn around when everyone is made aware of this vision.

- Setup periodic communication about the project vision, overall project milestones, and key developments; use video conferencing or webcasts for this.
- Publish regular e-mail alerts about clients' feedback.
- Prepare newsletters about the opportunities, changes, and challenges in the application industry or domain.

Overlapping Work Hours

Ensure that at least 1-2 hours of overall overlapping time is being used between distributed teams. An overlap of 4 hours would be the best case. Ideally, this pain should be shared between both onsite and offshore teams. But most of the time, we observe that only offshore locations change their working time to accommodate the teams in Western Europe or North America. Overlapping working hours provide the opportunity for teams to talk to each other and clarify their queries. If there is no overlap, teams will have to depend heavily on e-mail and other written modes of communication. Due to time zone differences, there will be delays in answering e-mails. With written mode of communication, people will have to spend more time drafting e-mails and clarifying all the misunderstandings. This delay in decisions and clarification will reduce the productivity of distributed teams because a major part of their time will be blocked from doing any productive work. I have seen teams with no overlapping hours calling themselves agile. Such teams cannot become truly agile because they will be slow to respond to changes and they will have to depend heavily on processes and tools over people interaction. These are against the key principles of the Agile Manifesto (http://www.agilemanifesto.org).

Technical Practices

Use technical practices that reduce the feedback cycle. Smaller cycles unveil issues at the earliest and help teams to solve them immediately. This is important to prevent the disruption of normal working because of any bad code or configuration. In one of my first agile teams, I remember waiting for 5 hours for our team from UK to come online and rectify the build failure caused by their code check-ins. If they had a proper gated check-in policy, where all the code was compiled and verified for any build issues, then we would not have lost several hours. As the distance between teams grows, companies will have to invest in XP practices. I have listed few of the technical practices followed by many successful agile teams. I have also provided links for further reading:

Continuous Integration -http://martinfowler.com/articles/continuousIntegration.html
Gated check-ins - http://msdn.microsoft.com/en-us/library/dd787631.aspx
TDD – Test Driven Development - http://en.wikipedia.org/wiki/Test-driven_development

BDD – Behaviour Driven Development - http://en.wikipedia.org/wiki/Behavior-driven_development
Functional UI automation (Selenium, Microsoft Test, QTP, etc.) -
http://www.infoq.com/articles/gui-automation-patterns
ATDD – Acceptance Test Driven Development -
http://testobsessed.com/wp-content/uploads/2011/04/atddexample.pdf

There are many more XP practices which can be followed. You should not be in a hurry to implement all practices simultaneously; rather take only one practice at a time, educate the team about it, give proper trainings, and then implement it in a phased manner. This will help teams to adopt the new practices without any major disruption or resistance. If you are creating a new team from scratch, then you could insist on having these practices from day one.

Sprint Duration

Try to keep sprint durations from 1 to 2 weeks. A lesser duration is advisable because it creates shorter feedback cycles in terms of planning and demo. This also helps teams to identify and manage risks better.

Ken Schwaber and Jeff Sutherland have introduced Scrum sprints of 30 days duration. Nowadays, the majority of the companies prefer2 week sprints. If you don’t have any idea; start with 2 weeks. After a couple of 2 weeks iterations, get together with the team, collect their feedback and decide whether you want to change the sprint duration or not. Personally, I have seen teams with sprint duration of 20 days, 15 days, 2 weeks, and even 1 week.

Checklist for sprint duration change

Reduce the sprint duration from 2 week to 1 week

1. Poor estimation (if the team complains that they work for 8 hours on the task estimated for 4 hours)
2. Poor quality (lot of bugs and performance issues)
3. Unable to meet the sprint objectives
4. Lot of scope creep – new features are added by the product owner or other people responsible for product management during the sprint execution.
5. Nature of deliverables (if the team is responsible for the delivery of small reports or documents which won’t take more than couple of days)

When the sprint duration is reduced, the team will meet more often for planning and review. This will contribute to better planning, estimation, and understanding of sprint objectives. PO will also get more time managing all the changes in the backlog due to frequent changes in customer priority.

Increase sprint duration from 1 week to 2 week

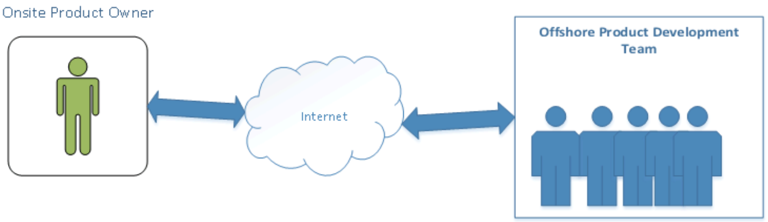
1. Product backlog is almost stable
2. The inability of the customers/PO/stakeholders to meet periodically to review the delivery
3. Nature of deliverables (some deliverables need long time to develop, integrate and test)

Knowledge Sharing

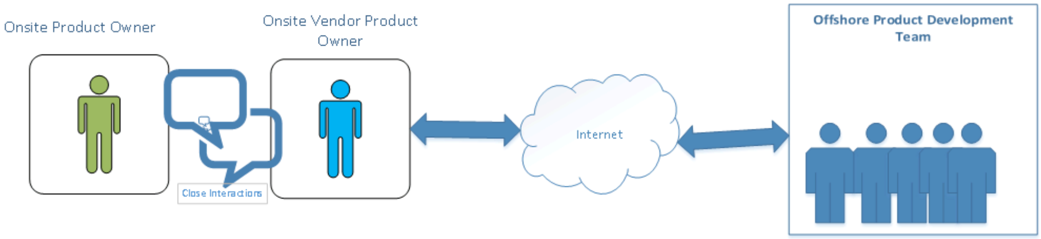
Apart from normal team meetings, plan for regular backlog grooming meetings.Setup separate meetings to discuss about the upcoming features. Such design sessions will help the team to provide early feedback to the product management team. Along with regular feature design meetings, insist on having regular technical design meetings between local and distributed team members. Technical discussions can focus on agile ways of implementing upcoming features or continuous improvement of existing architecture or framework. Similarly, best practices stored in a Wiki or SharePoint become useless until people discuss them in the meetings. Sessions like these also give an opportunity for developers from different countries to discuss and share their experience and knowledge.
Use video conferencing or webcams for such discussions.

Team Organization

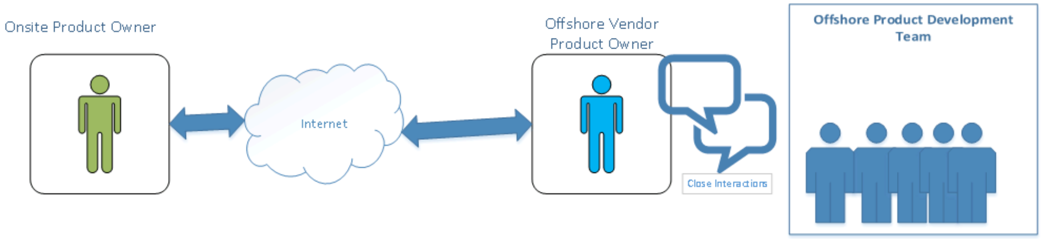
There are different ways of setting up teams for distributed agile development. Here, I will mention some of the most popular formats.



Type A - **Onsite Product Owner and offshore development teams**


Onsite Product Owner
Onsite Vendor Product Owner
Close Interactions
Internet
Offshore Product Development Team

Type B - **Onsite client Product Owner – Onsite Vendor/Partner Product Owner**


Onsite Product Owner
Internet
Offshore Vendor Product Owner
Close Interactions
Offshore Product Development Team

Type C - **Onsite client Product Owner – Offshore Vendor/Partner Product Owner**

Product Owners

A. Onsite Product Owners and Offshore Development Teams

This is the most popular format of agile teams.

B. Onsite Client Product Owners – Onsite Vendor/Partner Product Owners

The partner/vendor company assigns a permanent product owner at the onsite location to interact with the client team or the client product owner. They will be the real product owners of the offshore teams. The vendor Product Owner interacts with the client product owner or product management team to understand and get requirements. The vendor PO will work with the distributed agile team (or teams) to convert the requirements into actual software. They preside over the teams for planning, feature demo, etc. The vendor PO will, in turn, demo the working software to the client PO. This format is used when the client product or management team do not have any overlap of working hours with the feature teams or because they cannot spend time with the development teams.

C. Onsite Product Owners – offshore Product Owners

This model can be followed by offshore teams that have a good understanding of the products or by companies which have both onsite and offshore development teams. Like the previous model (Type B), the offshore PO works with the onsite PO to get the requirements and works with the offshore team to convert those requirements into working software.

Development Team

In most cases, the feature development teams are co-located at a single location. There are very extreme cases when the feature teams are distributed over different geographical locations. Even though we say that these distributed people are part of the team, they may not exhibit real characteristics of a good team. People at these distributed locations will form their own norms and informal working agreement which will be entirely different from their team mates in other locations. If such distribution is large (more than 3 people in different locations), then try setting up a separate scrum team at that location. Otherwise, they will behave like two separate teams working under one chain of command. In the long run, this will be more productive because of the lack of complex communication and other intra team dependencies.

Scrum Masters

Scrum masters play a key role in making the agile way of software development possible. Many companies make mistakes while selecting the person for this job. To avoid confusion, I have included links to a few articles which explain the role of the scrum master:

- Scrum Master’s Checklist - Roles and Responsibilities
  http://www.theagileschool.com/2012/03/scrummasters-checklist-roles.html
- What powers does a scrum master need to have?
  http://www.theagileschool.com/search/label/scrum%20master
- Can a scrum master be a developer also in the team?

About Abhilash Chandran

Abhilash Chandran is an Agile Software Development Manager, coach, and practitioner employed with Xerox Corporation. He is also an active software engineer and actively leads teams through software development projects. He specializes in coaching teams using eXtreme Programming (XP), SCRUM, and Lean software development practices and techniques.

His passion is to help people and teams discover the magic of Agile - continuous improvement and waste reduction.

He is an expert in Agile Project Management, Implementing Lean Software development practices like TDD, BDD,ATDD, Pair Programming, Clean Code, Agile Product Management, and Agile coaching (coaching teams and groups on Agile Methodologies - Scrum, Kanban, XP, Scaled Agile, etc.).

Chapter 4 - Three Steps to Success in Planning Your Offshore Project

By: Andy Jordan

When you think about outsourcing one or more project elements, what are you most concerned about? Missed deadlines? Low quality delivery?Inaccurate or incomplete scope? Increased risk? All of the above?

Chances are that you said ‘all of the above’, but guess what – the remote team is concerned about exactly the same things. When you are dealing with a team of people who are separated from you, everyone worries that the physical separation is going to lead to problems. While there are no guaranteed ways to succeed, by working together during project planning, and recognizing that you both share the same concerns, the chances of success can be greatly improved. That’s what I want to address in this chapter.

The first thing that both the vendor and the project manager need to recognize is that you are both on the same side. Too many remote relationships are built on a foundation of mistrust – especially if we are dealing with a buyer / vendor relationship rather than different departments within the same organization. At the most fundamental level, everyone needs the project to be successful and a collaborative approach to the project will make it much easier to work together and deliver that success.

Collaboration needs to begin with the planning of the project. At the most basic level, a project plan is fairly straightforward – the work is broken down into individual tasks that are sequenced with relationships between them identified, are measured, and assigned to resources. However, when work is outsourced,there are a number of considerations that can add considerable complexity to the plan:

- Is the work that is being outsourced fully understood? If work is being given to an outside group (whether internal or external to the organization) then there will inevitably be a learning curve where the remote team needs to understand what the project is about, what is expected of them, etc. And the project manager and his or her core team need to understand exactly where the responsibility of the remote team starts and stops. This may be even more complex if the work is being outsourced because of a lack of expertise within the core team – there may not be a good understanding of what is and isn’t possible.
- How easy is it to track progress? If the project is being executed using Agile methods, or if the work naturally lends itself to interim deliverables, then there will be less risk than if the work consists of just a single deliverable at the end of the outsourced tasks. If there are no obvious interim milestones then it will be harder for everyone involved to confirm that the work is on track.
- How easy is it to integrate the outsourced work with the rest of the project deliverables? If the outsourced element is a stand alone component, then there is less concern.But if it has to be integrated with other project elements, then not only must time be included in the plan for that work, but time must allow for learning curves, adjustments, etc. as two different elements are brought together.

Let’s look at each of these items in more detail and consider how they may impact project planning, and what you can do to ensure that these elements are appropriately managed.

Understanding the Work

This is by far the biggest challenge facing projects that have elements of remote or outsourced work. Often, it is difficult to even recognize that there is a lack of understanding or misunderstanding in the first place. I have seen many situations where the vendor and the buyer both felt that there was a common understanding, only to find that what was delivered was very different from what was expected. There are many reasons why this might occur.But in this chapter, we are focusing on how we need to plan for these situations, so let’s look at that.

Step 1 - Plan time for mutual understanding of the work

The plan to have a mutual understanding of the work has to begin right up front, even before a contract is signed or a team is assigned. The mistake that many organizations make is in defining in their requirements what they think they need in terms of specific deliverables – features, colors, design elements, etc. Instead you need to focus on defining the business need in two distinct ways:

- The overall purpose of the solution to be built - what it is for, how it is used, how it drives benefits, etc.
- How the solution fits into the larger corporate purpose of your organization - your business, revenue model, etc.

That is the area where your project and procuring teams should have expertise, and that will provide the most meaningful information to the vendor. While this information may never make it into the plan as a specific line item, it is critical to building that successful plan.

You should then expand these discussions to cover the model that will be used for outsourcing, the work to be performed by each team, etc. and the relationship starts with a strong understanding of who will do what (at a high level) and how it will be managed. In many cases, your organization will have its own project manager, and will almost certainly own the overall product relationship, even if the vendor is doing most of the delivery work. Of course all of this planning takes time, and that time needs to be reflected in the plan.

With a clearly documented and internally agreed upon business need, the following becomes much easier:

- Work breakdown, vendor vs. customer ownership, estimation, and resource allocation. There is a much clearer demarcation between roles on the project – much of the uncertainty around who owns what is avoided by allowing each of the work teams to develop their own plans to meet the business needs. The project manager retains ultimate accountability for ensuring that the plan reflects all of the required tasks, but the details are provided by the relevant experts. This approach also helps the various team members to get to know one another as they work together on project planning.
- Validation of the work – while the ‘how’ to test and validate the work may still be difficult to determine depending on the unique scenario, the ‘what’ constitutes success is much easier to

> understand because it is defined up front in terms of business needs. This helps to avoid disagreements about whether a solution is flawed or not – the all too common 'you don't get what you need, you get what you ask for' situation. I have seen many a potential problem avoided by agreement at the planning stage about the performance criteria of the completed solution.

A common outcome of this approach is that the project manager is faced with a delivery schedule which is at odds with the schedule that has been set for the project. Frequently, this results in the vendor being pressured to deliver in less time than is required to complete the work. While project schedules and deadlines are obviously important, the idea that a party who may lack the understanding of exactly what is required overriding the duration estimates provided by experts in the field, is flawed at best, and disastrous at worst!

Tracking Progress

One of the concerns that organizations often have with outsourced work is the lack of visibility into the work that is being done. This is one of the reasons why customers are often enthusiastic about Agile-based approaches as they force vendors to provide interim deliverables on a regular basis. However, this isn't always possible, and there will be situations where there are no deliverables until the work is complete, especially beyond technology. I worked with a sales team that outsourced development of a new sales compensation model and they saw exactly this - because of the interdependencies between different elements, the model was developed before they saw what was being developed. This doesn't mean that the work can't be tracked.However, the old Russian adage applies here – 'doveryai, no proveryai' which roughly translates into English as 'trust, but verify'.

Step 2 - Ensure that progress tracking is appropriate and objective

The group that is doing the work, whether it is a remotely located internal department or an external vendor should be trusted. If the relationship between the two parties does not have a foundation of trust then there will be too much focus on the people and the relationship, and not enough focus on the work. However, that trust shouldn't be blind faith, there should be attempts made to verify that things are going well, and that is what needs to be built into the project plan.

The plan should include regular checkpoints with clear agreement of what is expected - achievement of those checkpoint expectations demonstrates progress on the project. These may not be tangible deliverables, but they can still be 'real'. In the sales example, we had the vendor demonstrate the different elements that they planned to incorporate in the compensation models, the different roles and levels that were being developed, the variables that affected commissions, etc. In isolation, these are meaningless but they demonstrate continued progress towards the goal, and importantly they are objective measures rather than judgmental ones.

There is a tendency to assume that verification is easier when the work is being carried out by a remote team within the same organization than when an external vendor is involved, but this isn't necessarily the case. In the internal department scenario, the people involved may be more aware of how to 'get away with things' if they were so inclined; so I always treat verification in the same way regardless of whether the remote team is made up of internal or external resources.

In addition to providing regular status updates at weekly meetings, shared through collaboration tools and 'as needed' conversations between the project leads, I always ensure that the project manager schedules more formal gateway reviews on work items that he or she has less visibility into. This will be in the form of a structured discussion at key points in the project where all stakeholders review the 'checklist' of items required to pass through the gate and achieve the interim milestone. That may include the handover of interim deliverables, but it may simply be showing completed data models, confirming designs, etc. These checkpoints should be viewed as confirmation that the vendor's work is on schedule, and where possible should be tied to payment schedules. They also provide an opportunity for corrective actions to be identified and approved, if necessary. This is where the trust part of 'trust, but verify' comes in – if there is no trust between the parties then there will be a tendency to hide problems (or to assume that problems are being hidden) and there will be no ability to agree on a realistic corrective plan that will help to ensure that the project is successful.

Even when there is trust between the parties, there may still be a tendency to view the situation differently – avendormay view a two-day delay as minor, whereas a customer may view it as the start of a major problem. For this reason I always make sure that the criteria for success in these planned reviews is documented and agreed to up front. Examples may include anything from sign-off on a document to completion of certain modules of work. I have even seen measures around hiring people to perform the work and delivery of documentation. The key is simply to agree at the start of the project what success looks like at these milestones so that there is no subjective negotiation during the reviews. At that point, both parties should also agree on the size of delay, cost overrun, etc. that will trigger corrective actions for each of the milestone reviews.Thereafter, the review itself simply becomes a case of comparing actual values with the plan and acting accordingly.

When scheduling these reviews the project managerneeds to ensure that they occur frequently enough to prevent problems from escalating out of control, but also spaced out enough to allow for real progress to be made. Moreover, they need to begin late enough after the work starts to have something solid to review, and early enough before the end to provide time for recovery and correction.

Integrating the Work

In many projects, integration is something that is under-planned. The need to bring multiple project elements together into a cohesive solution necessarily occurs late in the project when time schedules are often underpressure, and project managers frequently seek to 'save' time by compressing the amount of time that is allocated for integration. This rarely works well, and when outsourcing is brought into the mix, it is a recipe for disaster. It should be noted that this is not necessarily a technical integration; it may simply be the integration of technology into business operations - the complexity can be just as significant.

Step 3 - Plan for a successful integration

You need to plan for integration right from the outset of the project, with an expectation that it will be difficult and will require adjustments to be made as different components from different sources are brought together. The need for well-executed integration is even more important if there haven't been interim deliverables, and any problems that are missed during this phase of the project will make it into the final release where end users will find the problems! The sales compensation plan project that I described earlier involved executives and the vendor travelling the country explaining the model to staff and then revisiting them after 3 months to address questions and concerns - an expensive integration, but one that helped ensure that things went smoothly. That may be overkill for your project, but you need to consider all of the implications when the different project elements come together as a complete solution – it is your organization that will be at the frontline of support and issues.

From a planning standpoint, if some of the work is outsourced, I always begin integration activities while the build work is still underway. At a minimum, the teams can build a better understanding of how they will manage the integration and in turn this helps them to identify where potential problems may occur. This approach also moves the learning curves into earlier phases of the project and avoids the need for rapid learning at the start of integration – further reducing the likelihood of mistakes being made.

To further ensure that integration is a natural extension of the build work that is occurring on projects, I also make sure that integration planning is a natural extension of earlier work. The integration plan shouldn't be built in isolation from all of the other work that is undertaken; rather the documents and artifacts that are developed for requirements, design, etc. should naturally feed the integration planning activities.

Conclusions

So what does all this amount to when we bring it together? In today's economy where organizations are looking to focus more and more on their core areas of expertise, and where the rate of change in technology continues to accelerate, the need for specialist functions to deliver projects is only going to grow. This means that the number of projects that use remote teams, whether internal or external, will continue to increase as organizations seek out those specialist functions, and if organizations don't do a better job of recognizing that those projects need to be managed differently,then they will continue to fail.

Organizations need to recognize that using outsourced workers can dramatically increase their chances of success, but only if the project is managed appropriately, recognizing that these projects are different from initiatives where all resources are co-located. There is no magic bullet solution, but taking a little bit of time and effort to structure this simple three-step approach can go a long way in helping you avoid the pitfalls.

About Andy Jordan

Andy Jordan is President of Roffensian Consulting Inc., an Ontario, Canada based management consulting firm with a strong emphasis on organizational transformation, portfolio management, and PMOs. Andy has a track record of success in managing business critical projects, programs, and portfolios in Europe and North America in industries as diverse as investment banking, software development, call centers, telecommunications, and corporate education.

Andy is an in-demand speaker and moderator who delivers thought provoking content in an engaging and entertaining style, and is also an instructor in project management related disciplines. He always strives to provide thought provoking presentations that drive his audience to challenge accepted norms while providing actionable content that can be applied in the real world. Andy's first full length book 'Risk Management for Project Driven Organizations' is now available.

Chapter 5 - Basis of Success - Good Governance and Multi-Level KPIs

By: Erwin de Bont

Introduction

After an intense period of successful negotiation and a new contract, you will realize that this is just the end of the beginning of what you tried to achieve when you made the decision to outsource. Soon, you will discover that the contract is not flexible enough to meet your ever- changing requirements. You may realize that the supplier is optimizing his profit to deliver only the agreed services against your smart performance indicators. Your suppliers stick to the contract, the atmosphere between you and supplier is getting vitiated, and in the end, you are unhappy. This is not what you hoped for! How can you make sure that the changing business goals are met, that the supplier is doing more than agreed, and that your expectations are being fulfilled?

Tightly written contracts can work against success if they inhibit the ability of businesses to adapt to evolving and highly complex market conditions. When it takes months of renegotiation to modify contract language, the moment of opportunity could be long gone. Should your agreements be radically different? Shouldn'tthey be designed to accommodate continual change? When written correctly, there is effectively no 'FINAL' contract. Terms and conditions are designed to accommodate flexibility in order to adapt to unexpected situations. Governance systems are in place to address circumstances that require creative, out of the ordinary resolutions. Measurement reporting should allow for aberration and unique events.

The fundamentals to start at are governance and performance indicators. However, in my experience, the majority of conflicts between suppliers and customers are based on bad governance and performance indicators. Your contract must at least covera governance and reporting structure that accommodates flexibility from both sides. Yes -flexibility for your supplier because when there is a single way of enforcement, you will discover that your expectations will not be met.

Where Do You Start?

Before you decide to outsource, you have to think about what you want to achieve by outsourcing. What is your company's objective? Is it convenience, cost cutting, improvement in quality, joint go-t--market, innovation, etc. If you outsource, it is useful that your potential supplier understands your objectives and ismotivated to contribute to your business objectives. The supplier's proposed contribution should be a part of the contract.After signing the contract, you should be able to measure the contribution of your supplier against your objectives. Moreover, you should be able to change your objectives because your market conditions will keep changing.

Governance & Performance Indicators

Your contract should enable flexibility. So, whenever you start thinking about outsourcing, start thinking about your objectives, create flexibility for future changes, and ensure that they are a part of the RFP and contract.

Of course, you have to make sure that all the objectives are measurable (performance indicator) and that the governance offers you the possibility to change the objectives and KPIs. But realize that in the end, it is not about how many widgets are promised or the speed with which calls are answered. Rather, ensure that the commitments follow jointly agreed behaviors that promote trust, transparency, and compatibility,which are hallmarks of such a relationship. The relationship is most important. Your approach to outsourcing must set the tone for the partnership and the relationship; partnership itself provides the impetus behind all progress. Goals can change. Reimbursement and incentives can change. As a result, processes may also change. A good governance and reporting structure should enable all this.

A governance annex in the contract describes the authorized responsibilities, the predefined consultation structures, and escalation mechanisms of both parties. Sure, there should be a clause which enables changesin the contract to meet security and compliance requirements.

In this chapter, I use the example of a large organization that wants to outsource many primary processes to help their customers. So, they outsource their core business. This company wants to achieve the classic objectives ofreducing cost and improving quality. If you are a smaller organization or have other objectives, you can tailorit according to your situation.

You have to consider that in a large company many departments reside with different operational dynamics. In a contract between two companies where different dynamics occur, a master agreement should cover only generic themes like payment schedule and default, intellectual property, etc. Also, governance and main KPIs must be a part of the master agreement.To make sure that every department can be successful in their sourcing ambitions, they must have the freedom to draw their own contracts with a supplier, within the strict boundaries of the master agreement. Suppliers may want to work independently. However, to support the ambitions of achieving company objectives and allowing departments the freedom to outperform, it is advised to set up a multi-level governance and performance indicator framework.

To facilitate individual departments, it is advisable to have a separate team consisting of experts in outsourcing. I will call this team Global Sourcing team. If you are a small organization, make sure that you have at least one person who is dedicated to outsourcing and make it a part of the business case before you decide to outsource. The Global Sourcing team is responsible for securing all strategic objectives and general requirements like security, compliance, escalations, etc. This team should mainly focus on the contract, the relationship, and the structure of the collaboration. Meanwhile, the operations are managedby the business units.

Besides that, you will notice that suppliers, after contracting, will have one customer teamor single point of contact to supervise the cooperation and bridge the culture differences. If you do not have a centralized approach to managing the relationship, you will end up with many different outsource situations that probably will not meet the organization’s objectives.I have had many experiences where after 2 or 3 years of the relationship, the outsource situation is fragmented, the business case is not met, and the internal resistance to outsourcing is increasing.Of course, the supplier will get all the blame for it. Besides that,the Global Sourcing Team can act as an overall intermediation between internal departments and suppliers. You have to protect yourself and the supplier so that one department does not spoil the mutual trust and atmosphere, and hinder the larger objectives.So, do yourself a favor and organize outsourcing centrally.

A good governance structure is nothing but a meeting schedule where all responsibilities and authorities between parties are clear for everyone working in the relationship. As guidance, indicators should be the input for discussions in the different fora. When you outsource activities, different performance indicators may apply. I see three performance indicators:

1. Strategic Performance Indicators (SPIs)?

    These performance indicators are intended to measure the quality of the objectives you have when you just start thinking about outsourcing.

2. Key Performance Indicators (KPIs)

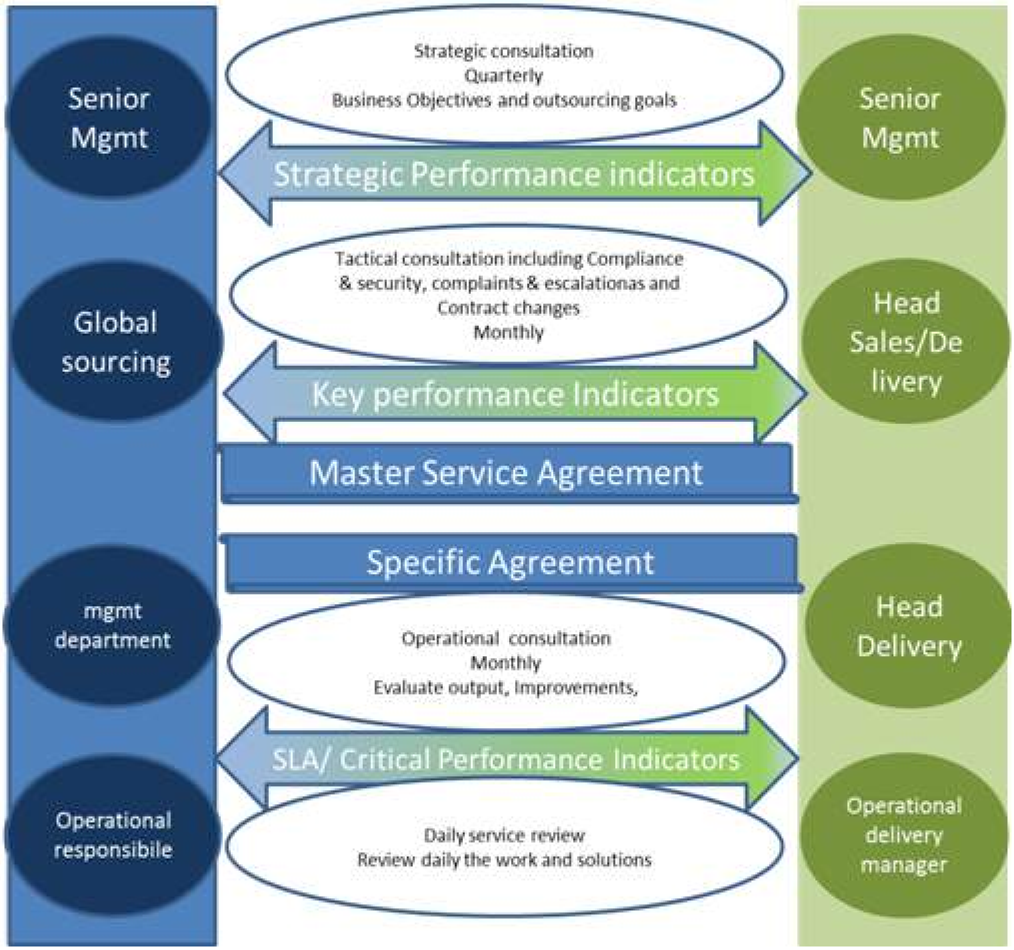
    These performance indicators measure the quality of the process, financials, and compliance of services according to the contract.

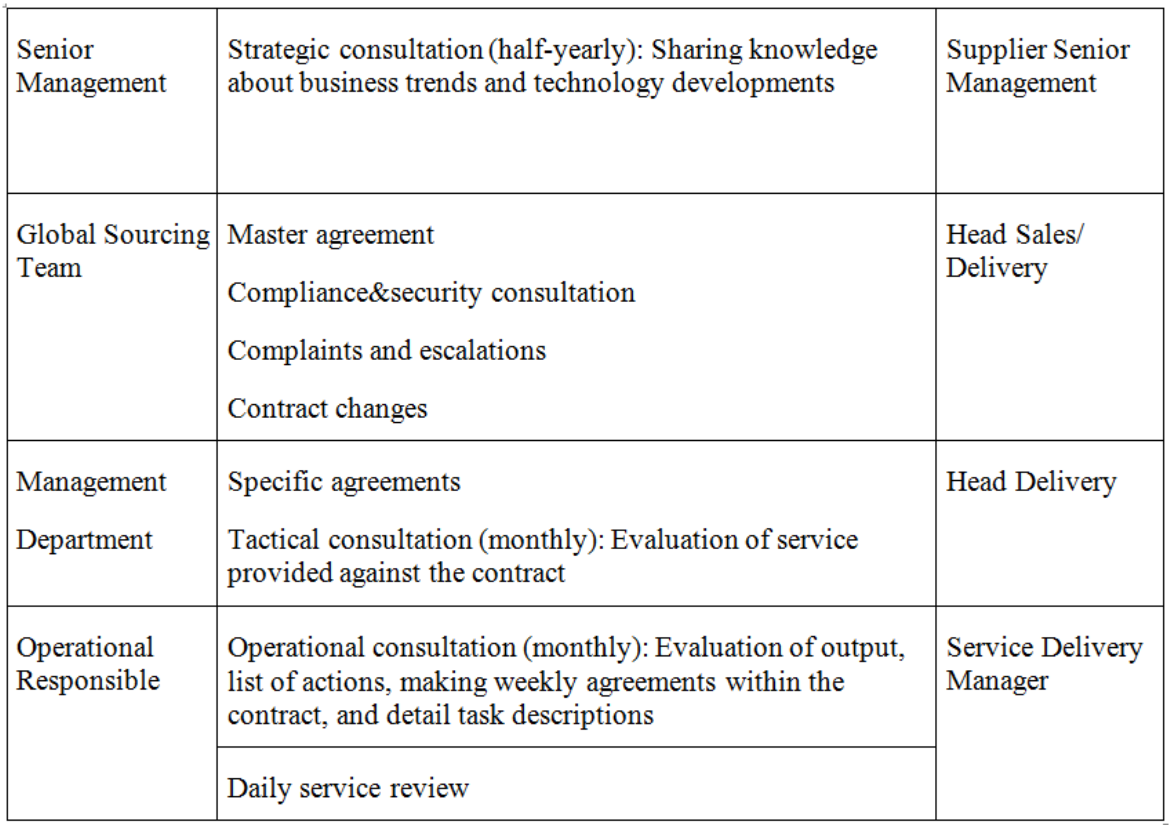
3. SLA: Critical Performance Indicators

    These performance indicators are required with regard to the quality of the outsourced services. In fact, these are indicators about the work you have outsourced.

Consultation Structure

The following types of consultation are part of the execution of the contract. The two parties do not charge any additional cost for this structure. These are formal types of consultation which are conducted on the basis of an agenda, with records in the form of minutes, with a list of actions and decisions. Of course, these meetings should accommodate building relationships. If it takes time to build relationships, these meetings can be used to agree upon matters. The picture below gives an example of a governance model.


Senior Mgmt
Strategic consultation
Quarterly
Business Objectives and outsourcing goals
Senior Mgmt
Strategic Performance indicators
Global sourcing
Tactical consultation including Compliance & security, complaints & escalationas and Contract changes
Monthly
Head Sales/De livery
Key performance Indicators
Master Service Agreement
Specific Agreement
mgmt department
Operational consultation
Monthly
Evaluate output, Improvements,
Head Delivery
SLA/ Critical Performance Indicators
Operational responsibile
Daily service review
Review daily the work and solutions
Operational delivery manager



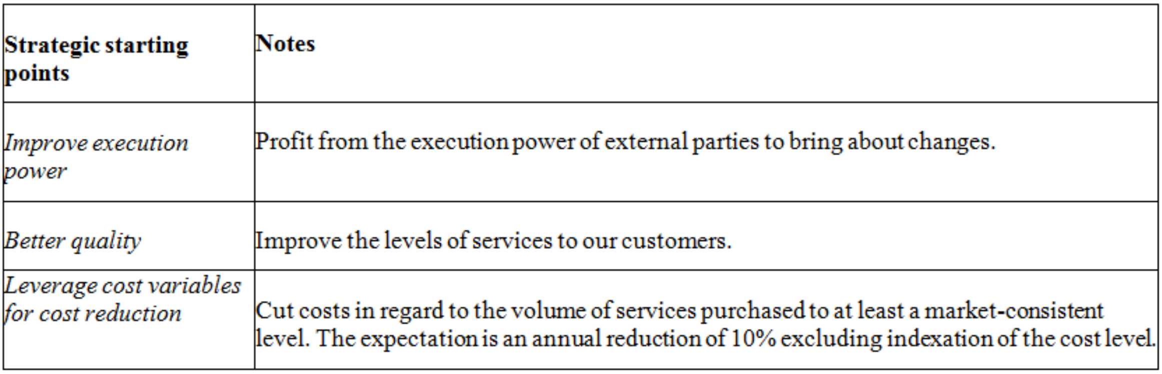
Senior Management	Strategic consultation (half-yearly): Sharing knowledge about business trends and technology developments	Supplier Senior Management
Global Sourcing Team	Master agreement Compliance&security consultation Complaints and escalations Contract changes	Head Sales/ Delivery
Management Department	Specific agreements Tactical consultation (monthly): Evaluation of service provided against the contract	Head Delivery
Operational Responsible	Operational consultation (monthly): Evaluation of output, list of actions, making weekly agreements within the contract, and detail task descriptions	Service Delivery Manager
	Daily service review	

Strategic Consultation

Strategic consultation is a platform for sharing knowledge, information, and ideas about available and suitable new technologies. Senior management of both companies must share company objectives and knowledge about the market, new business models, and technologies in order to stay successful as customer and supplier. Your supplier will be able to anticipate new developments that are relevant to you and you might profit from using a better technology.

During this consultation,an account is given in broad-brush terms of the delivered performance. As an example, here are the strategic performance indicators that reflect the large company.

The following table contains the strategic objectives of the example company. They lead the agreement.

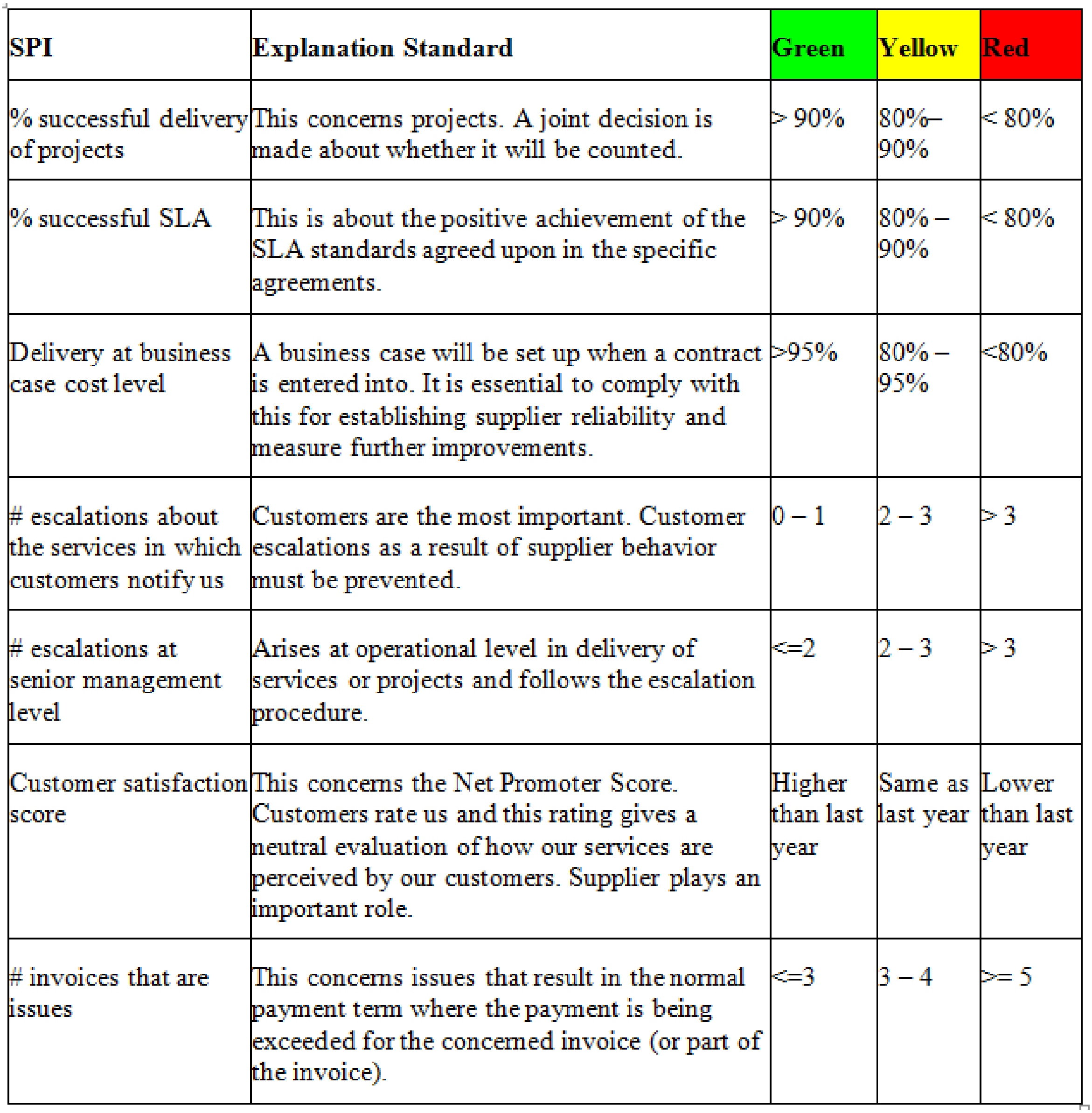


Strategic starting points	Notes
Improve execution power	Profit from the execution power of external parties to bring about changes.
Better quality	Improve the levels of services to our customers.
Leverage cost variables for cost reduction	Cut costs in regard to the volume of services purchased to at least a market-consistent level. The expectation is an annual reduction of 10% excluding indexation of the cost level.

Based on these objectives, you must develop indicators. Some indicators are given in the below table as examples.

The indicators given below are specified in the Master Service Agreement and they will be re-specified annually in the tactical consultation. At that time, the party proposing a departure from the standard agreement as initially specified will provide clear and reasonable reasons. If both parties cannot reach an agreement in the tactical consultation about the definition of a standard, the proposed new standard will be specified in the strategic consultation.

The performance actually achieved is prepared and determined jointly once a quarter in the tactical consultation meeting. If no agreement is reached in this consultation about this performance, decision making in the strategic consultation is requested. An example of setting strategic performance indicators is given in the below table:



SPI	Explanation Standard	Green	Yellow	Red
% successful delivery of projects	This concerns projects. A joint decision is made about whether it will be counted.	> 90%	80%–90%	< 80%
% successful SLA	This is about the positive achievement of the SLA standards agreed upon in the specific agreements.	> 90%	80% – 90%	< 80%
Delivery at business case cost level	A business case will be set up when a contract is entered into. It is essential to comply with this for establishing supplier reliability and measure further improvements.	>95%	80% – 95%	<80%
# escalations about the services in which customers notify us	Customers are the most important. Customer escalations as a result of supplier behavior must be prevented.	0 – 1	2 – 3	> 3
# escalations at senior management level	Arises at operational level in delivery of services or projects and follows the escalation procedure.	<=2	2 – 3	> 3
Customer satisfaction score	This concerns the Net Promoter Score. Customers rate us and this rating gives a neutral evaluation of how our services are perceived by our customers. Supplier plays an important role.	Higher than last year	Same as last year	Lower than last year
# invoices that are issues	This concerns issues that result in the normal payment term where the payment is being exceeded for the concerned invoice (or part of the invoice).	<=3	3 – 4	>= 5

The table contains a brief explanation of each SPI and its standards. These standards have the following significance:

- Green: Supplier correctly fulfils its role.
- Yellow: A score in the yellow column is an indicator (early warning) that the supplier is not fulfilling its role in accordance with expectations.
- Red: A score in the red column indicates that the supplier's performance is clearly below standard in fulfilling its role.

This simple cockpit gives senior management enough feet on the ground to discuss necessary improvements and business and/or market changes.

Tactical Consultation

In the tactical consultation forum,the agreement is continuously assessed for the extent to which the agreed services and service levels are being achieved as agreed: the assessment of performance against the agreement. The objective of this consultation is to discuss the performance of the supplier based on regular reports, audits, or benchmarks and deciding about desirable improvements in services, bonuses/penalties, and possible amendments to the contract (and Annexes).The Global Sourcing Team takes the lead for these parameters.

The supplier must collaborate in enabling the measurement of the KPIs and following up on the recommendations. The following indicators are the key performance indicators that apply:

- **Number of audits conducted on directed processes in accordance with the audit plan**

  An audit plan is prepared annually. The audit plan must cover all contract requirements including security and compliance. This plan specifies the types and number of such audits. The objective is to conduct all audits within the specified period and in accordance with the quality criteria (100%).

- **Percentage of audit recommendations that were followed up**

  The Global Sourcing Team agrees with internal and external auditors about which audit recommendations will be followed up by the supplier. The extent to which the agreed actions have been completed within the specified period and in accordance with the quality criteria is established annually. The objective is to implement 100% of the agreed recommendations.

- **Comply with the business case**

  A business case will be agreed upon for eachSpecific Agreement.

  Achieving the business case is a basic principle of the Specific Agreement. A price level is agreed for 5 years when entering into a Specific Agreement. The business case will have a three-year rolling forecast, i.e. in year 3 of the agreement the business case will be redefined for years 4, 5, and 6. The objective is to ensure that the Specific Agreement does not exceed the business case.

Operational Consultation

Operational consultation is concerned with coordination between both parties about routine operations. During this meeting (work-related), agreements in order to optimize the quality, effectiveness, and efficiency of the services are discussed. The objective of this consultation is the structured discussion of operational performance, issues, substantive agreements about actions, improvement measures, and work-related agreements.

The supplier will provide the services in accordance with the service levels. Supplier compliance with the service levels will be measured for each reporting period, as set forth in the Specific Agreements.

With respect to those components of the services for which a service level is not defined, the supplier should perform such services at least at the same level and with the same degree of accuracy, quality, completeness, and responsiveness as was during the 12 months immediately prior to the effective date.

Most operations are complex and contain lots of operational performance indicators. However, all indicators are necessary but not every indicator is equally important. In order to keep the relationship intact and not spoil it by discussing unimportant operational indicators, it is important to keep the main focus of the delivery by selecting only some indicators. I call them the critical performance indicators. So, a selection of indicators is tagged as critical. The number of critical performance indicators per Specific Agreement is at least 5 and at the most 20. The critical KPIs that are applicable at the start of a Specific Agreement must be described in the annex to the Specific Agreement.You can think of generic downtime of a system than judging the downtime per call.

In order to make changes in the allocation of critical KPIs, you can agree to initiate the change procedure with a transition period of 3 months. Changes to the critical KPIs are limited to a maximum of once a year. The number of critical KPIs to be changed will be reasonable, with a maximum of 50%.

Compliance and Security Consultation

It is also necessary to discuss the theme of complianceto security. At least this should cover legal matters, data security, system security, and compliance to the contract. You can discuss these topics during the tactical consultation. In reality, you will find out that this topic belongs to a specialist such as the security & compliance officers. It is advisable to organize this separately and let them report the summary during the tactical consultation that immediately follows the next normal consultation. Every company has a generic set of these requirements. The Global Sourcing Team can take care of that as part of its joint responsibility. The complianceand security consultation concerns coordination, information sharing, and making (work-related) agreements. The execution is based on a yearly audit plan. Also for business continuity, a yearly refreshed Disaster Recovery Plan must be drawn.

Complaints and Escalations

Make sure that there is a predefined generic complaint and escalation procedure next to the normal governance. Also, the Global Sourcing Team can play an important role here.   A complaint is an expression of dissatisfaction about the services and/or the staff provided by the supplier. Over time, when the relationship grows, make sure that your supplier also has room for handling complaints which are caused by your organization.

Only a few people are authorized to complain or escalate;for example, senior management, a business line manager or a vendor manager is authorized to submit one or more complaints to the supplier's designated representative. So, when you have a complaint and escalation matrix, make sure that there is a procedure for the supplier to deal with complaints.For example,

- The supplier tells the manager concerned within three working days about the actions that the supplier will take in order to eliminate the cause of dissatisfaction.
- The supplier will send a daily report about the follow-up.

The reason is that the sooner you are able to identify and solve possible irritations and/or complaints, the better will be the growth of your relationship, mutual understanding, and mutual performance of the outsourcing.

Contract Change

To ensure that the contract is a living document, make sure that there is a contract change procedure in the Master Service Agreement. Changes to the contract can be initiated by both the parties. Changes tothe Master Service Agreement or the Specific Agreement including annexes and schedules are only valid when agreed upon according to the change procedure. Neither the supplier nor the customer should oppose a reasonable proposal for a contract modification and should create such a Contract Amendment without unnecessarily delay.

Secure Continuous Improvement

The supplier will provide continuous quality assurance and quality improvement. The identification and application of proven new techniques and tools from other customers may benefit you, either operationally or financially. It is good to reach a mutual arrangement to implement programs, practices, processes, and measures so that the services are performed in accordance with the Agreement and that the service levels are improved. For each service level (both critical service levels and key measurements), the parties have established minimum performance standards (Minimum Service Levels) and higher performance standards that you expect the supplier to achieve (Expected Service Levels) because they can build onthe quality through their work experience.

Addition, Deletion, and Modification

You may add or substitute new critical service levels and key measurements. For example, such additions or substitutions may occur in conjunction with changes to the environment and the introduction of new equipment or software or means of service delivery.However, when such equipment or software or such means of service delivery is a replacement or upgrade of existing technology, there will be a presumption of equivalent or improved performance. All new service levels will be quantifiable, measurable, and objective, and have an Expected Service Level and a Minimum Service Level.

What if I have implemented all this and it still goes wrong?

OK, you have implemented multi-level governance and performance indicators and it does not work for you. Remember, having this implemented is just thefoundation. Despite this, there are some classical traps:

1. The mouse is dancing with the elephant. If you are a small company, do not select a large sexy company with a well-known reputation. You will never be of any importance to that large enterprise. Select companies of equal size and ensure that the volume of your outsourced activity is of importance to that company. Then you will receive the focus you desire.
2. To let your supplier outperform, it requires a lot of social skills. It is probably contradictory that, to be satisfied with your supplier,you must take the first step. After all, outsourcing is people

business. Treat every individual working for the supplier as an important colleague. Let them participate in your success and failures. But, do not forget to enjoy success together.

3. Penny wise pound foolish. Sometimes,claims discussions occur because of details. Remember your objectives. Make sure that your operational people understand your objectives for the partnership. One little claim can cause a lot of damage to people's motivations to outperform to achieve your objectives. I have never seen a company where the objectives are met by receiving a claim!! So try to create an atmosphere of zero claims. That will be a 'give and take' situation for you and your supplier.
4. Meet people and build a relationship. If you understand each other and like each other, you will put in additional effort. Plan activities of mutual interest into your trip. For example, if you are travelling to the Far East, plan on going to a cricket match or seeing the country together. Of course, it costs time and probably an additional night in the hotel. But it will pay back.
5. Evaluate the collaboration once in a while. Organize an offsite event with all key people from both parties and discuss the areas for improvement. Of course you can combine it with point 4 to be more efficient.

Summary

Ensure that you have multi-tier governance in place. That governance must be organized by a central team or person (depending onthe size of your company and the amount of work you offshore). Besides the formal construction of the governance, make it a people-centered business to become really successful!

About Erwin de Bont

Erwin de Bont has over 20 years experience in successfully managing many aspects of the Telecom and ICT Industry. His cup of tea is realizing strategy. In the past, he has successfully played operational and commercial leadership roles at board levels (Consulting, Application Management & Outsourcing) of companies such as Stork, CSC, Royal Burger Group and KPN. Erwin also started and built from scratch the national mobile telecom support organization for security services like police, fire departments, and transformed different ICT organizations and ICT environments as CIO. Because of his result-driven, multi-disciplinary approach with a commercial touch, he closed many multi-million deals and managed multiple large outsourcing programs.

During all his engagements, outsourcing or insourcing played an important role. He has built up a track record of outsourced and offshore engagements which are recognized both nationally and internationally. This track record demonstrates excellence fromthe suppliers' point of view as well as in representing the customer organization. Erwin's outsourcing experience covers multiple business areas (ICT, HR, finance, sales, delivery, program management, logistics, and production).

Chapter 6 - How to make the waterfall model work in a multi-shore setting

By: Anuj Kumar and Andreas Brilling

"We are short on people, can’t we just quickly offshore part of the work?"

"Why did the Offshore team not deliver what we wanted, when they had the specifications in time?"

"We are doing simple, old fashioned waterfall; why can’t we get even that to work with the offshore team?"

These are some common questions teams face or have to deal with when working with multi-shore projects, especially if they have not yet worked together. Teams often assume that using a particular methodology or approach will solve all issues that plague such projects, and they do not have to change or align their current way of working in order to make it work. There are many methodologies or practices used these days, each with its own benefits over others. However, none of them teaches that a successful multi-shore project can be executed without the involvement of all shores.

Some Context

We had completed one of the major releases of a project which was multi-shored a few years back. The project was based on the waterfall model and one of the major components was developed from offshore. Though the project was delivered, a lot of rework was needed by both the shores for various reasons.

Post the release, key stakeholders from both shores gathered for a workshop to analyze the experiences, learnings, and suggest improvements. One of the key findings was that the premise of working as separate teams in thoughts, decisions, and involvement resulted in overheads and rework. The common approach of throwing work "over the fence" had not worked all that well and had resulted in a few quality issues, rework, and frustrations on both the shores.

In this article we will look at a few ground rules, practices, and techniques that can contribute to effective project execution and a win-win for all the shores. Furthermore, we will elaborate on additional practices that can be introduced to make the waterfall model work in a multi-shore project, if for whatever reason it was chosen as the delivery model. In our case, this model was chosen because the customer was used to it and expected their fixed price projects to be run that way. Our customer was also used to elaborate specifications written in German which had to be translated into English for the offshore team. This is a challenge in waterfall organized projects and an even bigger challenge in a more agile setup.

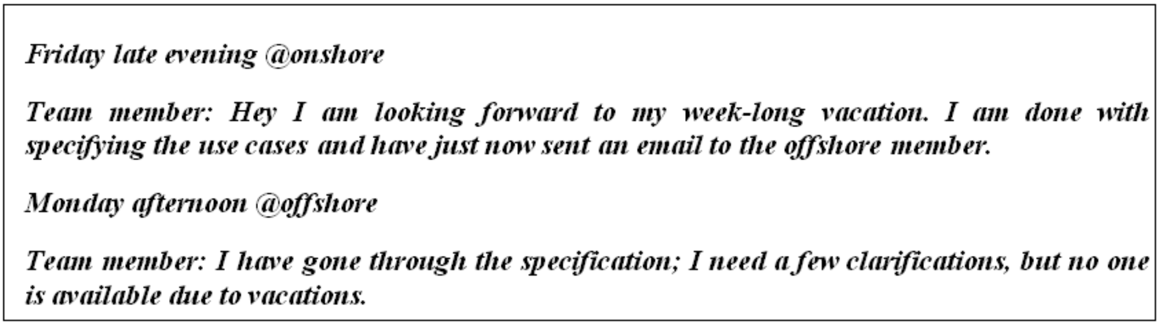
Basics: One Team, Communication, and Collaboration

Generally, multi-shore teams are not directly involved in the decision to have a distributed project set up, especially if the organization has little or no experience in working with a multi-shore team setup.

These decisions are initially taken to reduce cost; however, it should not be the only parameter to opt for multi-shore. Since the involved teams are not well informed, they are not prepared for such a setup, resulting in a lot of confusion, rework, miscommunication, delays, and ultimately an impression that multi-shore does not work with a particular method like waterfall. More often than not, teams start with some apprehensions or excitement without proper homework on the basics, perhaps because they feel ‘since it’s working for others it will work for us’.

Multi-shore projects are not different from single-shore projects when it comes to software development life cycles, all the aspects need to be extended and aligned with multi-shore teams taken into consideration. There are some building blocks which, if adopted early in the project, enable a better alignment of teams and results in better experiences on both shores. We will look at the three prominent building blocks.

One Team



Friday late evening @onshore

Team member: Hey I am looking forward to my week-long vacation. I am done with specifying the use cases and have just now sent an email to the offshore member.

Monday afternoon @offshore

Team member: I have gone through the specification; I need a few clarifications, but no one is available due to vacations.

There are cultural influences in a multi-shore project ecosystem (for example, country, education, and organization) which make the setup different and at times difficult. It is also a challenge to build a team culture from the very beginning. However, it is always possible to create a culture in small steps, by understanding the differences and narrowing them down towards a common way of working.

Though teams are multi-shored, they are working towards a common goal and should work as One Team. This means they need to be involved in all the discussions or decisions which impact the overall project. Release planning, changes to the core components due to changes in requirement or feedback from client, next code merge, dependency management, vacation planning, changes to the design are examples of topics that are often discussed and decided in the onshore team and the other shores miss out on these discussions and decision making processes.

Another perspective to consider is that apart from the "what" or "how" aspect of a particular application development, the "why" needs to be an integral part of a software development project. Unlike other industrialized sectors where the "why" is more or less fixed or known, in software projects it varies and is very contextual. If all of the team members are not well informed of the "why", it is most likely that they might not be able to revise the same while designing, implementing, and testing the developed components or modules they are working on. Educating all the team members on these aspects might sound like an overhead but in the long run it is very effective and beneficial for the end product.

The major principles for a successful One Team setup are:

- Fostering and nurturing a relationship
- Sharing responsibilities
- Mutual trust
- Accepting offshore as an extension of your onshore team
- Cultural amalgamation of thoughts
- Understanding that every issue has a resolution
- People matter just as much as the concept of offshore and onshore

Communication

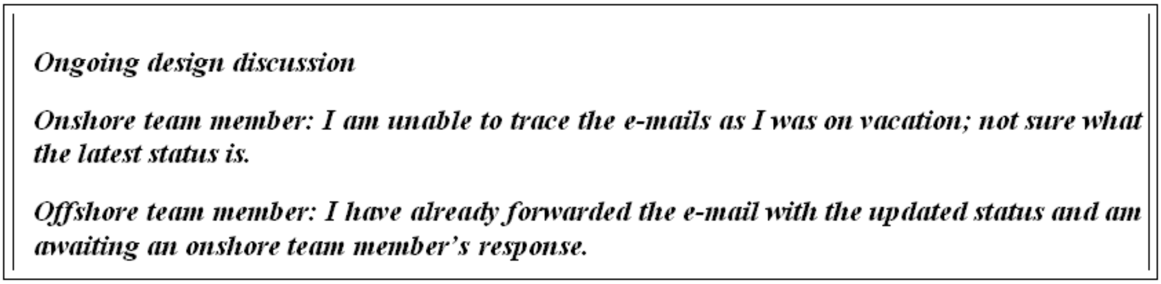
There is no escape! One needs to communicate more or less between teams, irrespective of the approach or method. An important aspect is to have the right level of communication for it to be effective. It does not mean that there is a need for too much communication among the team members. Communication must be channeled. With proper use of collaboration tools and techniques, it is possible to ensure that all the team members know what they need to know.

Some of the common mistakes are:

- Conference calls with hierarchical mode of conversation.
- Conference calls with only status reporting as the main focus.
- Only a few team members (and always the same members) on both sides attend the calls.

Communication should involve video conferencing, chats, and other modern techniques, instead of just voice-based communication. Face-to-face communication is always the best. Travel plans must be made on both shores for key roles and other members during various phases of projects. Many times, we have experienced improvement in communication after such visits.

Collaboration



Ongoing design discussion

Onshore team member: I am unable to trace the e-mails as I was on vacation; not sure what the latest status is.

Offshore team member: I have already forwarded the e-mail with the updated status and am awaiting an onshore team member's response.

Effective communication is no doubt an important building block for a multi-shore setup. There is an equally important second aspect of collaboration which is bundled in daily activities of the team. It requires an implicit communication between teams. It enables an assembly-line view of the project which is transparent and seen in the same way by all teams. This is primarily based on the choice of tools which enables the team to effectively share information such as design, code, test reports, reviews, issues, etc.

The key tools to enable collaboration are:

- Wiki-based tools for sharing information
- Issue tracking tools to track clarification, comments, improvements, and bugs
- Dashboard tools to get a holistic view of the project with key drivers being reported
- Activity stream tools work like a live wire for the project
- Automated builds and dashboards for generating and viewing reports and status

Apart from e-mails and other mandatory reporting mechanisms which have been mutually agreed upon, there are many other collaboration techniques which enable more freedom and foster team spirit.

These collaboration techniques can be broadly categorized based on the channel they use:

- Synchronous channels
  - Chat for quick 1:1 and group discussions
  - Voice/ video calls
- Asynchronous channels
  - Wikis for reference material and best practices
  - Jira for all ongoing discussion items, visible to all team members
  - Dashboards (reporting, Kanban, testing walls)
- Canteen or coffee break channel.

We all know how special canteen or coffee break conversations between co-located team members are. With the advent of Yammer/ social networks, it is possible to have a similar channel between dispersed teams.

Channel-based techniques, as mentioned above, make communication and collaboration better and effective. Their achievements are:

- Encouraging sharing of best practices
- Providing space for creativity and thinking
- Providing opportunities to connect
- Reinforcing purpose (why) and sense of progress
- Conducting quick surveys (likes/ dislikes) or feedback channel

Some usage examples are:

- Posting matters that are too small or too frequent for e-mails
- Sharing personal or work related updates
- Sharing insights and ideas
- Sharing successes and acknowledging peers

However, be careful; it can have disadvantages too:

- Considered as replacement for e-mails
- Used for urgent problem solving
- Expect immediate attention and response to messages
- Used for task delegation

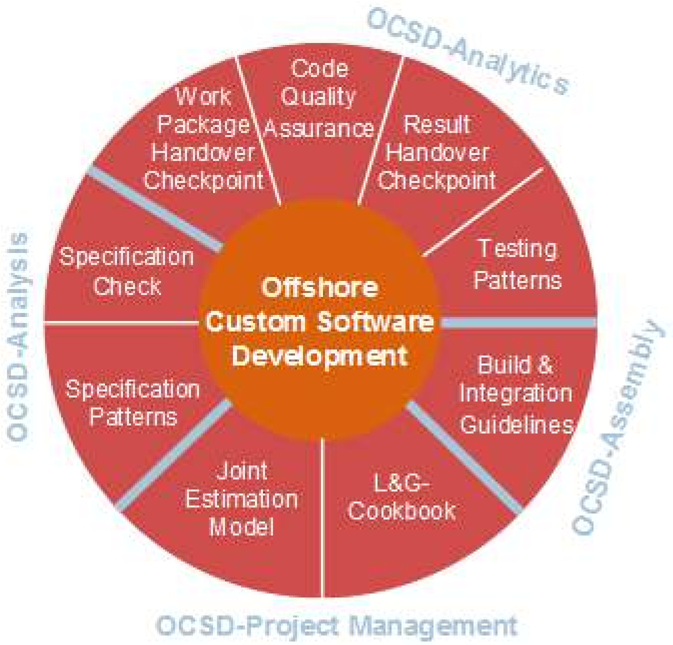
Using modern day techniques across shores enables culture bonding, i.e. it allows the teams to grow a project culture over a period of time and enables them to have a cohesive environment. The selection of techniques and tools should be left to the team. There should not be a prescriptive recommendation about which tools and approaches should be used by the team.

Specifics of the Waterfall Model

So far we have talked about the basics of One Team, communication, and collaboration. In this chapter, we would like to concentrate on the amendments to the waterfall model required to prepare it for multi-shore delivery.

We have established the OCSD (Offshore Custom Software Development) method to amend our PMI-based project management method and the well-known RUP (Rational Unified Process)

development method. The OCSD components can be thought of as an augmentation of PMI and RUP. In this chapter, we will introduce the most important ideas of the OCSD components. An overview of all the components is depicted in the following graphic.



Joint estimation model

The estimation established during the bidding process will be a core part of the subsequent offer, and is as such the basis for the development of the whole project. To a very high degree, the accuracy of this estimation will determine the success or failure of a project. If some parts of the system are developed offshore, new challenges come into play. Probably one of the most critical issues is the usage of different estimation processes or models, e.g. expert estimations and use case or function point estimation methods as demanded by the onshore and offshore organization. Often, the estimation for the parts to be developed offshore will be re-estimated by the offshore development team, after the bid is won, simply because estimation divided across shores does not always work under time pressure during the bid phase. If the estimated effort differs from the original estimation, the differences in the estimation methods make it very difficult to judge the differences of the estimations. This kind of opaqueness at a very early stage of the project can undermine trust and relationships. It is important to arrive at estimates that both teams believe in and commit to.

If the preferred estimation methods are different, make an effort to establish a joint estimation model by agreeing on one method. If that is not possible, establish a mapping between the different models, allowing each team to use its own well-known method to derive the estimation, assuring transparency and comparability between the different methods. As a result, the onshore and offshore team members believe in the estimates and can thus commit themselves to deliver within the estimated effort. Since both shores can still use their preferred methods, you can avoid the feeling of being ‘forced’ to use another method and use this to create a feeling of trust within the overall team.

Specification Check

A system specification is the central information hub of a custom software development project following the waterfall model, regardless of whether it is conducted in a co-located or a distributed manner. It is therefore crucially important that all stakeholders have a common understanding of the concepts and terms of which a system specification is made up.

The ‘Specification Check’ surfaces differing understandings of core concepts in system specifications and evaluates whether the structure of the system specification is appropriate for the different stakeholders. A couple of new challenges can arise when system specifications are written onshore but implemented offshore.

- The understanding of core concepts (“what do you understand by ‘use case’?”) can differ substantially.
- Sometimes, more details can be necessary for the offshore development team since the developers did not take part in the specification process and did not talk directly to the customer.
- The structure of a system specification might hamper effective knowledge transfer. Over-structuring a system specification as well as using a completely flat system specification should be avoided.
- Traces between specification artifacts or parts of a specification document can be insufficient from the offshore developer perspective (e.g., missing links between use cases and the data model), as well as from the management perspective (e.g., not allowing impact analysis).

Within the Specification Check (SC), these issues are assessed jointly by the offshore and onshore business analysts:

- Check of definition of core concepts and terminology used to structure the system specification.
- Check that the system specification will enable effective knowledge transfer. Over-structuring specifications can lead to a pigeonhole effect, where information is split at a too fine-grained level

and scattered over the whole system specification. This makes it difficult to restore the overall context of information.

Please visit the Bridge Knowledge Center to find the complete specification checklist: OCSD-Specification-Check.

Work package and result handover checkpoints

The handover checkpoints and the use of cohesive and self-contained work packages effectively mitigate core problems with structuring a distributed development process. The most important benefit of the checkpoints is increased communication between distributed team members. Two organizations that work together in a geographically separated manner require a precise concept for bundling and executing work. 'Work packages' consist of three kinds of artifacts:

1. Software requirements specification artifacts: A cohesive bundle of use cases, user interface descriptions, domain objects, and other software requirement artifacts. We call such cohesive requirements sets 'conceptual components'. Other specification-related information is also bundled into a work package:

   a. 'Tracing matrices': These are used to trace the software requirements specification artifacts back to the high-level requirements.

   b. Specifications of functional test cases.

2. Software design artifacts: Usually, a subsystem consisting of a couple of modules realizes the conceptual component of a work package. A work package will thus also contain the external interfaces of these modules (i.e., the external and more 'technical' view on the subsystem which realizes the conceptual component), as well as the internal high-level design of the modules, for example, which layers will be used.

3. Project management artifacts: The 'software engineering' artifacts, as mentioned under 1) and 2), are complemented by also bundling management-oriented artifacts into a work package. Some examples are the list of work units which are bundled into the work package (like the list of use cases, business rules, domain objects), as well as the schedule for the work package. Finally, the definition of quality objectives and acceptance criteria for the work packages are included.

This notion of work package supports:

- Precisely defined artifacts to be delivered by both teams on all the shores
- Clear handover points and handover processes
- Clearly defined acceptance criteria for all work packages

The above description may create an impression that work packages are bundled by the onshore team and then 'thrown over the wall' onto the desk of the offshore development team for implementation. This is not the case by any means. In fact, the opposite is true. From the very beginning, the offshore team is heavily involved in the bundling of work packages throughout the project lifecycle.

Before a work package can be implemented, the so called ‘Work Package Handover Checkpoint’ (WPHC) is applied. It is used jointly by onshore and offshore team members. After the implementation of a work package, the result will then be checked in the so called ‘Result Handover Checkpoint’ (RHC). In between these two mentioned checkpoints, comprehensive code quality assurance is applied. For example, continuous integrations, systematic code reviews, and automatic checks for architecture compliance (see also the chapter on code quality assurance).

The WPHC contains two kinds of evaluations – initial pre-implementation check and the check of specific work packages.

1. A) Initial pre-implementation check

- Overall goal: Ensure that the cross-cutting ‘production means’ (i.e. an appropriate description of the architecture to be followed, the setup of the configuration management processes, and build processes) are in place and mature. This check will not be applied to single work packages. It is applied to the ‘production means’ which will be used to implement all the work packages.
- Sub-goals:
  - Determine that the high-level requirements and the basic specification artifacts are precisely defined and a baseline is established.
  - Ensure that the high-level architecture and development means are in place and mature.
  - Ensure that the necessary processes (for example, change request management) are defined and agreed upon.

1. B) Work package check

- Overall goal: Ensure that implementation of a specific work package can safely start. As opposed to check 1A, this check is applied to each work package.
- Sub-goals:
  - Ensure that the content which will be implemented within this work package is clearly defined.
  - Determine that the internal design of the modules in the work package is defined.
  - Ensure that the implementation of the work package is planned, scheduled, and staffed.

Click here to find the complete work package handover checklist: OSCD-Work-Package-Hando-er-Checkpoint.

Result handover checkpoint (RHC)

The RHC encompasses only one kind of check.

- Overall goal: Assess whether the result of the work package implementation is finished and completed according to the defined quality objectives.
- Sub-goals:
  - Ensure that the implementation result complies with the software requirements specification, the low-level design, and the architecture.
  - Ensure that the result is complete and complies with the (previously) defined quality

objectives.
  - Determine whether the estimation for the work package was correct or whether a systematic error lurks in the estimations.

It is worth noting that contrary to work package handover checkpoints, result handover checkpoints are usually applied in agile approaches as well. There, they resemble applying the 'definition of done'.

Taken together, the notions of 'Handover Checkpoints' and 'Work Packages' support clear handover of responsibilities based on precisely defined work packages, allow earned value analysis, align the engineering disciplines, and ensure effective knowledge transfer in offshore projects.

Click here to find the complete result handover checklist: OCSD-Result-Handover-Checkpoint.

Code Quality Assurance

'Code Quality Assurance' provides guidance on how to make code quality objectives explicit and measurable. It provides guidance on how to introduce constructive and analytical code quality assurance techniques into offshore projects. Every software engineering discipline is to some extent impacted by the added challenges of offshoring. This also holds true for code quality assurance. The obvious challenges of geographical distribution and differing languages are:

- Decreased awareness about who is working on which piece of code, leading to possible inconsistencies of the code base.
- Difficulty in transferring functional knowledge, leading to misinterpreted requirements and specifications, and finally to wrongly implemented functionality.
- Difficulty in disseminating architecture and design rules, which can lead to non-compliance of the code to the architecture.
- Differing Engineering backgrounds lead to different understanding of code quality. The stronger industrialization ('more' roles and fine-grained division of work) can decrease the understanding of the overall system by the single engineer. Taken together, these issues can lead to specific issues like increased amount of 'code duplicates', and more generally, to non-uniform code quality of the whole system.

The 'Code Quality Assurance' method increases transparency into the maturity and quality of the whole code base – continuously measured, at all times. This means that the whole team knows where they really stand. Further, it supports an objective quality judgment by architects, projects leaders, and quality managers by using defined metrics, prepared heuristics, and experience values. This fosters a unified understanding of what 'quality' means in the project. Finally, it standardizes views of code quality metrics, considering established best practices. This allows effective dissemination of commonly agreed quality expectations.

Click here to find the complete code quality assurance checklist: OCSD-Code-Checklist.

Blend in New Techniques

Though the waterfall model has been in existence for a long time, there are quite a number of new-age and agile techniques, which if used effectively, enable a better execution of multi-shore projects. All the below techniques can be integrated in a waterfall approach.

Continuous integration

Continuous integration is a practice which enables early assembling of code units to a common shared mainline and enables reflecting on code quality, unit tests, and other kinds of measurable parameters.

This can be configured to a suitable frequency on a daily basis and the results can be taken to dashboards and development walls placed on all the shores.

This enables early detection of integration issues caused by design errors or misunderstandings, especially if development is carried out on both the shores. It can also provide a cockpit view to the ongoing project development activities.

These days, most projects use it in one form or the other; however, the degree of leverage varies. Dashboards on LCD TVs in a project area are a nice way to represent the current build status and compliance reports to the agreed quality indicators. We have seen that this is very effective in multi-shore project environments as it brings in transparency and current reporting.

Common design and review sessions

With the help of modern day's communication and collaboration tools like video conferences, Yammer boards, and Jira issues boards, it is possible that both teams can discuss and review the design for key components. This enables clear comprehension of problem statements, design, limitations, and common roadmaps for design evolution.

Teams on both shores will have an increased sense of ownership and the "over the fence" paradigm is removed to a large extent. Such sessions can be planned or ad-hoc; once a team starts finding such sessions useful, it tends to use it more often than spending time in writing a lot of sentences on design challenges.

Scrum-like organization of the offshore team

In order to foster ownership of the offshore team for their piece of work, the team should be responsible for organizing their share of work without involvement of the onshore team. One way of achieving this on the level of all team members, as opposed to just the offshore project manager, is to introduce a scrum-like organization of the work assignments in the offshore team.

The work packages agreed to be delivered from the offshore team become the scrum backlog with priorities derived from complexity and risk assessment (do the difficult, high-risk work first). In sprint planning, all team members can voice what tasks they would like to work on. Daily scrums can be an efficient way to identify obstacles that team members face and that need an architect's attention – offshore or onshore.

Sprints delivering working software will increase the trust of the onshore team on the capabilities of the offshore team to deliver. Concentrating on the difficult, risky work will on the one hand reveal challenges early and on the other hand, give enough time to find solutions.

Separation of work

Another important aspect is the distribution of work in case of multi-shore teams.

There is no thumb rule apart from the fact that local teams should be able to work independently and that dependencies across teams are identified and clear. This could mean that roles such as Business Analyst or Technical Architect will have to co-exist with local teams. It might look like an overhead but experience shows that it pays off in the long run.

Some of the patterns for distribution of work between teams in a multi-shore project are:

- Based on key SDLC phases (Requirement Analysis, Design, Implementation, and Testing)
- Based on functional components
- Based on a mix of SDLC phases and functional components

Conclusion

Many of the approaches and techniques described in this article are useful or even mandatory regardless of the delivery approach you might choose. In large teams (both on the client as well as on the development team side), waterfall is still a very valid approach to assure success. With the underlying principles of the OCSD method, we hope we have conveyed what it takes to make waterfall work in an offshore delivery model. Further improvement and even more efficient delivery can be achieved by blending in more modern techniques. The agile approaches which are becoming more and more important in modern software development, offer a vast set of ideas for such approaches. You have to pick, experiment, and chose what proves to be the best working set of techniques for your specific project. Good luck!

Note

Observations and recommendations mentioned in this article are purely the author's own views and are not related to and non-binding to the organizations they are associated with.

About Anuj Kumar and Andreas Brilling

Anuj Kumar

Anuj works as a Senior Manager with Capgemini India, based out of Mumbai. He has been working with custom software development based projects for the past 14 years. Most of his projects involve multi-shore teams. He has been associated with Capgemini India for the past 10 years. Prior to Capgemini, he has worked with NIIT Technologies. He has a wide range of experience working with clients across different sectors such as Finance, Telecommunication, and Automotive from countries such as USA, Germany, Netherlands, and Norway.

Andreas Brilling

Andreas Brilling is an Engagement Manager at Capgemini based in Stuttgart. He has more than 20 years of experience in software development projects in various international settings. For 12 years, Andreas worked for Hewlett Packard and Agilent Technologies before joining Capgemini. He has worked in and has led multinational teams in Germany, US, as well as in Australia. He has been responsible for the first major offshore project of Capgemini Germany utilizing Capgemini's extensive Indian workforce. Later, he used his experiences to broaden Capgemini Germany's offshore capabilities in custom software development. In that responsibility he has been an important driver for the creation of the OCSD (Offshore Custom Solution Development) methodology, which enhances Capgemini Germany’s development method to cater to the challenges of offshore delivery. He has coached many project managers in achieving offshore delivery success and has built together with his team, the Capgemini internal training “One Team Offshore Training”.He has personally delivered it numerous times to German employees as well as German-Indian mixed teams.

About Capgemini

With more than 130,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2012 global revenues of EUR 10.3 billion. Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business ExperienceTM, and draws on Rightshore®, its worldwide delivery model.

Rightshore® means working with the right people, at the right location, at the right time tailored to the needs and for the benefit of our customers. Capgemini Germany takes advantage of its nearshore development center in Poland as well as its offshore development centers in India.

http://www.capgemini.com

Chapter 7 - Combine Customer Service with Global Operational Excellence

By: Jean-Paul van Wieringhen Borski and Herke Schuffel

Last summer, my mobile subscription ended. So I did some research and made my choice entirely based on price. I found the cheapest provider. All the interactions that led to the contract were conducted through the website of this provider. I had to fill in all my data and the contract was sent to me by e-mail. Very smooth and efficient!

The new contract was supposed to start on July 15th and the SIM card would be sent to me exactly one week before that. So far, so good, isn't it?

July 8th was a day before my holiday started, so if the SIM card arrived one day late, I would miss it and be without a mobile connection after the 15th. So I tried to contact the provider to have the SIM card delivered earlier. However, there was NO way to influence the process, no human interaction was possible. I could only chat with a 'bot' through the website, but the artificial intelligence didn't understand my question. I was very frustrated. But in the end, the SIM card was delivered exactly on the promised date.

How can we translate this experience to IT? I chose 'low cost' and I got 'low customer service', 'no flexibility', but 'high operational excellence'. What I really wanted was low cost, high customer intimacy, high flexibility, and high operational excellence!

We have learnt that in delivering IT services there is no single solution to solve this puzzle. We have to be dynamic and agile and make use of the worldwide availability of resources.

Natural Geographical Selection

Usually, work for a customer is done onsite in case of managed sourcing or in environments that demand high security or maximum customer interaction. In other cases, we work from our onshore location to ensure good customer interaction (front-office roles), while operations take place onsite, nearshore, or offshore. Onshore availability in the team is always needed for local domain expertise, customer interactions, and language or legal constraints.

To extend our local capacity, we have selected a worldwide partner network for other (out)sourcing parts of the work done for our customers. We have differentiated the characteristics for our nearshore and offshore partners to have maximum match on the different customer demands.

Our partners for nearshore are:

- Relatively small (max. 500 FTE)
- Have the same CMMI levels as our customers
- Contracts are capacity based

While our partners in India are:

- Huge (min. 50,000 FTE)
- Have high CMMI levels
- Contracts are service based

Our choice is based naturally on the above characteristics. Until now, there was no preferred solution for our customer where both the options were applied.

> *Once, I was eating in a small restaurant in Pune with my Dutch colleague. My colleague went to the washroom and after 10 minutes he returned and said: "Do you notice that I do not wear a t-shirt under my sweater anymore, as there was no toilet paper. "Only a bucket of water!" The next day, I told my Indian colleague about our experience. He laughed and said: "Listen JP, if your car is dirty, you clean it with water and not with paper, so why do you need toilet paper?" He was right…*

The secret of having a successful global team starts with believing in its success and selecting the right people. I have noticed that most successful teams have key players who like to work in a global environment, are open-minded and not protective, and like to learn from different cultures. These key players are used to the fact that our near or offshore team members are really young compared to our local staff and the teams that have a more equal mix of men and women. The second most important key factor is trust and respect. As in all relationships, trust grows over the years. Invest in the relationship! In the years that I worked in India, I have noticed that most of the telephone calls from our Dutch team to India took place when something was wrong or needed. Hardly ever when something was implemented successfully! The trust grew when we started to call each other everyday. We came to know about the families, religions, sports, etc. We also started cricket sessions where we shared the amazing world of cricket and invited the teams for dinner where every team cooked its local specialty. Working within a global team enriched my work, life, and it is also fun.

> *To get fully integrated in India, I wanted to have my name in Hindi on the company business cards. So I asked my Indian friend to write down my name in Hindi on a piece of paper. Over the following days, I asked different people to pronounce what was written down. It was pronounced perfectly, although I have a very difficult name: Jean-Paul van Wieringhen Borski. After that week, I was sure that the Hindi words were indeed correct. So I ordered 500 business cardswith a scan of my written name on one side. Customers and co-workers were happy to see the new business cards. After a couple of weeks, I started wondering why there were six words in Hindi, whereas my name has only five words. Again, I asked people to pronounce this and they still did perfectly, until someone noticed that one part of my name was repeated: 'Jean-Paul van Wierin Wieringhen Borski' was printed on 500 business cards!*

Nearshore

In contrast to India, Eastern Europe is remarkably near – a day's drive to the east, or 'just past Italy' to the southeast. The days of the big political divide are long gone; over 20 years ago already. Even the countries of former Yugoslavia have lived in peace for over a decade.

As the economies are growing, the cultures are also leaning towards the West. This is especially true in IT. Working in IT has a special status in Eastern Europe. The brightest kids, both men and women, study IT and work in a market with low unemployment to earn salaries that are much higher than other people with university degrees. They often work for Western European customers. As a consequence, they travel to European countries ever so often and absorb parts of our life and customs, and gradually bridge the cultural gap.

> *Common knowledge sometimes works in a different way from what is expected… One evening, over dinner, I tried to explain the concept of the TV program 'The Voice of Holland'. I did not use the name, but explained the rules. After sometime, one of our Serbian colleagues said:"that sounds just like The Voice of Serbia''… I did not expect this. On the other hand, we often used the phrase 'I love it when a plan comes together' when we agreed upon a plan with our colleagues in Serbia. A recurring phrase in the eighties TV-series 'The A-Team', I expected that they would also know this phrase. But they did not react to that. So the next time, we played the tune from the A-Team. Still nothing! It turned out that 'The A-team' was never broadcast in Serbia. I thought the whole world would know that show.*

We found that our best practice in nearshoring is to work together as one team rather than subcontract parts of the work. Modern communication devices and availability of high bandwidth enables this. We call this an 'eXtended Resource Team' (XRT).

It is remarkably easy to work with a team in Eastern Europe. They are smart, eager, understand our culture, and help us to understand theirs:

> *One of our Romanian colleagues told me that he once has received a ticket from the police for overtaking another car on a street where overtaking was not allowed. So I asked him if the Romanian police were very active in checking the traffic. He said: "No, but I overtook a police car…" "How does paying a traffic fine work in Romania?" "Well, the police take away your license plates. You get them back when you pay the fine at the police station". "So, what did you do then? Parked your car and walked home?" "No, I didn't. I just drove home without the plates…"*

An XRT is a distributed team. Members work together, but work in different locations. All activities are managed by one person on a day-to-day basis, just like a team that works in one location. No fixed-price work packages, but activities that one person can handle in one day.

XRT Principles

Based on the characteristics of our nearshore partners, we have laid out a few principles for this to work: The first is 'We are all equal colleagues'. We treat every team member as a colleague – not as a replaceable extra resource, but as a long-term colleague. So we hire all Eastern European team members from our partner company. Not for one job, but for a year or longer. This way we make sure that we treat all team members in the same way when we assign activities or deal with availability.

The second principle is ‘We are one team’. As part of the team in Eastern Europe and the team in the Netherlands, we have to make sure that both parts of the team can do all the activities. In this way, we steer away from work packages and are free to distribute the tasks on a day-to-day basis where the availability and capacity are independent of the location.

The third principle is ‘We have daily contact’. Every day, the manager interacts with all the team members. So we make sure that the team stays together and has the same knowledge available to make the right decisions.

The fourth principle is ‘We create opportunities to get together’. The team manager visits the Eastern Europe part of the team at least every other month. The whole team gets together at the start of a new project and at least once a year.

This approach combines the benefits of working with your own personnel – quick, short lines, and no back-to-back agreements for every new deal with the benefits of Eastern European rates, quality, and scalability. In this way, we can achieve low cost, high quality, high operational excellence, and high customer interaction only if the team is not too big.

Size Matters

Based on the characteristics of our offshore partners, we focus more on the results of the services of the partner and help to achieve the pre-defined goals. The ‘how’ is managed by their high operational excellence. High operational excellence is possible if we help the customer in realizing the demands from our offshore partner.

A team can only be successful if the work package is suitable for this international team. Our experience is that, in case ‘size matters’, India is the best choice. Our partners in India can easily scale big teams, while we use our nearshore partners for smaller teams. As a rule of thumb, the size of a new team will be maximum 5% of the size of the partner’s company. For instance, if your partner has 300 FTE, the ideal offshore team that can be absorbed by the partner is maximum 15 FTE. The 5% rule is the result of a combination of idle capacity combined with the rate at which an organization can grow without any trouble.

> *During my stay in India, I recruited several testers and database engineers for a new assignment. Negotiations took longer than expected. So one Friday afternoon, I asked everybody to come to the office and sign the contract in order to start on Monday. Some people got different salaries for the same job, but I was expecting no problems because I assumed that they would keep this to themselves. As soon as the first engineer walked out of my office with the signed contract, he ran to his new colleagues showing the new contract including all the details. That afternoon, I learned a new definition of transparency!*

Customer Intimacy

Today, a couple of pure Indian players have large customers in the Netherlands. Their customers are mostly large multinationals with outsourcing experience. Ideally, to service the customer, onsite presence is the key. I have noticed that large Indian companies open offices locally for greater customer interaction and understanding. This strategy works, although the offshore ratio is a little less than before, the advantages are huge; so at the end of the day, more value for the customer.

> *After a long flight from Amsterdam to Mumbai, I was sitting in the bar of my five-star hotel in the middle of the night. The only thing I wanted now was an ice-cold Kingfisher beer. I ordered one and was enjoying the pleasant environment when the waiter came to me with a friendly smile, and I told him: "Please get me some foam". He poured the beer into my bottle without the 'foam' I had asked for and walked away. After ten seconds, he returned to me with a big wireless phone and said, "Sir, you asked for a phone?" I started laughing and from then on I know that 'foam' is not correct!*

Conclusion

We all want low cost, high customer interaction, high flexibility, and high operational excellence. We have learnt that in delivering IT services, there is no single solution to solve this puzzle, but that we have to be dynamic and agile and make use of the worldwide availability of resources. Size does matter. We knew that, right? If you are small and agile then onsite in combination with Eastern Europe is our choice. If you are large and have a lot of experience, then you might want the extra low cost of India in combination with onsite presence.

About Jean-Paul van Wieringhen Borski and Herke Schuffel

Jean-Paul van Wieringhen Borski

With 17 years of experience in IT, Jean-Paul has spent most of his time in international environments where he managed several offshore delivery centers. He began his career as a software developer at AND (Automotive Navigation Data). For AND and Closed Cap, he worked and lived in India (Pune and Gurgaon).

Currently, Jean-Paul works at Ordina in different roles but is always related to Application Outsourcing in combination with near- and offshore.

He developed a ICT Global Sourcing course for Ordina employees and its customers.

Currently, he works in the Business Development department of Ordina as Product Manager, Application Outsourcing and Alliance Manager, Near- and Offshore.

Jean-Paul loves to travel in India, enjoying the people, the colorful land, and of course the food!

Herke Schuffel

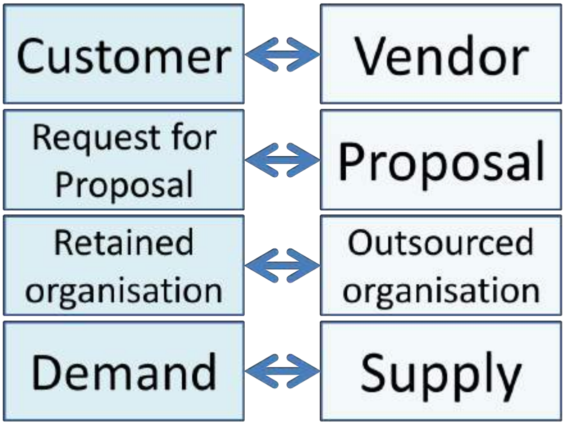
Herke has been in IT for almost 20 years now. He has spent most of this time in IT services, delivering maintenance support to customers in different branches and with technologies ranging from COBOL to modern open source technology. He is currently responsible for a business unit delivering application support on custom-built software in these technologies. In this role, he works with eXtended Resource Teams in Serbia and Ukraine.

Chapter 8 - Shipping-to-Partner or Partnership?

Author: Henk Woolschot

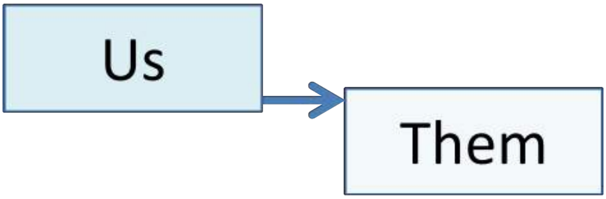
To be successful in the current business world, partnerships are necessary. Due to globalization and supply chain management, a single company cannot operate on its own anymore. But what exactly is a partner or a good relationship with suppliers?

Outsourcing: Customer versus Vendor



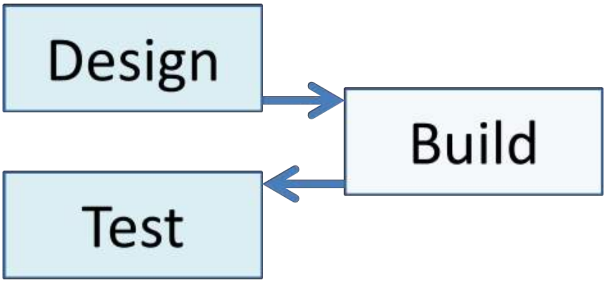
When it comes to outsourcing contracts, the discussions often shift focus on the supplier. What criteria should a potential supplier possess? Price, quality, and skills are important factors during decision making. How do we know that we have selected the right supplier? The answer to this question has to come from a smooth process, and because a lot of companies do not have much experience in this field, they hire way too expensive consultants to support them. At a certain point, they realize that this outsourcing can have a large impact on their own organization, and there is an increased urge to understand this process. This is how demand-supply is created.

Outsourcing: Us versus Them



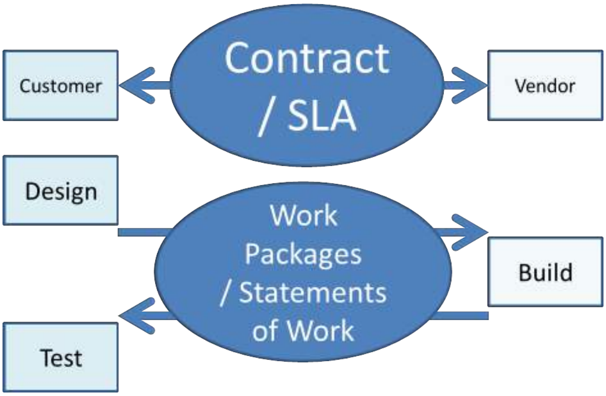
By the introduction of the demand-supply relationship, one automatically understands that the words 'us' and 'them' must be involved to a large extent. The customer's versus the supplier's thoughts create a certain distance. We know 'our' business, organization, processes, people, and tools, and 'they' don't know these (not yet). Not only is due diligence needed but also a lot of knowledge sharing sessions. The focus of these sessions is to transfer knowledge from the customer organization to the supplier. How fast are they able to do this? Since we have outsourced this, it has become their task and responsibility. ‘But they are far from being equivalent!’

Outsourcing: Design versus Build and Build versus Test



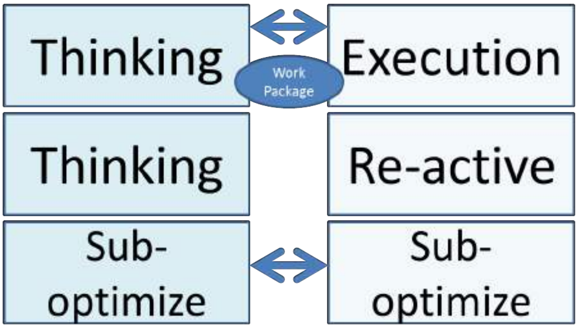
Every relationship starts with a first project. Thereby, a design-build-test model arises within the IT organization. The amount of iterations may vary from case to case. ‘Classic waterfall’ or ‘Agile’ methods are being applied around this ‘outsourcing model’. ‘The wallet’ can be stretched by letting the partner design (technical) and test (unit). And hence, the shipping-to-partner-model is born.

Outsourcing: Shipping-to-Partner



The shipping-to-partner model is further explored, and is used at the company level for making contracts and Service Level Agreements. Legal and Finance departments review the contract without any intention of ‘true partnership’. For projects, ‘Work Packages’ as well as ‘Statements of Work’ are drawn. If ‘we’ have to design and test, and ‘they’ have to construct, how can we send good working packages to them, and how can they provide us with great results? Moreover, along with this model, entry criteria and acceptance criteria are drawn, especially, *acceptance criteria*, because they are the supplier.

Characteristics of the Model



A shipping-to-partner model results in reactive behavior of the supplier. 'We' think about how and what 'they' have to do. At the point, when a working package is sent to the supplier, a lot of thinking has already been done. The supplier reviews the outcome and tries to understand this. But the gap is never caught.

The output of the supplier in this model is only the code. And often, not the needed code, but in the best case the code that has been asked for.

One downside to this model is *sub-optimization*. Of course, the 'handshake' is being watched, whereas the integral process is not. 'How we do business with the customer is an internal case'.

Risks avoiding behavior can be found in 'capturing a lot'. Financial risks are being covered by fixed-price-constructions.

Output of the Model

The output of this model is *services according to contract/SLA* at the company level or *deliverable for the SoW/work package* at the project level. In a project, this will be a code according to certain standards which has probably been tested by a unit test. But after all it's just a code, and nothing more.

Outsourcing: Partnership

A partnership model is *something completely different*. Real partners do not hide behind contracts and processes. Here, partnership can be metaphorically compared with a marriage. Most probably, you will not make a contract in your marriage specifying responsibilities; for instance, who is in charge of cooking, and what criteria the food has to fulfill. Real partners enjoy the result of their partnership together. Also in outsourcing you can enjoy good result like increased business satisfaction.

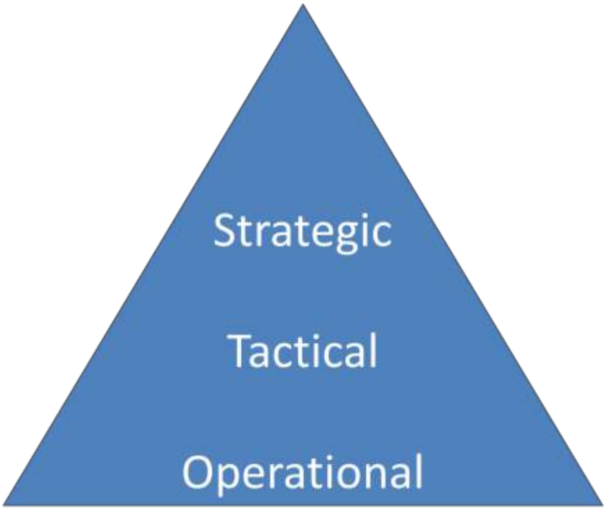
Characteristics of this Model

Real partnership starts with *trust*. Trust can have different features. Trust can be expressed through confidence that your partner will help you, even though the recent status differs from the originally made arrangement. Trust means, not being punished for your mistakes. Trust in the relationship itself helps you believe that it will last forever and not only for ‘the period of the contract’.

Another aspect is *openness* to each other. Being open regarding your qualities is easy, while showing openness about your shortcomings is a completely different story.

Is True Partnership Possible?

This question is hard to answer. How do you realize and implement a real partnership? If you make a long-term contract that follows the shipping-to-partner model, how do you communicate this transition to the partnership model within an organization? If you are a part of an *outsourced company* of a *retained organization*, you probably won’t have pleasant feelings towards the enforced partner. Outsourcing has been the decision of somebody else, and now are you supposed to go for real partnership?

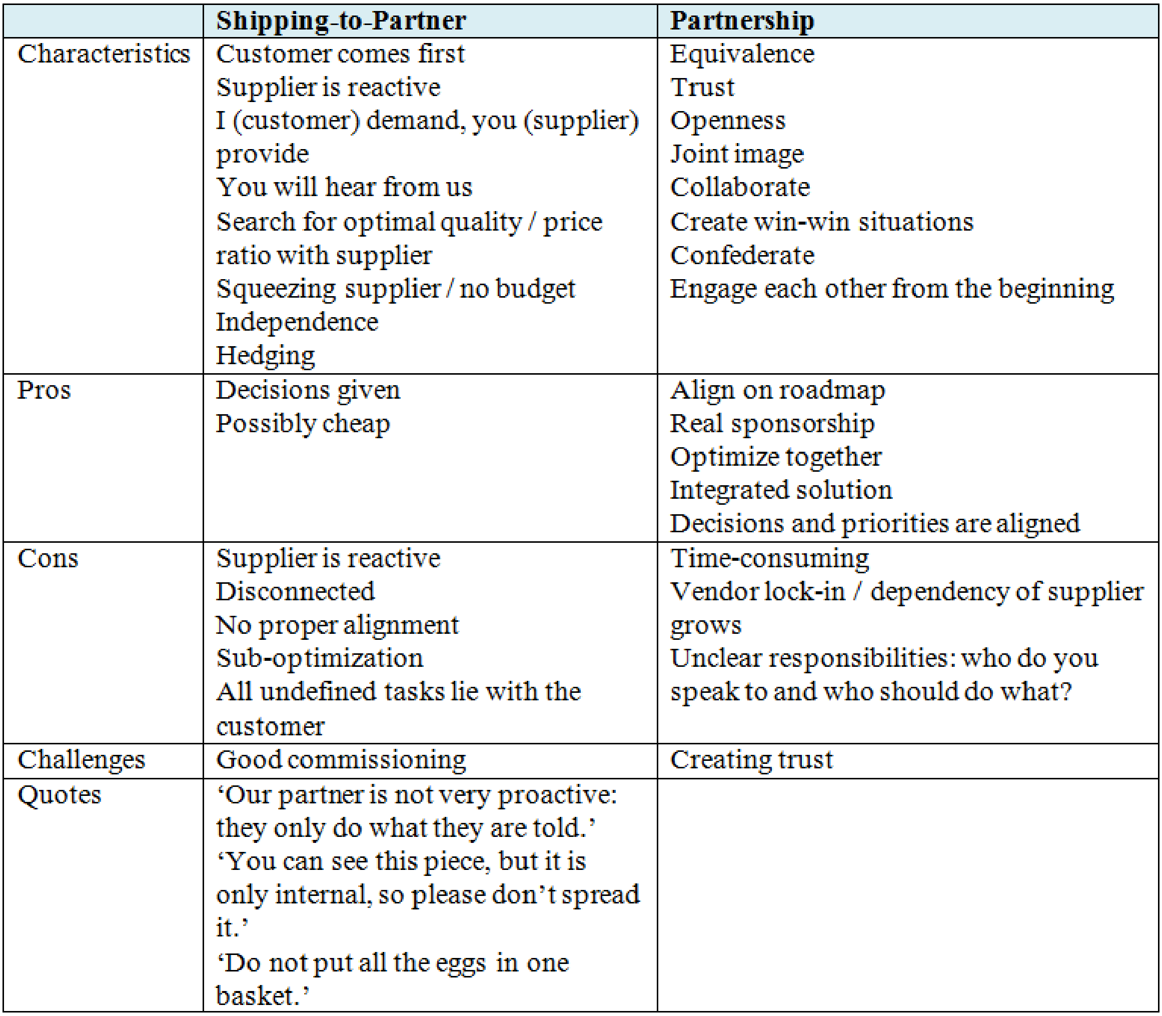


First of all, we should take a closer look at the company. Strategically, it is possible to define a partnership model. But will employees at the operational level really accept this? It feels like an *arranged marriage* and if this is supposed to become a *love marriage* one day, there are a lot of parameters that must be considered. In a hierarchic organization, acceptance is more likely, but if its structure is intrinsic, the success of the partnership model is doubtable. In more flat organizations, plenty of discussions will arise about who has to be the one true partner.

Another factor is the *'what's in it for me'*. Again, we will get different answers on strategic, tactical, and operational levels.

Outsourcing: Shipping-to-Partner of Partnership

Let's put things together:



	Shipping-to-Partner	**Partnership**
Characteristics	Customer comes first Supplier is reactive I (customer) demand, you (supplier) provide You will hear from us Search for optimal quality / price ratio with supplier Squeezing supplier / no budget Independence Hedging	Equivalence Trust Openness Joint image Collaborate Create win-win situations Confederate Engage each other from the beginning
Pros	Decisions given Possibly cheap	Align on roadmap Real sponsorship Optimize together Integrated solution Decisions and priorities are aligned
Cons	Supplier is reactive Disconnected No proper alignment Sub-optimization All undefined tasks lie with the customer	Time-consuming Vendor lock-in / dependency of supplier grows Unclear responsibilities: who do you speak to and who should do what?
Challenges	Good commissioning	Creating trust
Quotes	'Our partner is not very proactive: they only do what they are told.' 'You can see this piece, but it is only internal, so please don't spread it.' 'Do not put all the eggs in one basket.'	

Finally: Some Considerations

How to Choose from Now?

The aforementioned can be read as a guideline to help you determine what model is most suitable. Of course, no binary or algorithm can be applied to this. However, it is important to be aware of situational factors which should influence your decision about which model can be used at what time. Besides the mentioned considerations, other aspects, such as the number of suppliers needed, are crucial as well. What other kinds of relationships do I have? Possibly, you are the customer and the supplier and there is a ‘chain of partnerships’.

Do I Need Multiple Partners or a Single Partner?

Many organizations ask themselves the following question: How many suppliers do I need? Choosing *multiple vendors* or a *single vendor* is a serious decision. The reason cited most often is the ‘fear for vendor lock-in’. But does something like *partner lock-in* exist? Just imagine that you marry more than one woman / man simply to prevent a lock-in!

Who Exactly is My Partner?

A classic example is the End User-Business-IT-Suppliers connection. As a supplier you are interested in End User satisfaction. So it matters what the relationship between Business and ICT is, when you are ‘only’ the partner for the IT department. If shipping-to-partner happens in more than one step the characteristics will expand. For example ‘supplier is reactive’: if the IT is already re-active towards the business, and supplier is re-active to IT you can imagine the impact. From these thoughts of the supplier, the questions ‘Who is my partner?’ and ‘Is that the IT department or the end user?’ are important. Furthermore, will I ever become a business partner if I sign a Shipping-to-Partner contract with an IT department?

Summary

A good model always depends on the situation and can change anytime. Good relationships are sustainable, and therefore, a process is needed, and this takes years. There will be moments in which your consciousness about business will be high, but also times where this awareness is lesser. This chapter may help you choose the right model, and provide you with the necessary confidence and advice in situations where everything goes wrong.

About Henk Woolschot

Henk has been an IT expert for 23 years because of his great professional experiences in the field of outsourcing and offshoring.

Currently, Henk is Engagement Director / Delivery Head for InsuranceHCL TechnologiesContinental Europe.

He is specialized in global delivery management, outsourcing, and offshoring of application services and infrastructure services. Since 1995, Henk has gathered a lot of practical experience in setting up centers for Application Services (development, testing, maintenance, renewal, replatforming) / Application Packaging and Infrastructure Services offshore (India, South Africa), as well as nearshore (Romania, Hungary).

Henk has worked with many Indian vendors in different business / engagement models. Not only does he know the customer side, but also the ‘back-office factories’.

LinkedIn® https://www.linkedin.com/in/hjwoolschot

The Art of Managing Remote Teams

We hope you had a pleasant read and have gained some new and useful knowledge that will help you in your work and bring you success! In the near future, the next book in this series of books “The Art of Managing Remote Teams”, will be published. This fourth book will guide you through cultural aspects in managing remote teams.

To get the latest news and updates about the release, please sign up at our website:

http://bookoffshoring.bridge-outsourcing.com/

To get in touch with us, if you liked the book or want to give feedback, e-mail

h.messer@bridge-outsourcing.nl

We would appreciate your review on Amazon.