



How to

COMMUNICATE

effectively with a **REMOTE TEAM**



lessons
learned in
offshoring and
nearshoring

HUGO MESSER

ERIK SNIJDER | RAJIV MATHEW | DAVIDE CASALI
ARJAN FRANZEN | JUTTA ECKSTEIN

~~*~*~*

How to Communicate Effectively With a Remote Team:

Lessons Learned in Offshoring and Nearshoring

Written by Hugo Messer, Erik Snijder, Rajiv Mathew, Davide Casali, Arjan Franzen, and
Jutta Eckstein

~~*~*~*

~~*~*~*

How to Communicate Effectively With a Remote Team:

Lessons Learned in Offshoring and Nearshoring

Written by Hugo Messer, Erik Snijder, Rajiv Mathew, Davide Casali, Arjan Franzen, and
Jutta Eckstein

Amazon Kindle Edition

Edited by Laura Roberts of Laura's Proofs

Text Copyright 2014 Hugo Messer

All Rights Reserved

Amazon Edition, License Notes

Thank you for downloading this eBook. You are welcome to share it with your friends. This book may be reproduced, copied, and distributed for noncommercial purposes provided the book remains in its complete original form. If you enjoyed this book, please return to your favorite eBook retailer to discover other works by this author.

Thank you for your support.

~~*~*~*

Table of Contents

[Forward](#)

[Introduction](#)

[Chapter 1 – Communication Challenges in Working with Offshore Teams](#)

[About Erik Snijder](#)

[Chapter 2 – How to Communicate With Your Software Delivery Team](#)

[About Rajiv Mathew](#)

[Chapter 3 – The Tools of Collaboration: Selection and Culture Fit](#)

[About Davide Casali](#)

[Chapter 4 – Process, Rhythm, Metrics, and Feedback](#)

[About Hugo Messer](#)

[Chapter 5 – Remote Teams and Communication](#)

[About Arjan Franzen](#)

[Chapter 6 – Oral Versus Written Communication](#)

[Chapter 7 – 10 Tips for Communicating Successfully](#)

[About Jutta Eckstein](#)

[Conclusion](#)

[The Art of Managing Remote Teams](#)

[Connect with Hugo Messer](#)

Forward

If you have any questions while you are reading this book, please do not hesitate to contact me.
You can reach me via Twitter: @hugomesser.

If you do not have a specific question, but you think we should talk, you can send me an email at hugo@ekipa.co, call me at +31654230708, or reach out to me on Skype (user name: hugomesser).

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 your knowledge, please email me.

~~*~*~*

Introduction

Communicating with remote teams is challenging for most people. Communicating with people in your own country with whom you share a language, culture, and many other similarities is already challenging. With people from another country, time zone, culture, and language, it is even more challenging. With any offshore project that does not proceed as expected, people often name communication as the main reason.

Remote communication can be both enjoyable and frustrating. I still love opening my PC, starting Skype, and talking with my colleagues from India, Ukraine, or Holland. I find it exciting to share with my own development team in India the many things I have learned from talking with people who work in Latvia, Ukraine, and India. I enjoy having a Monday morning meeting with team members from three locations and deciding our strategy for the week. Collaborating with diverse nationalities to complete projects is highly rewarding. At the same time, I have also been misunderstood as well as not understood what is happening on 'the other side'. If one of my managers in India is unsatisfied, and I need to figure out what is going on, it is more difficult to resolve through Skype than it would be locally. Her perspective is also different from mine, so it takes skill and practice to understand. It is frustrating when you thought you clearly communicated your ideas for a certain function or design only to receive something that is entirely different from what you had in mind 2 weeks later.

Through practice, we learn how to communicate. This was true when we were kids, and it is still true as adults when we find ourselves in a new team with people from different locations who are using tools instead of face-to-face communication. If you focus on the frustrating part above, it becomes difficult. However, if you view it as enjoyable, you will find ways to make it work. If you are a strong communicator and/or lucky, and you have the right team that has remote work experience, communication may even work as if the team is in your local office.

In this book, we look at different perspectives for what works and what does not work when communicating with a remote team.

In the first chapter, Erik Snijder, senior manager at [KPMG Netherlands](#), writes about his practical issues with communication and methods to bridge the gap between languages and cultures.

In the second chapter, Rajiv Mathew, head of marketing at Compassites Software, explains how to communicate effectively with your software delivery team. He is a hands-on technology marketing and communications professional with proven expertise in multiple facets of the marketing spectrum.

Chapter three is written by Davide Casali, an experienced director and startup advisor with more than 11 years of experience and a hybrid background in design, psychology, and technology. He has led cross-disciplinary teams of designers and developers, created and advised successfully funded startups, and coached and trained employees of both global and local organizations (DHL, EY, Atos). Currently, he works for Automattic, the makers of

WordPress.com. Davide shares his experience on how to choose the perfect tools for collaboration from multiple locations.

In chapter four, “Process, Rhythm, Metrics, and Feedback,” I analyze the differences between communicating with a local team versus a distributed team. I present the Bridge Canvas, a tool that helps with structuring the communication.

The fifth chapter is written by Arjan Franzen. He has 14 years of experience in the software development industry in various fields within the engineering sector of software development. His industry experience is in telecoms and the financial industry. Arjan is development lead at NICE Systems in the Netherlands. He shares his experiences with communicating with offshore teams using an ‘over the wall’ strategy versus an ‘inclusive, dedicated team’ strategy.

In the sixth chapter, a short intermediate article, I focus on the impact of oral and written communication on project success.

The seventh chapter is written by Jutta Eckstein. She works as an independent coach, consultant, and trainer from Braunschweig, Germany. Her knowledge in agile processes is based on over 15 years’ experience in project and product development, and she has written several books on agile (distributed) software development. In her article, she provides a collection of top ten tips for establishing and preserving successful communications with global teams.

Hugo Messer

[Back to Top](#)

~~*~*~* **Chapter**

1:

Communication Challenges in Working With Offshore Teams

Written by Erik Snijder

Introduction

Throughout the past decade, I have worked for several service providers and then later as a consultant. During that time, I regularly interacted with offshore teams—sometimes as colleagues and sometimes as client/vendor relations—regarding outsourcing. Working with remote teams presents additional challenges that must be addressed explicitly. Overtly addressing the communication challenges faced in offshore collaboration is a key success factor for making these engagements work.

Practical Challenges to Be Addressed

Some of the communication hurdles are very practical in nature. For instance, the first obstacle that comes to mind is differences between time zones, which require advanced planning. For example, when wrapping up a workday in the US, staff members in India may no longer be available. Without explicit agreements in place, your schedules will only overlap during the early morning hours, so you should plan your regular meeting accordingly. However, time differences may also be leveraged. I once was involved in a data center migration with project members from both Eastern Europe and the US West Coast, which allowed us to support migration weekends in shifts that were more convenient for all participants.

Communicating across multiple countries usually involves many telephone conversations, which by nature are non-visual exchanges of information. Aside from the audio quality of the connection, there is a much bigger issue here: the verbal element is only a limited portion of communication, and by definition, all body language is lost over a phone line. That may seem obvious, but you often see people making all kinds of gestures when talking on the phone. Being aware of this phenomenon, you should be very explicit in your communication. Any ambiguity or disagreement should be discussed very explicitly, and it is good practice to ask for confirmation and feedback several times throughout the duration of the call. Without taking specific precautions, the chance that your message will come across clearly becomes limited.

Conference calls are even more challenging. At the beginning of any project, it is good practice to agree on conference-call etiquette. Asking participants in advance to create a quiet environment, engage in dedicated participation, clarify call time, appoint a chairperson, and devise some general rules for courtesy can work miracles in making these calls successful. You also must realize that a conference call is not the only option. In fact, it is quite often an inferior option. From personal experience, troubleshooting a connectivity issue with experts from Pune (India), Edinburgh, and Frankfurt was unsuccessful using a phone line; however, exchanging IP addresses via chat (Microsoft Communicator) increased the speed and quality of the communication by at least 200%.

Language: English Please

The common business language used in many international projects is English. In today's world, most project participants master the language to some extent. Nevertheless, some challenges need to be recognized. Even among native speakers, there is a difference among accents. Ideally, someone from Bangalore should be able to understand someone from London, Brisbane, or Los Angeles. Based on experience with these examples though, I can confirm that this is not always the case. Often, you have to familiarize yourself with different accents. By making small talk upfront, you create time to adapt to local variations, and over time, you will better understand the accent being used.

Communicating with non-native speakers is even more challenging. If you meet a senior plant manager in the south of Germany or an engineer in Paris, it may be obvious that English is not their mother tongues. Talking slowly and aligning your vocabulary aids in mutual comprehension. In some cases, your colleagues may seem to do okay. They quickly respond and engage in conversations; however, at other times, they may misunderstand vital points when making decisions. Non-native speakers may have difficulty noticing the subtleties of native speech. Again, asking for explicit feedback and confirmation can save you much trouble.

Culture as a Dominant Factor in Communication

Aside from the practical issues and language challenges, culture is a dominant factor in any type of communication. This is especially true—and something you must be aware of—when working with people from different continents and very different cultural backgrounds. An important aspect to take into account when communicating across culture is the difference between “high-context” and “low-context” cultures.

In a low-context culture, most of the message is explicitly communicated. For instance, Dutch speakers are known for being blunt, so we are on the low-context end of the spectrum. Countries such as India are on the complete opposite end of the spectrum. Much of their communication is implicit, and only a good listener (from the same culture) will truly understand the message based on its context. A message from someone in a high-context culture can vary based on the situation (e.g., the participants in a meeting or recent event). Coming from a different cultural background, this can be very confusing. Below are some generic examples to help illustrate this.

Hierarchy: A group of Dutch people will usually discuss any topic, and all participants can have a say regardless of their positions within the organization. This is not so in the Indian context where all participants will look at the person in charge for guidance. He (or she) will not be contradicted in public, which can cause confusion, as subordinates may express different views in other situations.

The Concept of Face: Nobody likes to be proven wrong in public, but in Asian cultures, the resulting impact cannot be overstated. To avoid anyone ‘losing face’, people will try to avoid putting their counterpart on the spot or making commitments in public meetings, as either may not be achievable. Therefore, when working with Indian or Asian staff members, discussing project risks or potential delays can become a cultural minefield.

Relationships: In Asian and Indian cultures, it is very important to establish a form of relationship before entering into a business engagement. Without an understanding of this mechanism, you may feel as if they are reluctant to ‘get to the point’. Alternatively, Europeans are perceived as being blunt—and even rude—in their haste to make decisions without properly establishing relations.

Generalization can be incorrect when assessed in detail because everyone is different. However, at the same time, some stigmata do hold value in creating the correct mindset when engaging with people from different cultures. By default, Indians will answer your questions confirmatively and go out of their way to please you. They will be late to inform you of risks or mistakes and will make any effort to solve the issues beforehand. Knowing this can help you to rephrase your questions and invest in the relationship before you become overly explicit in the type of information you need.

Western Europeans (especially the Dutch) are perceived as being very direct and sometimes rude and insensitive. For my Asian friends, it was quite often unclear who was in charge (if anyone) or whether we, as a group, were capable of making decisions. Having a decision enforced by senior management seems impossible, as we only commit to something with which we all agree; consensus seems to be our only *modus operandus*.

Conclusion

It is important to be aware of the communication challenges mentioned above. Understanding these potential issues brings you halfway to addressing and solving them. By being aware, you know what to discuss with your counterpart if communication creates unexpected results. In any case, it is important to invest in personal relationships. Educate yourself on the culture of your colleagues or counterparts. If possible, make sure you have face-to-face contact in the early stages of a project. When reaching out to someone from another culture, make an effort to engage not only on a professional level, but also on a personal level. The challenge is just as big from the other side.

I am convinced it is worth the effort. For me, working across cultures has been a life-enriching experience. As a principle, I believe in the power of diversity in teams and organizations, as participants leverage one another's strengths and compensate for one another's weaknesses. Once we are able to solve the practical issues and bridge the gap between languages and cultures, the potential in cross-cultural collaboration is huge.

About Erik Snijder



After university (Applied Physics at TU Delft), Erik started a career in IT twenty years ago. Following various roles in software development, Erik worked with CSC in the Netherlands, starting as a consultant before becoming involved in outsourcing activities as both a bid manager and solution architect. Later, Erik was responsible for the New Business Development Team of CSC in NL that included the aforementioned roles in addition to transition manager.

In 2006, Erik joined LogicaCMG, heading up the Business Creation team and later as a member of the management team of Outsourcing Services. After just over 4 years in various sales and business development roles, he entered the field of consultancy again when he joined EquaTerra, a company focused on sourcing-related topics.

As EquaTerra was acquired by KPMG in 2011, Erik is now part of the Shared Services and Outsourcing Advisory teams within KPMG. Throughout the past couple of years, he has supported clients in every stage of the sourcing lifecycle from strategy, selection, transition, and transformation projects in both a consultant and/or program management role. He is active mostly in IT (both infrastructure as well as application services) but also in other types of business process outsourcing engagements.

"For me, the interesting thing about outsourcing is the complexity. It is about technology and services, but no assignment can be truly successful without thorough understanding of HR, legal, and financial challenges. Even more important: outsourcing is always about people, which makes it challenging and rewarding at the same time with every new engagement. Understanding people, their motives, and their cultures are essential in addressing the challenges and realizing the potential value of sourcing relationships." ~ Erik Snijder [Back to Top](#)

~~*~*~* **Chapter**
2:

How to Communicate With Your Software Delivery Team

Written by Rajiv Mathew

At the end of the day, software development is all about people. You need effective communication channels that help people work together to achieve the vision of your firm. Throughout the past decade, I have worked in the IT industry and have come across some wonderful tools that aid software teams with their communication challenges. Let us look at a few of them.

Team Briefing

This meeting is often used by a project manager to meet with the entire team and to communicate updates to the project. Every time major changes occur on a project, a team briefing can be called. This needs to be done on a timely basis so that team members have a chance to share feedback on the project status. It also increases coordination among team members as they become more involved with the discussions. Team briefing is a flexible platform for open communication; however, it is best to keep them focused so that they do not become overbearing for the development team members.

Interdisciplinary Meet

Very often, it becomes critical for various teams to meet and discuss project priorities. It may not be possible for the Quality Assurance (QA), User Experience (UX), and development teams to be sitting next to each other during the development phase. In such cases, interdisciplinary meets are held to brainstorm and clarify any questions that team members may have. The advantage to this kind of meeting is that each team understands the other teams' needs and can make changes in their work as needed.

Offsite Meeting

Taking your developers outside of the office for discussions every once in a while often helps to break barriers, make way for fresh ideas, and encourage teamwork by allowing developers to bond well. It is a great boost to morale, as a break from hectic schedules helps the teams to crystallize their thinking. A team lunch is an effective mechanism for loosening up an anxious team. We have even seen instances in which team members attend a cricket match or a rock concert together.

All-Hands Meeting

The all-hands meeting is typically a monthly meeting in which everyone in the office discusses organizational-related issues. All hands meetings are typically rounded off with some snacks and coffee.

One-on-One Meeting

Experts have proven that some people reach a better understanding when you take them aside to talk to them on a one-on-one basis. Therefore, communication with an individual is often done through a one-on-one meeting.

Office Visuals

Nowadays, many software firms have an electronic notice board in the reception area that communicates anything of relevance to anyone who comes into the office. This helps developers to remain current on anything that is happening within the company, informing them of any news of which they should be aware.

We have also placed posters in strategic places within the office such as in the break room and near the water cooler where people are likely to relax while taking a break. Some messages can be communicated at the workstations of team members.

Newsletters

Sending a quarterly newsletter to the entire organization is very useful because it provides employees with an overall perspective of the company and its projects. Inviting the employees to contribute gives them a forum in which they can express their views. Similarly, newsletters regularly sent from the human resources, finance, administration, and marketing teams helps the development team gain a wider perspective of what is happening on the operational side of the company.

Intranet

The intranet is an effective tool for communication in the workplace. One of the companies I previously worked with used the collaboration platform *Jive*, which is an effective tool that enables developers to share best practices on projects. My current organization uses *Yammer*, another great option. The advantage to using these platforms is that they help with knowledge management. Think of it as a Facebook clone within an organizational setup. The intranet is the best place to share technical insights on various projects so that knowledge does not remain tacit in nature.

Company Websites

Even though the corporate website is an external-facing medium, many team members visit the website often for company updates and news. Featuring developers on the company website

adds a sense of belongingness, and thus, increases employee engagement and retention. I have often posted press releases and multimedia on the corporate website and then sent links to the developers so they can read about what is happening in the company. If your onshore client has an extended team offshore, and you are a partner, a micro site can be used to communicate information to your team. I remember using Google Pages as a micro site to create and share information with a partner many years ago.

Storyboards

This is an excellent way to communicate the vision of a project or product to the entire development team. Even complex problems can be broken down into simple sketches so that everyone understands them. White boarding ideas and having developers use the storyboarding methods helps on all kinds of projects. If you have a distributed team, you can do this via Skype with online storyboarding tools such as [StoryboardThat](#).

Process Flow Diagrams

When developers build a platform, explaining the user flow or the process flow is extremely critical for understanding how the system is supposed to operate. Once the process flow is clear to the developers, the development would most likely be easy. This is where we also make use of wireframes to demonstrate the interaction flow using click-through prototypes.

Twitter

We have created hashtags for certain technical topics in the past, and then developers—including those from other companies—have provided multiple insights for solving problems. It is an interesting social experiment to try.

Live Feed

We were working on a big project that involved more than 100 developers, so we used Skype to start a group chat in which team members posted updates for the day, and people observed the updates on a big monitor screen. The entire experience worked much like the big TV screens in stock markets!

Phone

Phone is probably the best method as it gives instant access to the members of a team. You can find answers quickly. Building rapport with people in direct conversation is easy. People and teams who are geographically separated can communicate all at once. Conference calls can connect end-users to the team, thereby, allowing the team members to understand user needs better.

Email

In the office setting, email plays a vital role. While email helps maintain an official log of all conversations regarding a project, email overload can create ambiguity in relation to the project. The major drawback of too many emails is that much of the content becomes lost in email chains. To avoid this, you need to be super disciplined in tracking all the data points through comprehensive documentation. With a distributed team, you should use collaboration tools such as Dropbox or Google Drive to overcome geographical barriers. Be prudent with how email is used and try to have project-related data captured using online project management tools.

Tech Talks

Presentations are the most engaging ways to communicate to smaller groups. They act as a reference for employees who face ambiguity at a later stage in the project. Presentations always reiterate the goal of the project. Tech talks can also help to clarify doubts quickly.

Training

Training is one of the best methods of communicating with a software team. Once you conduct a training session for a particular technology, you are ready to push projects that stand on that technology. Live projects that are preceded by training give team members a sense of responsibility.

Feedback

Feedback is one of the most neglected forms of communication in a software development team. Feedback requires immense focus that can be driven culturally by the HR team. Make feedback an important aspect of your performance-review cycle every 6 months. In addition, the principles of 360-degree feedback are great for communicating within a team. Encourage developers to seek feedback on their work actively. For example, if you are a developer, you can ask for feedback from your fellow developers, quality analysts, business analysts, project managers, and any other important stakeholders in the project. Typically, we track only three criteria in each feedback form:

- What was done well?
- What could have been improved?
- What are your recommendations for the future?

Listening!

Always listen to the conversations within your team. Take note of their suggestions and feedback for better performance. Complimenting members on their input encourages others to

provide their input as well. The management should place special emphasis on listening to their subordinates while they communicate with them.

The 30-Second Rule

Another tactic that often works is to use the “30-second rule” while communicating with development teams. If you can respond to the message within 30 seconds, then you should do it. If you can respond to a developer’s query instantly, he or she will be more efficient. This holds true for any client-related communication. Faster replies make clients feel that you are fully engaged on the project and on top of things at all times.

Meeting Rhythm

People often say that meetings are the biggest time-suckers at the office. That statement cannot be truer in today’s fast-paced world. Hence, it is highly recommended that meetings be planned and kept as short and specific as possible. You should also try your best to have team members avoid distractions (such as phones) so that the meeting is a fruitful one.

Monitoring & Metrics

The effectiveness of communication should be measured occasionally. We have done short internal surveys inquiring about whether people feel as if they are being communicated with and whether they experience any communication barriers within in the workplace. These are all soft issues and should be monitored and measured in sensitive manners. Many large multinational corporations (MNCs) have separate internal communications teams that do this. Some typical questions that we ask in surveys include:

- Are you happy with the information you are receiving about what is happening within the company?
- Do you feel that you are up to date with the news that we publish externally?
- Which medium of communication do you prefer: email, newsletter, SMS, etc.?
- How often do you want to be updated about internal news?

Conclusion

Excellent communication within a development team increases productivity exponentially. Keeping everyone on the same page is critical to project completion and business success. The criticality of communication for project success cannot be neglected by any software delivery company worth its salt! The best approach is to continue learning from your communication

channels and asking for feedback from your team. Communication should be taken seriously, especially since you can pay a huge price for lack of proper communication within a software development team!

About Rajiv Mathew



Rajiv Mathew is the head of marketing at Compassites Software. He is a hands-on technology marketing and communications professional with proven expertise in multiple facets of the marketing spectrum. He loves branding and has been in love with it from his schooldays!

Throughout the years, he has worked on numerous marketing initiatives, including marketing communications, online marketing, recruitment marketing, internal communications, public relations, events, collateral design, sales enablement, social media strategy, and brand management. Prior to joining Compassites, he worked at Thought Works Inc. as a marketing and communications specialist where he was exposed to many aspects of technology marketing. Prior to that, he was a sales consultant at Oracle where he sold database products to the North American market.

[Back to Top](#)

***~*~*~*~* Chapter**

3:

The Tools of Collaboration: Selection and Culture Fit

Author: Davide 'Folletto' Casali

Choosing the perfect tool is often difficult. It can be simultaneously the most important choice and the least important choice you make. It is a paradox because, without a tool, you cannot collaborate (bear in mind that reference to a “tool” here does not necessarily mean a software tool); however, if you are unclear about what you are trying to accomplish, you may find yourself choosing something that is neither software nor specific for collaboration.

The Implied Process

If you search for advice, most of the time, you will find recommendations for specific tools that include lists of the benefits of using one over another along with a promise that they will deliver a certain set of results.

Nothing could be further from the truth. The first thing to understand is that *every tool has been built with certain cultural assumptions and processes in mind*. This is a good thing, but it must be acknowledged. If it is not understood clearly, there can be a big misconception about what that tool can achieve and how. There is no such a thing as a neutral tool, and any tools that claim to be neutral are just trying to widen their target audiences, creating collateral damage in the process.

This underlying process cannot be easily dismissed because each team has a specific set of processes, cultures, and habits—formal and informal—that are adopted and in place. Even new teams adapt to these guidelines, mostly due to the composition of individuals’ assumptions. Three possible scenarios exist:

- The tool’s processes match the team’s processes. This is a perfect fit.
- The tool’s processes vary from the team’s processes, but the team is willing to experiment and adapt.
- The tool’s processes are different from the team’s processes, but the team does not want to change.

Unfortunately, what happens most often is the third scenario. A tool is chosen for either political reasons or personal tastes without gaining a complete understanding of formal and informal processes already in place. This ultimately leads to either abandonment of the tool or to costly refactoring of perfectly fine tools that would have thrived in a different organization.

The expenses associated with refactoring lead to an increase in costs for the overall tool as well as maintenance costs in the future. Moreover, if the tool is changed too much, the overall result will be a decrease in collaboration throughout the entire team.

This is incredibly evident in larger corporations with big implementation budgets that, rather than choose a tool with a new process that could improve their work, they instead choose a tool and manipulate it to work with their processes, often times ruining it.

The second scenario is uncommon because people often perceive a tool as a solution in itself. Since we have thus far acknowledged the implied processes in a tool, now we can see clearly that for the second scenario to work, we would need to gain acceptance from all members of the team as well as their willingness to work through both the trying and adapting phases.

This scenario, for example—one that is pushed by Base camp—is one in which a company not only promotes the tool, but also a specific way of using it.

“We don’t use Basecamp to plan; we use it to communicate. That’s a deep difference in approach.”

~ [Ryan Singer](#), Basecamp

In addition, approaches that begin by focusing on the methodology, such as GTD (Getting Things Done) by David Allen, address this issue before discussing the specifics of which tool to choose.

“Anything that you trust will feed back what you want to be reminded of, and when, will work. It’s not the tool but the contents and how appropriately you engage with it [sic].”

~ [David Allen](#)

They both explicitly declare a specific method for working, so you can more easily understand that the tool is only part of the adoption; the other part is the adaptation to the processes. To address this critical initial moment of choice properly, ask yourself two things:

- What is the underlying process that this tool embodies?
- Which of our existing processes will translate well into the tool?

Formal and Informal Processes

The other trap many people fall into when choosing a tool is exactly that—choosing a tool. Focusing too much on a single, individual tool will cause you to miss the big picture, and in the end, will lead to *adoption by luck*.

“Approximately 70% of projects fail because of lack of user acceptance.”

~ [Forrester Research Report](#)

There are two reasons leading to this perception:

- People view certain tools that have been used clearly and historically as collaboration tools as given and do not include them when making their choices. The email is one such overlooked example.
- People hope that a single tool will solve all their needs.

Of course, this worldwide view fails in reality because, most often, we find a multitude of different tools that work with one another in different ways. The first thing to understand is the difference between the formal processes (model) and the informal processes (reality):

Model: This includes the formal rules of the organization that detail how things should be done: best practices, laws, gatekeepers, hierarchies, approval processes. All of these are formal and slow to change, and to a certain extent, they are well understood (or at least, possible to communicate).

Reality: Most people, however, follow practices that exist beyond the formal processes. Yes, maybe you have to send an email to obtain approval, but one may first ask in person each time to make sure it will go through. Maybe there is a certain way to begin new projects, but someone discovers a faster way that will work as long as the project stays small. In short, people work around processes every day with a network of actions that is unmapped.

This disconnection leads to many problems when choosing tools. If you choose a tool that maps a process that people work around every time, your tool will simply be ignored because that is not how people were doing things.

Many people who love structure and organization react to this reality with more structure and rules. This is clearly the wrong choice because models are slower to adapt to change—and things change all the time. The result is usually bureaucracy and a company that becomes trapped by itself.

However, if you just acknowledge these two distinct levels, you can correctly build upon them and discover how they can guide your decision. In short, **do not be blinded by formal processes; acknowledge informal ones as well when you choose new tools.**

By the way, this does not mean to translate the informal processes into formal ones. Doing so would be an error; it would only make your company more rigid.

The Three Speeds of Effective Collaboration

In my experience with small startup teams and big corporations, I have always found a pattern. I am still unsure as to whether this links to some type of underlying human nature. However, after more than 10 years of doing design work and 5 years of consulting with companies about how to become more social and collaborative, I have been able to outline a model that has worked thus far.

The model is based on the idea that collaboration has three speeds upon which it is based, these speeds must all coexist, and they cannot be conflated into one and remain effective.



Often, you will find that a team needs to discuss certain things quickly through short but intense exchanges. This can be everything from a question that requires a simple answer to complex arguments that clarify something specific. This usually happens via chats.



However, not everything requires this type of quick turnaround. Some discussions may happen in longer timeframes, some maybe delayed, and others may not be deemed as important. Maybe one of the people involved is unavailable. This realm is usually covered by email.



Finally, certain types of content communicate by themselves. The third speed is the slowest: the archive. Oftentimes, documents that are stored for future use do not seem to be parts of the collaboration effort, but it is very important that they are stored somewhere that is easily accessible.

Three elements make up the complete model of the Three Speeds of Collaboration: real time, async, and storage.



RealTime

This is the speed at which *you must be there* to engage in the conversation. This kind of collaboration happens often in one-on-one discussions with many messages exchanged in a short amount of time with quick replies. Sometimes, this can happen with more than 2 people, but it is unlikely to reach a large team. For this speed to work well, it is very important to have a good notification system in place. The characteristics of real-time speed are that it is:

- **Fast:** The interface reacts instantly—within milliseconds.
- **Synchronous:** Everyone must be present at the same time.
- **Interactive:** Interaction is delivered in a short time—ideally, in under a minute.
- **Ephemeral:** What is said is valuable for the people involved but does not persist beyond that.

Examples of real-time tools include:

- Microsoft Skype
- Google Hangouts
- Atlassian HipChat
- Slack
- Microsoft Lync
- Phone and Conference Calls
- Meetings
- Internet Relay Chat (IRC)
- Chat Software

Async

This is the speed at which *you will be there* at some point to reply in the conversation. This form of discussion involves small groups of people. Usually, the groups consist of 1 – 3 participants but not often more than 10 or conversation becomes very difficult. It is frequently represented by content displayed in an activity flow. The characteristics of async speed are that it is:

- **Slower:** The interface may take a few seconds to deliver communications.
- **Asynchronous:** The other person or people are not necessarily present at the same time.

- **Delayed:** There is no expectation of immediate reply. Even if you know the person, you can usually expect a reply within a certain timeframe, usually in the range of hours or days.
- **Persistent:** It is persistent with regard to historical record.
- **Linear:** The structure is often chronological.

Examples of async tools include:

- Email
- Socialcast
- Microsoft Yammer
- Atlassian Jira
- Facebook
- Forum Software

Storage

This is the speed at which *you are not there* anymore in the conversation after you write your message. This is a form of broadcast communication: one person writes, and many people listen, often in a long timeframe. It is often a piece of content that is able to stand on its own, covering a specific topic or subject. The characteristics of storage speed are that it is:

- **Slow:** Publishing content may take several days and include an editorial review.
- **Completely Asynchronous:** The author may not even work with the team anymore.
- **Autonomous:** No answer is required. The content is self-contained and communicates by itself.
- **Timeless:** It is very persistent. Documents that are stored may remain for several years.
- **Organized:** A multilevel structure exists with hierarchies, categories, tags, and metadata.

Examples of storage tools include:

- Atlassian Confluence

- Microsoft Sharepoint
- GitHub
- Box
- DropBox
- Google Drive
- Database and Storage Software

In Practice

I work with Automattic—the maker of WordPress.com: one of the world’s biggest, fully distributed companies. We employ 240 people spread throughout 30 countries (as of 2014), making even more evident the needs and constraints of fully distributed teams.

WordPress is clearly at the core of our technological infrastructure, as you will see below in the tools we utilize:

- Real Time: Slack
- Async: WordPress with P2 Theme
- Storage: WordPress

It is even more interesting to note that, until early 2014, Automattic used an even more basic tool in place of Slack: IRC. The reason was mostly historical; it was chosen more than 10 years ago, and thus, was well integrated.

Given the size of the company, the simplicity of this approach often surprises people. However, it is also clear evidence of how much more important the internal collaboration practices are and how simple tools can go a long way.

The company is mostly flat and structured in independent groups with each group having a Slack channel and a P2. Therefore, if you need something specific, you must write a post or have a chat in the right place to receive the answer you need. There are no hierarchies to overcome, bosses to consult, or procedures to follow.

In addition, everything and any level of detail is visible to everyone—from the code committed to the discussions surrounding company strategy. This means that the need for access control, blocks, limitations, and so on is almost nonexistent, removing a huge burden from both our processes and tools.

This is especially interesting because a previous company I worked with was attempting to change to a more collaborative environment; however, the pool of enterprise tools we had at our disposal were “obscure by default.” Therefore, every time someone needed to setup a new group or space, the default setting was private and restricted, so you had to manually add each person. Having such a default setup creates silos and harms collaboration. On the other hand, you will also notice that committing to **transparency is a cultural choice**; it is a management choice that affects multiple levels. *The tool is just a consequence.*

If you want to read more about Automattic, you should read *The Year Without Pants* by Scott Berkun, which describes the day-to-day details of working with us. Let us look at a few more examples.

Example 1: Google-Centric

Google covers all three speeds with the following tools:

- Real Time: Hangouts
- Async: Gmail
- Storage: Drive

Google is doing a great job with Google Apps. They cover every side very well with a good degree of integration. Having it available as both private and business is also a great addition.

Example 2: Microsoft-Centric

A solution entirely based on utilizing Microsoft tools could follow this pattern:

- Real Time: Lync
- Async: Socialcast
- Storage: SharePoint

This type of approach is often found in companies that are entirely Microsoft-centric and are most commonly larger corporations. They clearly have an advantage in terms of systems integration.

Example 3: Atlassian-Centric

Atlassian is an excellent company that is very focused on collaboration, specifically remote team collaboration. Their suite is very well designed, and you can find a good solution using the following set:

- Real Time: HipChat
- Async: Confluence (the activity-stream view)
- Storage: Confluence

Example 4: WordPress-Based

The important thing is to cover all the aspects and have them interact well with each other. You can cover a lot with open-source tools. For example, WordPress.org, the open-source project, uses:

- Real Time: IRC
- Async: WordPress With P2 Theme
- Storage: WordPress Used as a Wiki

While WordPress does not have a real-time option, it is complemented by IRC, which is used due to its easy integration capabilities and simple, unrestricted access.

I have created symbolic examples above that link to specific sets of tools, but the most important aspect of the three-speed approach is that you can **mix different approaches** well and without any problem. Often, by utilizing **basic integrations** of these tools, you can resolve most problems. For example, mixed approaches could include:

- IRC + Confluence + SharePoint
- Slack + Gmail + DropBox
- HipChat + WordPress P2 + Confluence

Most importantly, these solutions work well with how your team already works with regard to formal and informal processes.

Do not overthink it! Put more effort into the testing, adoption, practice, and rollout aspects of the collaboration platforms than you do in selecting and configuring the tool. Do not focus on having it perfect from day one. Instead, plan to experiment with it and take incremental steps to adoption.

About Davide Casali



Davide Casali is an experienced director and startup advisor with more than 11 years of experience and a hybrid background in design, psychology, and technology. He led crossdisciplinary teams of designers and developers, created and advised successfully funded startups, and has coached and trained employees of both global and local organizations (DHL, EY, Atos). Currently, he works for Automattic, the makers of [WordPress.com](https://www.wordpress.com).

Believing as Leonardo da Vinci reportedly once said, “Simplicity is the ultimate sophistication,” he is specialized in social experience design, behavior change, and motivation applied to interactive and social systems. He has a specific focus in personal growth and leadership coaching to drive successful teams.

During his years of professional experience, he led the design of consumer and enterprise services for more than 2 million users worldwide in different markets, industries, and countries. He worked with Nokia, MasterCard, Red Bull, Ferrari, Bank of England, Intesa Sanpaolo, Banca Sella, Unicredit, Vodafone, Telecom Italia, Benetton, and more.

His designs won the Aretê Award, eCommerce Award for Innovation in Delivery, and the Samsung GOLD Prize, and he was listed in the ADI Design Index 2010, the selection for the Compasso D’Oro.

He is an internationally recognized speaker and lecturer (IxDA, LIFT, Creative Mornings, General Assembly, Social Business Summit, Social Media Week, and Frontiers of Interaction). [Back to Top](#)

~~*~*~* **Chapter**

4:

Process, Rhythm, Metrics, and Feedback

Author: Hugo Messer

Communication is often seen as the biggest influence in a successful distributed collaboration. When I speak with people who have had challenges working with a remote team, they always point to communication being the main obstacle. What is communication though? How does communicating with a remote team differ from communicating with a local team?

In this article, I analyze the meaning of communication in remote collaboration. I also share some experience that will demonstrate how to structure your communications so that your offshoring endeavor delivers value.

What Is Communication?

I think everything people do in business is some form of communication. We continually communicate in different forms (e.g., writing, speaking), through different media (e.g., email, Skype, SMS, WhatsApp Messenger), and on different levels (e.g., chitchat, operational, reflective, strategic). Communication is influenced by many subtle factors such as the participants' cultures and accents as well as whether they are introverted or extroverted. Because communication covers such a wide spectrum with so many variables, focusing on communication per se is not the solution to making remote collaboration work. We need to look at other aspects.

How Does Communicating with a Remote Team Differ from Communicating with a Local Team?

In many ways, it is not different at all, and in other ways, it is very different. On an individual level, members of remote teams lack face time in the office. Face time with your colleagues allows you the opportunity to bond and chitchat about things that matter to you. Because remote team members lack face time, we do not have the same chance to build strong bonds as those who see each other regularly, resulting in a weaker understanding of one another's motives.

Even if we use video conferencing to 'see' each other, we miss the emotional cues. Last week, one of our partners demonstrated his product to our Polish and Indian product owners. After the Skype call, he told me that he really had no idea what Krishna (our Indian colleague) thought or how he felt about the product because he could not see any emotion on his face.

At the project level, we miss chatting at the coffee machine about the task we are working on together. We miss the brainstorming sessions about the product we are creating and the clients we

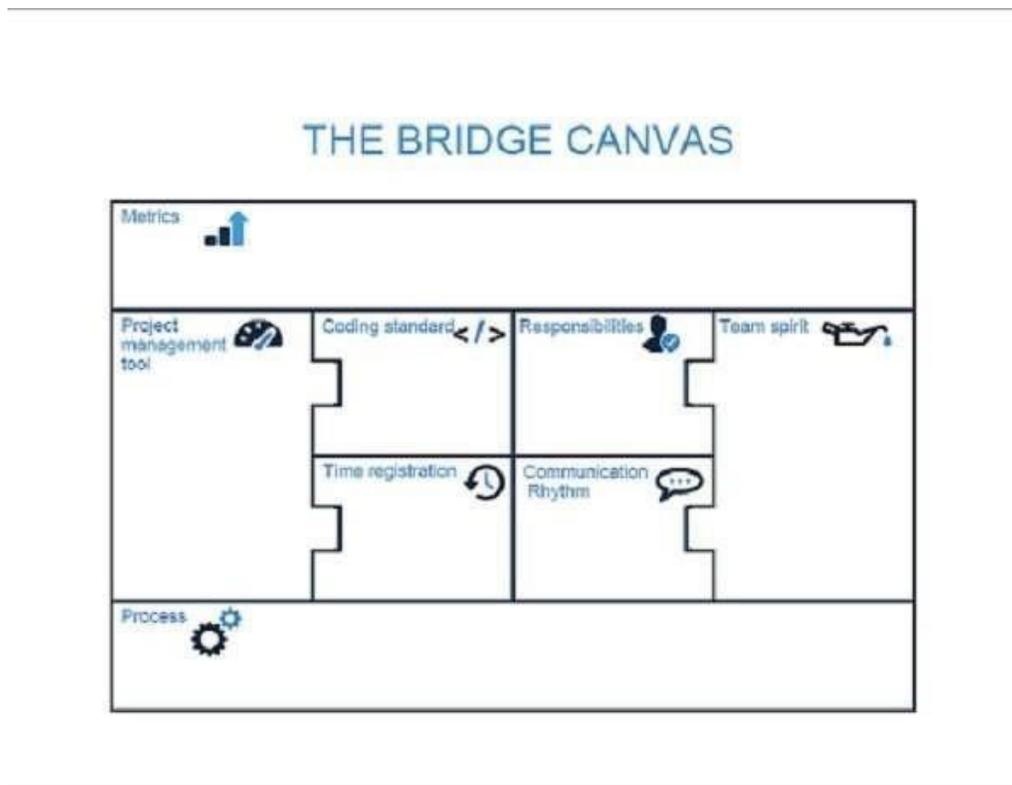
are serving. This lack of face-to-face communication has a strong impact on the level of understanding that remote team members have about the product, its purpose, and the clients who use it.

At an organizational level, we miss the inspiration gained from talking with the company CEO over a beer at Friday night gatherings. We miss the way the core values of the company influence the behavior of the people working within the onshore office.

To bridge these divides, we need to give more thought to remote communication. When we communicate with people in our office who are from the same culture, we do not experience conversational barriers. However, when we speak with remote colleagues who are from another country, we have to think more deeply about the forms, media, and levels of communication we choose. Let us look more closely at how we can go about this.

The Bridge Canvas

As part of our Bridge Method©, we have developed a Canvas that structures how people think about communication. Our Canvas covers every role in remote collaboration; however, I will focus more on areas that I have found to have the greatest impact on communication: process, communication rhythm, and metrics.



Process

Process has been elaborately described in our book [How to Organize Offshore and Nearshore Collaboration](#). What is important to understand is the impact that your software development process—and thus, the engagement model—has on communication.

Agile Versus Waterfall Methodologies

In an offshoring context, waterfall encourages the idea that the onshore team does the thinking, and the offshore team does the implementing. Onshore members speak to customers and develop clear requirements. Offshore members then begin programming those requirements. I am not a proponent of waterfall, so I may state things too negatively here for people who have had good experiences! However, in my opinion, waterfall can cause some challenges in communication:

- Offshore team members miss communicating with clients.
- The requirements are communicated in writing.
- Offshore members are less engaged in developing the solution required.
- Many question and answer rounds are needed to clarify the requirements.
- Much less (if any) face-to-face communication exists between onshore and offshore members.

The agile strategy has several principles, one relating strongly to communication: “Individuals and interactions over processes and tools.” I believe this mindset results in agile being more suitable for remote communication than waterfall is. Scrum has a set of planned meetings that allows the teams to interact continuously. Customers are also expected to be involved in the meetings. This collaboration provides solutions to the challenges described above in the waterfall model:

- The entire team interacts with customers.
- Requirements are fleshed out in team conversations so nobody misses important information.
- Questions can be discussed during scheduled meetings, and ideally, the product owner is always ‘on hold’ to help answer questions, even if he is onshore.
- The whole team has daily face time.

Fixed Price Versus Time and Material

Working on a fixed price creates the assumption that the onshore customer can sit back and wait for the delivery from the offshore team. To reach a fixed price, requirements need to be 100% clear—and they never are. This leads to the notion that the requirements process is complete and all that is left is the execution. Of course, customers tend to believe the implementation is done offshore, so communication during the project is not needed. Therefore, when questions arise, the onshore people are not always available, and when offshore team members are able to reach them, they feel as if they are intruding on their busy schedules. Offshore team members also feel that they should solve things themselves because the requirements were already specified and agreed upon in advance. However, when the offshore participants attempt to resolve issues themselves, they do not always deliver the results the customer had anticipated. The relationship becomes a traditional client-supplier relationship, which results in communication that is based in large part on ‘sides’ and ‘us versus them’.

Some companies master the management of fixed price projects even with part of their team offshore. In most cases, this model includes onshore project management. A customer outsources a project to a local company and obtains local project management (and/or analysts, account managers, scrum masters, etc.). The project management, in turn, communicates with the offshore team so the customer is ‘protected’ from remote communication. Of course, this has a large impact on the communication stream: there are always more intermediaries, resulting in the loss of information, and the weakest link can derail any project. On the other hand, if experienced project managers are involved, they may be able to overcome many remote communication hurdles.

When time and material is the model, every hour worked is an hour paid. By default, this causes buyers to be more alert because wasting time is costly. One may assume that the offshore team members (being suppliers) can relax because their hours are paid anyway. Of course, suppliers want to create happy customers, so the aim should always be to develop great software quickly, efficiently, and within budget. This type of engagement tends to be more effective, and in my experience, delivers better results. The relation tends to become a partnership, where the partners communicate continuously to improve the collaboration process.

Metrics

How do we measure performance? On each level (operation, process, and strategy), we need to identify performance indicators. The best place to start is with arranging meetings at each organizational level to discuss metrics that indicate progress. I always prefer to use simple examples; however, in a big, multinational collaboration, more complex key performance indicators (KPIs) could be invented as described in Erwin de Bont’s article in [How to Manage Offshore and Nearshore Collaboration](#).

At the operational level, a useful indicator is team velocity (or the growth of the velocity over time). This indicates not only how effective each team is over time, but also how each team compares with other teams. At the process level, the value of the collaboration can be measured on a scale of 0 – 10. Each ‘side’ reports a value each week that can be discussed during weekly meetings. On a strategic level, time to market and product line growth are useful indicators of a fruitful collaboration.

Once metrics have been defined, it is important to outline how they will be recorded and communicated to everybody involved as well as how they will be discussed and acted upon.

Rhythm

Depending on the size of the company, people involved in remote collaboration communicate on at least three levels: strategic (CEO), process (project or process managers), and operational (scrum master and team).

Agile has a set of meetings scheduled on the operational level. The product owner, scrum master, and team members are involved in the sprint planning, demo, and retrospective meetings. The scrum master and team members manage the daily standup (involving the product owner when needed). I have learned that this meeting rhythm is absolutely crucial to and beneficial for remote collaboration. When teams do not use this method—especially the daily standup—things quickly slide down a slippery slope.

In most cases, the product owner is local, and the scrum master and team are remote. By following the scrum meeting rhythm, both shores automatically communicate daily. This increased communication creates the possibility to continuously adapt, remove hurdles, and create bonds between all team members.

At [Bridge](#), We have learned that the role of process manager is crucial. In our Bridge method, we have an onshore account manager and an offshore process manager. Ideally, the customer also assigns a process manager (this can be the product owner or the manager who is responsible for the success of the collaboration). These three managers attend weekly Skype calls that last about 15 – 20 minutes in which they discuss alignment: *Is the team on track? Are they communicating effectively? Are they still following the agreements that were initially made? What is going well, and what could be improved?* This information is then stored (e.g., in Google docs or Atlassian Confluence) and shared with the entire team, so each week the collaboration improves a little more.

The top level communicates every 4 – 8 weeks. In the initial stages of collaboration, it is recommended that they meet at least once every 4 weeks. They discuss the progress of the strategic objectives of the partnership: *Are the KPIs still on track? Are we providing value to one another? Do we have synergy? Do we work well together? What are the plans for future collaboration?*

For example, a big Dutch/Swedish telecom company assigns a “change manager.” They always have one change manager from their company (local) and one from the partner’s side (remote). These change managers speak weekly. The product owners report to the change managers so they are always aware of the project dynamics. Every 3 weeks, the operational managers meet to communicate progress and the achievement of operational objectives. Every 6 weeks, a meeting is held at the strategic level. On every level, this results in continuous communication and adaptation and fosters deeper ties that result in a long-lasting partnership that increases in value as time passes.

Conclusion

With some practice, communicating with a remote team becomes similar to communicating with a local team. To achieve this balance, it is important to start with discussing how the teams will communicate and then formalizing this in a communication plan. The Bridge Canvas can help with this. With everybody involved, you discuss the impact that the software development process and the engagement model have on communication. You then devise some KPIs to measure the performance at different levels in the collaboration. These measures are then discussed regularly during pre-planned meetings. The combination of thinking, measuring, and meeting rhythm results in smooth communication.

About Hugo Messer



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

To know more about Hugo, check out his [website](#). You can also read the [blog](#) or watch videos at [YouTube](#).

About Ekipa Academy

Ekipa Academy is the first online platform educating people on distributed teamwork. Our training programs teach you how to implement agile and scrum in your distributed team, how to manage intercultural teamwork, what tools to use. Courses are offered online, as live training and in-company. We add coaching to help you implement the improvements. Our trainers are all passionate about distributed work. Their backgrounds vary from agile, software development to cultural differences and virtual teamwork.

Website: <http://academy.ekipa.co>

[Back to Top](#)

~~*~*~*

Chapter 5:

Remote Teams and Communication

Author: Arjan Franzen

The first time I heard a colleague from the sales department talk about the future of outsourced software development was in 2004. I was quite surprised and a bit cross with him. I recall that he tried to give me the impression that I had better pursue a career other than software development since, “Bangalore is the place to be! So much of the current software development is going to be done there!” I was a Software Engineer at the time and thought my career would be severely blocked by this ‘outsourced work’. Throughout the years that followed, I became involved with a number of projects in a number of different roles. These projects were sourced either partly or wholly in countries other than the Netherlands (NL). Since that time, I have become familiar with the possibilities and limitations of outsourcing. I have learned not to fear, but rather embrace and enjoy, working with remote teams. A combination of local and remote teams can genuinely improve the quality and innovation of the software stack the organization is tasked to support. Working together can easily be beneficial to everyone involved. These are my experiences regarding communicating with remote teams.

Over the Wall

When I started becoming involved with outsourced software development, I worked for a medium-sized IT solutions provider. The teams we managed were located in India. Before we started, we were given a cultural sensitivity/how-to-interact-with-remote-teams course, and then we began working with the teams. I wondered what made this ‘over-the-wall’ way of working so dominant. I saw my colleagues operating in a similar way with their projects. This project management method can be summarized as a very simple customer-supplier type relation. In this specific case, it was also an internal relationship since both the India team and the NL team were part of the same company. The NL team can be viewed as the customer and the India team as the supplier. Essentially, the NL team was the intermediary between the ‘real’ customer (paying end customer) and the software developers in India. The solution provider built the software according to end-user demand. In my experience, this model is the simplest way to work together. However, this model is based on a number of incorrect expectations:

- In a relationship built on little trust, end-customers are expected to purchase a commodity in the form of software features.
- The specified requirements are sufficient to deliver a valuable product to users.
- End-customers expect the outsourced teams to ‘hit the ground running’.

These incorrect expectations need to be managed from the start. It turns out that the background for this ‘over-the-wall’ collaboration was by company design. The classic method of working was to create batches of work based on technical skills. This means that requirements are established first, then they gain approvals, and then they progress to the design phases, etc. Since this way of working was the standard, the end-customer intermediary team (our NL team) was essentially in charge of finding a solution for ‘the wall’. It may have been my first project with outsourced development; however, it certainly was not the project manager’s first. The NL team quickly set about managing the expectations and tearing down the wall.

Improving

We quickly learned that no amount of detailed specification (design or requirements) would render the exact software system you envisioned when writing the specifications. This means that there is a constant need for communicating with end customers (users, influencers, and purchasers). We needed to demonstrate a usable product based on the initial requirements. We had engaged the end-customers with per-feature communications about details, priorities, and designs. This approach solved 2 of the 3 project risks. There was, however, a challenge to this approach: the first workshop. This was the moment when the end-customer would realize how far along we actually were as well as how well we had understood their problem in the beginning. These initial workshops would take the longest with new customers, and we faced the risk of negatively influencing the end-customer relationship if we were not well prepared.

We learned to prepare these meetings in such a way that allowed us to uncover details that were not addressed in the initial specification. We included the entire team in the workshops, meaning we co-presented features and designs (if technical representatives were present) to the end-customer using a video conference solution to include the India team. We had taken steps not to separate development from design. This allowed feature design and development in lockstep. Achieving this was possible because we had established open communication and cultivated trust between all locations involved. Changing the way we communicated, from being focused on skills and phases to concentrating on value and features, ensured that we quickly gained a good understanding of the end-customer's problem.

Inclusion

Sometime later (working for a software product company), I was able to take the things I had learned from previous experiences a step further. This time, there were differences in how the contract was set up. The country in which the outsourced teams were located was different (Ukraine) and so were the organizational arrangements. The remote teams were part of a different company—an actual supplier. We had made sure that we did not need an intermediary team in the Netherlands and decided to make the remote teams an integral part of the software development organization. The goal was to maximize the flexibility of the resourcing. When planning software releases, we must ensure that we are constrained as little as possible with regard to what team develops what release. These types of routing constraints can make a release plan an unsolvable puzzle.

Once we were onsite, we found motivated team members with many questions on their minds. We started immediately. We approached each team member as if he or she were a new hire in our NL office. This helped us tremendously because trust is often lacking in new teams. We set up an introductory training program that detailed the technical environment and the market for which we were producing software. We specifically went for both technical knowledge and environment because this would help establish a sense of purpose for the remote teams. We spent about 70% of the time explaining the work processes, tools, and architecture. The remaining 30% of our time was spent discussing the business: who are the customers? What are their domains? At one point, I had made the mistake of including a slide twice in my presentation. It happened about an hour after I had shown the first slide to the team, explaining the architecture that was on it. I saw the mistake immediately but thought I would see how it played out. Would they detect the double slide, or would they quietly sit and politely nod 'yes' the entire time. I continued to present the slide for the second time. After 10 seconds, I was interrupted by the lead developer in the team: "I'm sorry Arjan; I believe you already went over this topic. Is that correct?" My trick worked! I was happy to discover that they were paying attention, understood what I had to say—and most importantly—they did not feel as if they had to suppress feedback regarding the quality of information provided (double).

After the first day of the course, we had dinner together and visited a local pub. This combination of training and discussions helped the Ukrainian team understand our situation. On

our side, we began to trust the members of the team because we saw firsthand how they approached the difficulties involved with becoming familiar with a completely new environment. Discovering shared values between team members and customers helps greatly to establish basic trust between teams. This trust is the basis of a good working relationship.

Please note, “Your mileage may vary” regarding this procedure, depending on where you outsource your work. Even after enjoying a night of dinner, discussions, and getting to know team members better, the next morning, you may still find that cultural differences cause communications to remain unchanged—stifled.

Back Home

After we kicked off the project and were back in our office in the Netherlands, we made sure we stayed true to the rules of our agile software development process. This means attending daily standups Monday through Friday with the product owner and technical liaison in NL and the scrum master, developers, and testers in Ukraine. To improve connection, we installed webcams in both conference rooms and shared the desktop of the machine that displayed the scrum board during the standup, meaning that this brief, daily, 15-minute meeting referenced a single screen. The entire team and all the stakeholders shared the same view. A challenge with standups is to keep them within the 15-minute timeframes. Standups tend to generate discussion on technical topics. A remark often heard during the standup is to “take the discussion offline,” after which a conference call is set up to discuss the matter in more detail. The result is that meetings and conferences are kept to a minimum. The standup ends each day with the scrum master asking whether any of the tasks on the board are being impeded. If so, it is up to the scrum master to schedule follow-up meetings involving experts.

In addition to the daily status meetings, there are skype chat groups that are always open. This allows all people who are involved to ask questions. The culture of asking questions is rewarded in the improvement meetings (retrospective) with honorable mentions. These meetings are perhaps one of the most important aspects for long-term collaboration. The retrospectives allow members to course-correct the collaboration and are great ways to build trust and gain feedback regarding processes. A critical step is to make sure the retrospective is conducted without placing any blame. The structure of a retrospective is that each team member places notes on a 2-column whiteboard, explaining what went well and what can be improved. The scrum master leads the session, goes over what went well, and lists all the feedback in the document. After completing the session, it is critical that members follow up on the improvements and provide feedback in the next retrospective. Each sprint has 1 retrospective. The idea behind this is that the feedback is bottom-up rather than top-down. The team is encouraged to provide some sort of feedback to improve the process as well as the product. We have found that sharing processes between locations, especially in the case of outsourced software development, is critical to success. The ‘over the wall’ type collaboration must be broken down into smaller workable processes. In the end, one of the few exceptions in terms of processes for the remote teams was that our Ukrainian teams were unable to participate in the

‘improvement days’. Improvement days are a way to innovate and motivate teams. We found it difficult to implement this process abroad.

While setting up the close collaboration with our outsourced team, we found it beneficial to invest in global infrastructure from the start. This means we must set up an environment in which cross-functional teams are able to use the exact same central services for developers, testers, product managers, and project managers. In organizations of all sizes, this means having a good level of support from internal IT organizations and the possibility to use cloud-based services that are accessible from both the main and remote sites. What we experienced is that, by allowing local and remote teams to use the same services, we increase both trust and flexibility in sourcing the releases.

In the end, we discovered that the total package of communication, trust, and processes found in this model were very beneficial to our organization since flexible scheduling is something every company needs. We were able to achieve ‘hit the ground running’ type projects because of the way we structured the knowledge and communication.

Conclusion

While much outsources work is accomplished with economies of scale in mind (large batches of similarly skilled work), we found that focusing on flexibility and flow made a much more powerful setup in outsourced development. Maintaining open communication with teams and end-customers regarding priorities, problem domains, and requirements is critical. We found that working in this manner increased the total throughput (productivity) and positively influenced the provided value to our customers.

About Arjan Franzen



Arjan Franzen has 14 years of experience in the software development industry in various fields within the engineering sector of software development. His industry experience is in telecoms and the financial industry. Arjan is development lead at NICE Systems in the Netherlands. As such, he is advocating and implementing the principles of continuous delivery and lean software development at NICE Systems. NICE Systems is a worldwide leader in intent-based solutions for contact centers and the financial industry. [Back to Top](#)

~~*~*~* Chapter 6:

Oral Versus Written Communication

Written by: Hugo Messer

This short, intermediate article will explain the different impacts of oral and written communication on project success. To become an effective remote collaborator, it is important that you carefully construct your messages and their structures. People have a natural inclination to sit down and ‘just start communicating’.

In a more traditional, waterfall-like collaboration, you likely use one of two scenarios to explain your software requirements:

- You call your team in India to explain it verbally (“It takes too much time to write everything down”).
- You send a short description by email or Skype chat to your offshore team (“It costs me too much time to make a formal functional design”).

Let us consider both methods of communication and their impacts.

In scenario A; you call your remote team leader to discuss a specific function or aspect of your project, and she listens carefully. Some people may make notes while others ask many questions, but all aim to understand what you require. You spend 20 minutes talking with her, and then you both confirm that you understand things clearly. The programmer starts coding. Unfortunately, she stumbles upon a logical question and starts wondering what the best solution might be. (Either she did not understand something you said during your conversation, or you thought you had explained everything but left something out, or she thinks of a better solution than what you had envisioned.) She now has no reference to anything you discussed in your verbal communication. Her only option is to call you again to clarify. You may spend another 20 minutes discussing the situation and then still have the same scenario happen again the next day (for either the same circumstance or another part of the project).

The issue here is knowledge. You have certain knowledge about what you require—knowledge your offshore colleagues do not have—and you try to communicate this verbally. Your mind works differently when speaking than it does when writing something down. When you write, you are forced to think deliberately about every detail of your requirement. You think more deeply about what you really need to say, and then you only write what is necessary. It also allows you more time to consider how your offshore team will perceive your message (the better you know each person, the better you will know what information you need to provide). Another advantage of writing is that you are not the only person who is able to explain the requirements. The offshore programmer can ask his team leader or a colleague to read your message, and together, they can arrive at a solution. This helps to avoid the extra rounds of communication.

In scenario B, you send your programmer or team leader an email or Skype chat message to explain your requirements. After receiving your message, if he has questions, he will most likely reply to your message. You reply with an answer, and you now have a thread containing questions and answers. This process may repeat itself for each requirement you have, and most likely, you both send a separate message to address each new question. Two things happen here:

- The communication is terribly unorganized due to having multiple threads, the messages are lost, or the messages become confusing.
- The only method for fully understanding the requirement is written text.

Alternatively, if you explain the requirement in writing first and then follow up your message with a phone call to discuss it verbally, the programmer's mind has a chance to process the information more fully. She has the opportunity not only to think about your requirement, but also to consider any doubts, questions, or improvement ideas she may have. Calling her after you send a written communication will most likely result in a discussion that more effectively clarifies the information. If the programmer thinks of additional questions after your phone call, she can either call again or ask her colleague. To make both of your lives easier, it would be best if the written documents were not emails or chat messages, but rather through an online system with structure and overview.

In my experience, the three golden rules for offshore communication in a waterfall model are:

- Invest time in writing your requirements, and think carefully through each detail; ideally, you should agree on a standard method for documenting requirements.
- Use voice (Skype/phone) to clarify your requirements. You should also ensure that the programmer writes down everything you discuss so you can later verify what was agreed upon as well as that she understood the requirements clearly.
- Save all written communication in an online project management system.

A variation of rule number 1 could also be to communicate a requirement verbally to an offshore programmer and ask her to create a written transcript of your discussion for you to

verify. This may save you the time you would spend writing things down but will most likely only work if you know the offshore programmer well.

In an agile collaboration, the requirements are mostly fleshed out in Skype meetings. Business users or product owners formulate short “user stories” of 1-2 sentences. In the sprint grooming or planning meeting, the product owner, scrum master, and team members discuss in functional and technical terms what each user story entails. The stories are then split up into smaller stories or tasks so that each user story remains within 4-8 hours of work.

Now the question is how to record the details that were discussed in the planning session. If we assume that everyone understood the requirements because they all participated, we run the risk of being wrong. It is advisable to ask the scrum master or a team member to record the details in writing in an agile tool such as Jira, TFS, or Pivotal Tracker. In a team that has no previous experience working together, it is beneficial for the product owner to confirm that the written details match his own perceptions. Over time, the need to record and verify the details will decline as the team establishes a common understanding. We learn what to expect from our colleagues.

In conclusion, it is important to find the right balance between oral and written communication. Depending on the collaboration process (agile versus waterfall) and the maturity of the team, this balance will develop over time.

[Back to Top](#)

~~*~*~*

Chapter 7:

10 Tips for Communicating Successfully

Written by Jutta Eckstein

“Creating an effective global team from multiple sites involves several key success factors: building trust, encouraging open communication, building personal relationships, and bridging cultural differences.”

~ Erran Carmel (American University, Washington D.C.)

Erran Carmel, who studied about 100 global software companies, summarizes not only the key success factors for global teams, but also the ingredients for a successful communication. Communication can only be fruitful if there is trust between the people involved, they have a sense of connection, and they are aware of their cultural differences and similarities. If you are

using an agile approach, you will pay a lot of attention to communication anyway because it is a requirement of the process. For example, the Agile Manifesto states, “The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.” Erran Carmel explains that successful communication is a key success factor for effective global teams, which makes agile—with its focus on communication—a natural fit to global development.

Non-agile global teams often have as few communication points as possible because communication is regarded as a potential source of failure. Sometimes, projects establish only a single point of communication. The belief is that the fewer points you have, the fewer misunderstandings will occur. Moreover, non-agile processes assume that most of the communication between different sites will take place via written documents. Yet, communicating via documents implies a huge risk: the assumption that documents are free from culture. However, the opposite is true—we read and write everything from our cultural background. Even if you assume that you have written, for example, the requirements in an unambiguous way, your culture still guided you. Consequently, it is unsurprising that people from different cultural backgrounds have—while sometimes only slightly—different understandings of what is intended.

In this section, I want to provide a collection of my top ten tips for establishing and preserving successful communications for global teams. These tips are based on my experiences with working in different global settings and are, in this way, an excerpt of my book, *Agile Software Development with Distributed Teams*. Naturally, in this book, you can find more details on the subject matter. Here are my top ten tips for communicating successfully:

1. **Increase Communication:** When working with global teams, communicate more and not less than when working with collocated teams. You need to ensure as many communication opportunities as possible. Much of the communication between members of collocated teams is free when meeting at the water cooler, over coffee, or in the hallway. They have the possibility to overhear conversations and jump in when they are related to the topics they are working on. For a global team, none of these situations occurs. Thus, you need to create communication opportunities—the more the better. The opportunities can be defined by regular virtual meetings, virtual tools (such as instant messaging), frequent face-to-face visits, or recurrent phone calls.
2. **Simplify Communication by Ensuring Trusted Relationships with Global Team Members:** It is very difficult to establish such a relationship virtually. Therefore, I highly recommend ensuring some face-to-face time—especially in the beginning of the joint collaboration—to create such a bond. Once people have met, it is much easier for them to trust one another. Knowing people by seeing and speaking with them personally is different from knowing people only virtually. There seems to be some truth in the saying, “trust needs touch.” Virtual communication that is supported by diverse tools is a great means for working virtually together most of the time; however, to build and maintain a trusting

relationship, you have to ensure face-to-face time occasionally. The guideline is that the closer the distance, the more frequent the face-to-face time should be, and the further the distance, the longer the face-to-face time should be. For example, if the distance between the sites requires four hours or less of traveling (e.g., if the development effort is spread out between Central or Northern and Eastern Europe), it is possible to travel back and forth every other week. If the travel time is longer, then the traveling takes place more seldom, but the trips are longer. For instance, if you are spread out between the USA and Asia, for each trip, you should stay for at least one week. Obviously, you will not travel there every other week, but maybe every two to three months. Trying to cut communication costs to save money will cost you more in the end.

3. **Allow Social Time During Regular Virtual Meetings:** If you are working in an agile way, you will conduct a daily synchronization (also known as daily Scrum or daily standup) even with a virtual team. Normally, the team members use the time in this daily synchronization to report (to each other) on what they have completed since the last synchronization (the day before), what they intend to complete before the next synchronization (the next day), and whether anything is hindering their progression. For a virtual team, it is essential that you allow additional social time in which team members share whatever is important to them at the time. It could be something about family life, sporting events, or even the weather. Becoming familiar with your peers' lives helps create deeper mutual understandings and closer bonds. Certainly, it also makes sense to allow social time during other regular meetings (if you are not working in an agile way).

4. **Keep the 'Trust Threshold' in Mind:** The quality of relationships differs. Even in our private lives. Most people have friends who they still consider very close friends even if they only see each other every two to five years. While with others (those who we sometimes think we are close friends with), not talking for four months results in superficial conversations because the relationships lack serious substance. For work relationships, the risk is even higher. After even a relatively short time—in my experience, after eight to twelve weeks—the trust level is reduced in such a way that it reaches a critical point: the trust threshold. Once you have reached the trust threshold, it is very likely for statements in emails (or any other kind of virtual communication) to be misinterpreted in bad ways, causing people to refrain from collaborating and start pointing fingers at each other. After something like this has happened, reestablishing trust is much more difficult than not allowing it to reach this point to begin with. Ensure that there is some communication among members of a virtual team within the eight to twelve week period to prevent reaching the trust threshold. Not everyone needs to see everyone, but some people need to make sure that there is some sort of continuous direct connection.

5. **Be Sensitive With Your Words:** The words we use sometimes refer to a specific culture, time zone, or geographical region. For example, if team members at the headquarters speak about the nightly build, they may not consider that it is daytime in another location. Even referring to a location as a ‘remote site’ is difficult, because in a global setting, all sites are remote in some way. Be aware of the words you use and try to come up with words that work at all sites. This will allow you to create a project language that will further assist you in establishing a joint culture.
6. **Be Aware That Humor Seldom Translates:** Although good humor pulls a team together, you must be very careful using humor on global teams in which people come from different backgrounds. Do not underestimate the harm that can be done to teams when people do not have the same sense of humor. Even when people share a native language, the sense of humor can still differ. For example, there is a big difference between American and British humor.
7. **Empower Ambassadors, Expatriates, and Frequent Travelers to Create Joint Understandings:** Misunderstandings are often based on ignorance or unawareness. Being familiar with all the involved sites helps to overcome misconceptions. It is a good idea to have an ambassador who will represent his or her home location from a distant site. The ambassador works as a regular team member but ensures that neither site (his home or his visited) is ignored; that problems are stated, understood, and addressed; and that general misunderstandings are clarified. Expatriates—both while they are working at other locations and once they have returned to their home locations—help create mutual understandings between different sites just by knowing both cultures better. Individuals who travel occasionally provide a similar perspective. All of these individual connections help foster personal relationships and support the lean principle of *genchi genbutsu*— meaning, “Go and see for yourself” (e.g., before making any assumptions).
8. **Change the Meeting Location:** Whenever there is a face-to-face meeting for people who reside in different locations, ensure that the meeting takes place at changing locations. For example, it is a good idea to rotate the meeting locations over all sites involved so that every location acts as a host occasionally. Changing the meeting location gives more people opportunities to get to know different members of the team better—both the ones who are traveling and the people who are working at the host’s location. Moreover, it helps to develop deeper understandings of different cultures by visiting the location and reporting your findings at home after the trip. Furthermore, this way the burden of travel is equally balanced.
9. **Adjust the Meeting Times:** For any kind of virtual (regular) meeting, you must balance the burden of ‘bad’ meeting times, which includes very early mornings,

very late evenings, or even nighttime. This can be accomplished by having one site suffer the inconvenient time one week and another site suffering the following week. This also demonstrates mutual respect among all locations involved. Of course, this is unnecessary if the time zones do not differ by much.

- 10. Use the Same Communication Bandwidth for Everyone Participating in a Meeting:** Especially for the daily synchronization, we realized the importance of using the same bandwidth at all locations. Maybe you have some team members at one site who could meet in person; and at a second site, team members have access to a great video connection; and at a third site, team members only have access to audio. In this case, it is important that everyone connect via audio. This ensures that everyone—even the ones in the same location—participates in the meeting using a headset so nobody can make use of a richer communication bandwidth (e.g., face-to-face interaction) than others can.

Based on this list, it should be obvious that successful communication cannot be separated from trust, personal relationships, and respect for culture differences. Yes, some of these tips imply higher costs. However, miscommunication is much more likely to have a higher cost in the end than ensuring a successful communication throughout the whole undertaking. Keep in mind what my colleague, Ken Pugh, once said so eloquently and wisely:

“You will pay the costs of a face-to-face meeting, regardless of whether you have one or not. Not having one may cost more.”

References

Agile Manifesto. <<http://agilemanifesto.org>>

Carmel, E. (1999). *Global Software Teams: Collaborating Across Borders and Time Zones*. Prentice Hall.

Eckstein, J. (2010). *Agile Software Development with Distributed Teams: Staying Agile in a Global World*. Dorset House.

Liker, J. K. (2004). *The Toyota Way: 14 Management Principles from the World’s Greatest Manufacturer*. McGraw-Hill.

About Jutta Eckstein



Jutta Eckstein holds a MA in Business Coaching & Change Management, a Dipl. -Ing. (the equivalent to a Master's Degree in Engineering) in Product Engineering, and a BA in Education.

She works as an independent coach, consultant, and trainer from Braunschweig, Germany. Her knowledge in agile processes is based on over fifteen years' experience in project and product development. She has helped many teams and organizations all over the world to make the transition to an agile approach. She has a unique experience in applying agile processes within medium- to large-sized distributed mission-critical projects.

She has published her experience in her books, *Agile Software Development in the Large*, *Agile Software Development with Distributed Teams*, *Retrospectives for Organizational Change*, and together with Johanna Rothman, *Diving for Hidden Treasures: Finding the Real Value in your Project Portfolio*.

She also focuses on techniques that help teach and learn and is a main lead in the pedagogical patterns project. She is a member of the Agile Alliance and a member of the program committees for many different European and American conferences in the area of agile development, object orientation, and patterns. At the last election in 2011, Jutta was designated for the Top 100 most-important persons in German IT.

Jutta is a partner of IT communication, which is an alliance of two independent consultants. Nicolai Josuttis and Jutta Eckstein founded this alliance in 2004 based on the knowledge that the real reason for the failure of especially large projects is miscommunication and not technology. Therefore, the alliance helps (large) projects not only with technical expertise, but also with setting up adequate communication and team structures.

[Back to Top](#)

~~*~*~*

Conclusion

While communicating and working with remote teams can be challenging for many people, the aim of this book is to provide you with real-life tips and strategies that will not only strengthen your communications, but also your relations with remote team members.

As stated in the Introduction, communicating with local people who live near and work with you every day—those who share a common language and culture among other things—is already challenging and can even be overwhelming to some. Working with people from other countries, time zones, cultures, and languages presents an even larger challenge. When an offshore project does not play out as expected, people often name communication as the reason.

In this book, six practitioners shared their experiences with offshoring, nearshoring, and global collaboration to help you make the most of your offshore experiences. The goal of this book is to give you practical insights on *How to Communicate Effectively With a Remote Team*. The topics we covered in this book include:

- Communication Challenges in Working With Offshore Teams
- How to Communicate
- What Tools to Use
- Process, Rhythm, Metrics, and Feedback
- Remote Teams and Communication
- Oral Versus Written Communication
- The 10 Rules of Communication

We hope our experiences in offshoring and nearshoring, as well as our vast experience and supportive strategies, can help you build strong, productive teams and contribute to your success in all your future endeavors!

~~*~*~*

[Back to Top](#)

The Art of Managing Remote Teams

We hope you had a pleasant read and have gained some new and useful knowledge that will both help you in your work and bring you success! In the near future, the next book in this

series will be published: *The Art of Managing Remote Teams*. This fifth book will guide you through the aspects of communication with regard to managing remote teams.

For the latest news and updates about the release, please sign up at [our website](#).

Thank you for reading our book. If you enjoyed it, please take a moment to leave me a review at your favorite retailer. We would love to hear from you whether you want to let us know you enjoyed the book or you want to provide us with feedback.

Connect with Hugo Messer!

[Email Hugo Messer](#)

Follow me on Twitter: <https://twitter.com/hugomesser>

Reach out to me via Skype: hugomesser

Subscribe to the Ekipa blog: <http://blog.ekipa.co/>

[Back to Top](#)

~~*~*~*