

# MIND THE GAP!

THE AGILE APPROACH  
TO OFFSHORING

“Xebia’s Indian team lives and breathes Agile, and that, I think makes a significant difference for someone who is tasked with managing the complex, challenging job of distributed software development.”

---

SANJIV AUGUSTINE

PRESIDENT LITHESPEED  
AUTHOR OF “MANAGING AGILE PROJECTS”

“It is the commonly held belief that Agile and distributed off-shore developments are incompatible, this is not the case. Disciplined and skilled teams with a strong sense for the “we” are an effective way to overcome the distance. This booklet shows best practices on how to approach this.”

---

DAVID NORTON

RESEARCH DIRECTOR GARTNER

“After 10 years of highly successful co-located agile development with XP & Scrum, my department is entering a new era with virtual teams consisting of members in both Eindhoven and Bangalore. A big thank you to Xebia for showing us that Distributed Agile Development truly works in daily practice.”

---

CHRIST VRIENS

DEPARTMENT HEAD MIPLAZA  
PART OF PHILIPS RESEARCH

“The Xebia team coach provided hands-on experience gained from diverse agile projects. This was very beneficial for the implementation of Agile in our R&D organization in India.”

---

HEDWIG BAARS

HEAD R&D ERICSSON INDIA

“Xebia pulled the development of our traveler information system out of a black box and made the process flexible and transparent.”

---

MARTIEN LEGET

PROGRAM DIRECTOR INFOPLUS  
PRORAIL

## **Mind the Gap! The agile approach to offshoring**

First edition, first print, December 2009

---

This work is licensed under a Creative Commons Attribution - Noncommercial - No Derivative Works 3.0 Unported License.

See <http://creativecommons.org/licenses/by-nc-nd/3.0/> for license details.

|              |  |
|--------------|--|
| Author       | Guido Schoonheim, CTO Xebia Group            |
| Contributors | Vikas Hazrati, Maarten Winkels, Maurits Rijk |
| Sponsor      | Jeff Sutherland, Co-Creator of Scrum         |
| Design       | Commplot, Zeewolde                           |

© 2009 Xebia B.V, Utrechtseweg 49, 1213 TL Hilversum, The Netherlands.  
<http://www.xebia.com> - [distributed-agile@xebia.com](mailto:distributed-agile@xebia.com)

---

# MIND THE GAP!

THE AGILE APPROACH  
TO OFFSHORING

# Table of contents

## FOREWORD

|   |    |
|---|----|
| Jeff Sutherland, Co-Creator of Scrum          | 5  |
| Daan Teunissen, CEO Xebia Group               | 7  |
| Introduction                                  | 9  |
| Mind the Gap! - Distributed Agile Development | 11 |

## ESSENTIAL 1

|   |    |
|---|----|
| Build empowered, distributed teams with a shared vision, shared goals and shared responsibility | 13 |
|---|----|

## GUIDING PRINCIPLES

|   |    |
|---|----|
| 1.1 Work in distributed teams with team members in multiple locations. This is OneTeam <sup>®</sup> | 15 |
| 1.2 Client instills a shared long-term vision and shared short-term goals into the team             | 19 |
| 1.3 The team takes full responsibility and shared ownership to achieve committed results            | 21 |

## ESSENTIAL 2

|   |    |
|---|----|
| Offshore knowledge workers also require an open culture and relationships based on equality | 23 |
|---|----|

## GUIDING PRINCIPLES

|   |    |
|---|----|
| 2.1 Team members on both continents are considered equal            | 25 |
| 2.2 Shared culture and personal relationships triumph over distance | 27 |
| 2.3 There is no substitute for working with good people             | 29 |

**ESSENTIAL 3**

Transparent results are the only measure of progress 31

**GUIDING PRINCIPLES**

3.1 The client experience is the same and the client has as much control as working with a local team 33

3.2 Metrics: The same complete and transparent insight for client, team and Xebia 35

3.3 Working, tested software at every iteration is the only real proof that work is completed 37

**ESSENTIAL 4**

Building quality in, results in lowest development costs 39

**GUIDING PRINCIPLES**

4.1 High quality is the secret ingredient for faster delivery 41

4.2 Use the same standards and tools so that it does not matter where you work 43

4.3 Improvement is a continuous and relentless process 45

**CASE STUDIES**

ProRail PUB System 47

AlbumPrinter (Albelli) partnership 48

TBD partnership 48

About Xebia 49

# Jeff Sutherland, Co-Creator of Scrum

From its beginnings in 1993, Scrum was about hyperproductive development at 5 to 10 times the productivity of waterfall projects by empowering teams through a carefully crafted framework for collaborative work. We brought joy and enthusiasm back into software development through shared goals, transparency, respect for people, and team spirit. It required working together face to face in short sprints, and delivering working software at the end of every iteration for review and feedback by the customer.

The challenge now was to achieve this hyperproductivity with distributed work, since early research showed co-locating teams to be essential to the high performance that Scrum achieves. Distributing work across teams and geographies used to result in loss of shared work, breakdown in collaboration, and poor performance. In 2006, Xebia took up this challenge, with teams in the Netherlands and India, and successfully bridged this gap using the principle centered approach outlined in this booklet.

This makes Xebia the first service organization to reproduce sustainable distributed hyperproductivity across multiple projects. With teams one half Dutch or French or American and one half Indian, first co-located and then distributed, that demonstrated that local velocity could be distributed and that teams could cross geographies while maintaining localized velocity and quality and linear scalability. Their scaling model is remarkable as it violates Brooks law which says that expanding team size should radically reduce productivity per developer.

For the first time, the promised goals of outsourcing are comprehensively achieved – lower cost, ready access to talented developers and the ability to rapidly scale teams up and down without loss of core intellectual property, core competence, and core innovative capacity. Even more remarkably, this was done while maintaining a hyperproductive state and linear scalability of teams without loss of individual developer productivity, empowerment, and sense of accomplishment.

This booklet describes the secrets of that accomplishment – openness, transparency, equality, and sharing of work in alignment with agile principles and values. I recommend this approach to all companies striving to achieve global success in software development. It readily solves the distance and time zone barriers and the cultural and communication problems of offshoring by treating the whole team as equal partners in self-organization. This model is the key to financial success, personal growth, and customer value for distributed, offshore teams.



**Jeff Sutherland**, Co-Creator of Scrum

# Daan Teunissen, CEO Xebia Group

In 2001 I started Xebia with the conviction that we could offer much better quality and tangible results to our customers by applying real craftsmanship. Centered around a central mission of being the Authority in our field, we always kept very close to our value system and guiding principles. This has resulted today in an IT services organization of 200 people, spread across the Netherlands, France and India.

We focus on Java technology and are early adopters of – and contributors to – the Agile method. Over the years we have grown to the position of the most prominent training and consulting organization for agile software development in each of our markets.

Sharing Knowledge is one of our four Values, and we feel obliged – and proud – to share the best practices for accurate, agile, distributed software development with you. The experiences of our own software development center in India have convinced us beyond a doubt that Agile and Offshoring are a perfect fit.

Many of our clients have found their competitive edge using agile development. It is with good reason that some of the more successful venture capital companies require the IT companies they invest in to use agile software development methods. To our operations in India, Distributed Agile Development is the obvious key for success.

Stemming from our cooperation with the Scrum Alliance, we have presented our vision and experiences gained from our own small, midsize and larger projects at numerous agile conferences in the past years. We have also learned a lot from colleagues and other companies active in the field. Xebia's consultants, agile coaches, ScrumMasters and Scrum Trainers – in Europe and in our offshore center in India – apply the practices we have learned on a daily basis and improve on them where they can.

This booklet bundles all this knowledge and experience for the benefit of CIO's, IT managers and project leaders responsible for successful offshore relationships. The booklet focuses on underlying principles that you can apply to your own organization.

With high-quality yet low-cost skilled resources available from remote locations, the need for a well defined distributed agile approach has become obvious. I hope this booklet gives you the insight and practical guidance you need to truly benefit from these remote development centers.



**Daan Teunissen**, CEO & Founder of Xebia Group

# Introduction

In the past, offshoring often seemed more difficult than it was rewarding. Challenges in culture, time zone differences, physical distance, team communication and work distribution make traditional offshoring difficult. This booklet proposes a new yet proven approach to offshoring. It focuses on team dynamics and interpersonal communication using agile processes to close the geographical, cultural and linguistic distances.

Centered around four Core Essentials, this booklet offers tangible best practices, based on real world projects, to improve the output of your strategic offshoring engagements. It will show you how to deliver rapidly and with the same quality as local hyper-productive agile teams and it will demonstrate the benefits of lower cost, flexible scaling and a greater talent pool.

## Your profile

Most likely you are a CIO, IT manager or project manager dealing with offshore software development. Either for cost savings or other strategic reasons. You are interested in learning how to get offshore development working. Mixed stories of success and failure are well known throughout the market and you need to maintain control of this mainstream movement.

Quality is important to you and your team. You have initiated a meticulous process to reach the level of quality you require. At the same time however, budgets are tight, business demands are growing and you have to find the delicate balance between being open-minded to this new paradigm and safeguarding the best practices that your culture, teams

and processes have developed and implemented.

Starting small, maintaining control, total transparency, the highest quality, flexible adaptation to your organization, excellent communication and tangible results are essential to you and your team.

### Reading this booklet

Throughout these pages you will find pages in **purple**. These are the main pages explaining the principles. If you want to move straight to the high-level concepts, just read the purple pages.

The other pages discuss the principles in greater detail and explain the best practices that make it work in real life situations. If you want to know all the details, start at the beginning and keep reading.

### Stay in touch

We hope this booklet will serve as the opening line in a conversation between you and us. We invite you to apply this material to your organization, and welcome any feedback or experiences that you wish to share.

Find us at <http://xebia.com/distributed-agile/>  
email us at [distributed-agile@xebia.com](mailto:distributed-agile@xebia.com)

# Mind the Gap! Distributed Agile Development

## THE CORE ESSENTIALS OF DISTRIBUTED AGILE DEVELOPMENT

---

- ▶ Build empowered, distributed teams with a shared vision, shared goals and shared responsibility
- ▶ Offshore knowledge workers also require an open culture and relationships based on equality
- ▶ Transparent results are the only measure of progress
- ▶ Building quality in, results in the lowest development costs

For each essential we will discuss principles and practices used to make Distributed Agile Development (DAD) work.

---

Agile processes bring about a dramatic increase in productivity and quality. This is achieved through a high degree of communication and interaction, short iterative development and a strong sense of team responsibility. These are the same characteristics that make agile methods – and Scrum in particular – the ideal solution for offshoring challenges.

The agile approach involves light touch and principle-centered management. If you provide a concise set of strong principles to stimulate intelligent, responsible behavior, everyone will be able to make the right decisions. This serves to maximize everyone’s initiative and any decision that follows is automatically the right one.

This book describes the four core essentials for agile offshoring, followed by guiding principles and best practices. This is the frame of reference that makes Distributed Agile Development (DAD) work. Many details are not included, and are in fact up to the teams. After all, we in IT are in the people business.

---

*‘AS WE LOOK  
AHEAD INTO THE  
NEXT CENTURY,  
LEADERS WILL  
BE THOSE  
WHO EMPOWER  
OTHERS.’*

*BILL GATES*

# 1

## Build empowered, distributed teams with a shared vision, shared goals and shared responsibility

### GUIDING PRINCIPLES

---

- ▶ Work in distributed teams with team members in multiple locations. This is OneTeam®
- ▶ Client instills a shared long-term vision and shared short-term goals into the team
- ▶ Team takes full responsibility and shared ownership to achieve committed results

Software development is a complex environment and offshoring only complicates this further. It takes special effort to collaborate effectively with people in a different time zone, different location, different culture and different language. All these differences are forms of distance separating people. Central to the agile paradigm for successful offshoring is to approach this gap as a people problem.

By forming a single team with members at both locations, we create an environment of rich communication and shared responsibility. There are no specification hand-offs to the offshore developers. Direct human communication within the distributed team is the norm. People at both locations share the same goals, do the same work and are dependent upon each other for the team's success. Team members strengthen each other and progress is completely transparent. There is no hiding. The team as a whole integrates easily with the client since part of it is at the clients location.

IT exists to deliver value in the most effective and efficient way possible. Effective delivery is achieved when close communication and tight business feedback are used to build exactly what is required. A shared vision is crucial to guide us to the right product. The client is in control of what to build. Efficient delivery is possible when skilled people work as a team and use their own initiative, creativity and responsibility to come up with the best ways to generate the most output. Shared short-term goals provide focus to the team. The team is in control of how to deliver.

Well-distributed teams can perform at the same level as co-located local teams.

---

*'GROWTH  
AND PROFIT  
ARE A PRODUCT  
OF HOW  
PEOPLE WORK  
TOGETHER.'*

RICARDO SEMLER

# 1.1

## Work in distributed teams with team members in multiple locations. This is OneTeam<sup>©</sup>

OneTeam<sup>©</sup> is a philosophy that removes the distance between locations by creating a single distributed team. The team as a whole is responsible for the project's success. They commit jointly to the shared vision. There is no 'us-them' relationship and the whole team works together as a homogeneous, interdependent group to make the project a success.

To implement this principle we use the following practices:

**Co-locate during the project start and set up a travel plan to strengthen personal relationships, shared context and shared ownership**

At the start of any distributed project the offshore team members travel. The team works together at the client location for two or three iterations (about two weeks per iteration) before distributing and scaling up. This initial, co-located phase helps build personal relationships, set common goals and shared vision for the project and establish shared technical ownership. Travel agreements are made so team members travel in both directions at regular intervals during the project to refresh this shared context. Team members at the same location sit in the same team room regardless of discipline (developer, tester, analyst, ...).

**Involving all the team members in the scrum cycle meetings structures all the necessary formal communication**

Agile methods such as Scrum work in short iterations of between two and four weeks. These use a simple but highly effective schedule of meetings which all team members attend at all locations by way of video-conferencing. Every iteration starts with a planning and calculation meeting. During the iteration there are ten-minute meetings every day

to review progress toward the iteration goals. At the end of the iteration there is a demo to the client and a shared retrospective that focuses on further improvements. This structured cross-geography communication creates a transparent and self-improving environment, which makes the team more productive.

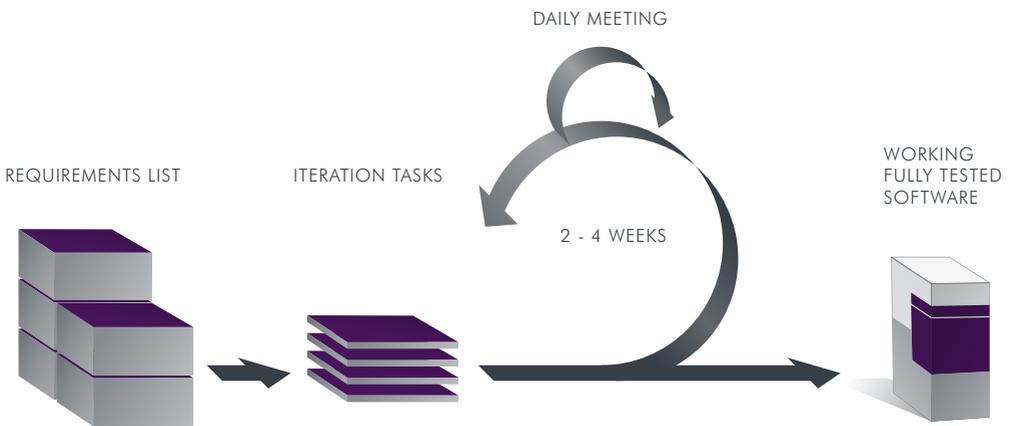
**Remove any barrier that supports the ‘us-them’ mentality – there is OneTeam® and there is no room for separation**

When working across geographies, teams need to work hard to diminish barriers such as time zones, languages and cultures. All team members make clear agreements about communication processes, working methods and standardized tools. Continuous efforts are made to build trust and keep communication open across locations. It is important that all team members are considered equal and help each other to reach the iteration goal. Success is defined at team level. Any solution that includes isolation of tasks or responsibilities at one location is not acceptable. Isolated knowledge will break communication and slow the team down on the long run.



### OneTeam® concept

Forming a single distributed team with team members spread across multiple locations is central in the OneTeam® concept. The focus on forming a single team and on how people communicate most effectively brings the best out of team members, no matter where they are.



### Scrum cycle pulls the team together

OneTeam® uses the Agile Scrum process to facilitate communication. Working in short iterations with shared goals and responsibilities ties the team together. The Scrum meeting cycle provides all necessary formal meetings, resulting in very low overhead. By working in short increments with daily meetings the team members stay perfectly in sync on all locations and reach the same results as a local team.

# 1.2

## Client instills a shared long-term vision and shared short-term goals into the team

The client is responsible for communicating the projects long-term vision. Based on this vision, the client sets short-term goals for each iteration. Knowledge workers need to know the bigger picture to make their work efficient and valuable. This context allows them to make the right decisions as a part of their daily work. A deep understanding of the client's vision and context is what makes the difference between a knowledge worker and a typist. Limiting a highly skilled knowledge worker to a need-to-know basis or detailing tasks to a level that discourages thinking will dull his interest and deactivate his most valuable tool: creativity.

To implement this principle we use the following practices:

**The client is responsible for defining what the team works on and can change priorities after every 2-week iteration**

All requirements are collected into a single work list for the project. The client owns this list and prioritizes work based on business value and cost estimates made by the team. Requirements on this list are in a user story format. With each iteration the client and the team select the highest priority user stories in the planning meeting and these constitute the workload for an iteration. The iteration workload is fixed once the iteration starts. The client can reorder the rest of the user stories at any time, thereby changing the priorities for future iterations completely if necessary.

**The team only works on tasks that are necessary and agreed upon to complete the client's priorities**

Each user story requirement includes acceptance criteria that can be verified. The team will work to meet the acceptance criteria only. Once they are met, the user story is done. The team will only work on user stories assigned by the client to the current iteration. Other work (such as major technical improvements) must follow this route before it can be picked up.

**The client repeats the aim and the 'why' behind the 'what' for every iteration to the entire team**

At the start of a new iteration the client discusses the user stories with the highest priority with the whole team. Offsite team members are included via videoconferencing. The client explains the importance of the required functionality and puts this into a larger context within the entire project and the project aim. The team asks questions about the details of the user story and discusses different ways to implement them, explaining consequences of the different solutions to the client. At the end of this planning meeting there will be a shared understanding between the client and the team about what user stories to implement, why these user stories matter and what the best way is to implement them.

# 1.3

## The team takes full responsibility and shared ownership to achieve committed results

Having a team that gets things done is mostly about making the team feel responsible and empowered. To feel responsible the team members are held accountable only for team results. The responsibility to deliver is therefore assigned to the whole team and not to specific people. The team is empowered and mandated to take all decisions about how to implement features. This gives the team the means to complete all the tasks in time and meet all high-quality standards.

To implement this principle we use the following practices:

**The whole team commits to delivering the planned iteration goals with the mandate to do whatever is necessary**

The team breaks the work down into technical tasks at the beginning of each iteration after the planning meeting with the client. The team identifies dependencies and decides on architecture. It does this with all team members on all locations involved as equals. The team commits to completing all the tasks of the user stories that they themselves have estimated to be feasible in the iteration. During the iteration the team can only change scope after reaching agreement with the client. In order to reach the iteration goal the team can adapt technology as necessary.

**During a daily 'standup meeting', team members across locations discuss what tasks to pick up and how to help each other**

A standup meeting is a daily status meeting taking no more than 15 minutes. It is called a standup meeting because all attendees remain standing as this keeps things short.

Each team member answers three questions:

1. What did I accomplish yesterday?
2. What can I best do today?
3. What is hindering me?

The whole team and the client attend the meeting. This is the platform to synchronize daily planning and tasks, and to help each other.

**Personal disciplines (tester, analyst, developer...) are less important than achieving the iteration goal together**

Many disciplines are needed to create software, often represented by different persons in the team. Even so, the whole team – not specific people – bears responsibility for the results. Within a team specific people can have specific knowledge or experience that makes them more suited for specific tasks, but completing these tasks is the responsibility of the whole team. The focus of experts within the team must be on enabling the whole team to work on tasks of which they have expert knowledge, not on completing these tasks themselves. This means, for instance, that a developer in the Netherlands will take up part of the tasks of a tester in India if that is what is required to make the iteration a success.

# 2

## Offshore knowledge workers also require an open culture and relationships based on equality

### GUIDING PRINCIPLES

---

- ▶ Team members on both continents are considered equal
- ▶ Shared culture and personal relationships triumph over distance
- ▶ There is no substitute for working with good people

The difference in performance between a worker going through the required motions and a true knowledge worker who is passionate in both heart and mind is a difference of night and day. Still, traditional offshoring providers very often treat their staff as simple factory workers. This results in low output and a high number of defects.

Knowledge workers in other countries have needs similar to those in your local workplace. Responsible work in an open culture where expression and communication are not limited. When people are treated as equals, they become motivated developers.

We have found that people everywhere enjoy doing a good job, if you let them.

The next thing that makes a big difference is the engineer's seniority and level of skill. That's why hiring and knowledge management are the most important processes in a knowledge company.

A company culture of openness and excellence combined with working as equal members within agile teams is the reason why Xebia has less than 5% attrition in both India and Europe.

---

*'20 YEARS AGO, IT WAS ABOUT GOODS AND SERVICES. TODAY, IT'S ABOUT KNOWLEDGE WORKERS.'*

STEPHEN R. COVEY

## 2.1

# Team members on both continents are considered equal

A key ingredient for motivated knowledge workers is an environment that allows them to take responsibility. Distributed teams work as a single team and there are no differences in role, responsibility or mandate between team members on either side. Not everyone is the same – people have personal expertise. This is not mandated by where you reside. It is mandated only by how skilled you are.

To implement this principle we use the following practices:

### Team members at both locations have an equal say in software architecture and design

Architecture is a team activity in which all members are expected to participate. The traditional model in which an onshore architect lays out working instructions for offshore developers is simply not done in Distributed Agile Development.

Architecture and design decisions are described in living documentation, such as a Wiki. This allows the whole team to access and update them easily. All decisions are documented meticulously, complete with the reasoning behind it. The documentation also includes alternatives explored. Discussions involving multiple team members ensure that knowledge about the decisions is shared between locations. Tools such as electronic whiteboards, conference calls and desktop sharing help communicate ideas and opinions clearly and openly. No architecture decision is sacred. There is always room for improvement by any team member.

**Work is done using ‘pull’ planning and team members at all locations pull from a single list of iteration tasks**

Following the iteration planning, all work is broken down into a single list of technical tasks. Every team member or pair picks up tasks from this list to complete. Each person must complete his task before starting work on a new one. Each team member decides personally what task to pick up, judging how to optimize learning, productivity and interest. In some cases, tasks can only be completed at a certain location, for instance, because of security constraints, bandwidth or accessibility. This is the only constraint for working on a task.

**Feedback on quality and performance is open and bi-directional**

At the beginning of the project the team convenes for a ‘Norming and Chartering meeting’. This is used to establish common practices and rules about how the team works, including working hours, code quality agreements, versioning and build procedures, and tools to use. This way code quality becomes objective and measurement is automated. Each location reviews code written on the other location. After every iteration the team completes a subjective questionnaire with items such as ‘how do you feel about the quality and productivity delivered onshore/offshore?’ This brings issues into the open.

## 2.2

# Shared culture and personal relationships triumph over distance

There is a secret to working effectively across distances. It means occupying the same ‘mind space’. If the person on the other side is someone you know personally, someone who shares your context, your way of thinking about problems, your working culture and technical standards – and that person is someone you respect – then distance is no problem at all. This makes building this shared ‘mind space’ the primary concern when starting a new Distributed Agile team. Afterwards the team members’ understanding of the other location’s cultural dynamics and strong personal relationships will overcome the barriers of distance in communication.

To implement this principle we use the following practices:

**During travel on a project, the first priority is to share, to learn and to build relationships**

During the initial co-location period at project start – and for all other project travel – emphasis is placed on building relationships and sharing context information. Team members from different locations pair program together, have project discussions together, have lunch and drinks after work together. There is a lot of implicit information at the client site that needs to be transferred to visiting team members. It works very well for the host team members to take the visiting team members out sightseeing. This allows team members to get to know each other both personally and professionally.

**Relationships formed enable team members to see each other as individuals and to focus on similarities**

Personal relationships make the difference between ‘that difficult resource that does not do what I want’ and ‘Vikas my buddy who I enjoy talking to about technical issues’. Engineers around the world are excited about the same things – technical challenges and a job well done. Keeping the focus on these similarities is more important than any of the ways we differ from each other. This allows the team to use individual strengths and not get bogged down in differences. By focusing on the positive similarities, teams work together to reach major increases in productivity.

**Company culture and project culture are based on the same strong principles and value system. These are more important than personal differences**

The cultural values of an open workplace are shared at all locations. The project culture is defined explicitly. This acts as a strong reference that does away with personal differences. In these values we are all alike. Personal differences that hamper team productivity quickly come to the surface in the iteration evaluations, where both team and management can sort them out.

## 2.3

# There is no substitute for working with good people

As mentioned in the book *Good to Great* by Jim Collins, it is essential to have the right people 'on the bus'. Working with senior, motivated people strongly increases productivity and helps do away with problems related to motivation and people management. This makes hiring one of the most important processes.

To implement this principle we use the following practices:

### Hiring the same level of talent at both locations creates trust and respect

When working across great distances it is important to work with people who truly are your equals. This means applying the same strict standard for hiring talent at different locations. This establishes a shared definition of what it takes to work here. This healthy parity across geographies eases communication and creates a feeling of trust and respect.

### A good professional brings communication skills and proactivity to a team regardless of location or culture

A good professional offers more than just technical skills. Skills can be trained, but they will be meaningless without the right mentality. A professional mentality includes a proactive approach to problem solving, an eagerness to learn and commitment to perform at the highest level. The mental ability to think abstractly and the personal skill to communicate clearly are just as important. Professionals are vocal during discussions and have opinions backed by logic. While communication patterns may vary between cultures and persons, all professionals bring these attributes to a team regardless of location. This helps steer the teams in the right

direction and manage and resolve the impediments that hinder team productivity.

### **The 5x productivity difference between skilled and unskilled knowledge workers makes it cheaper to work with seniors**

A common misconception is that relatively less knowledgeable and more junior team members can save on project costs. The difference in productivity between skilled and unskilled knowledge workers can be as much as five times as great. This offsets the increased hourly cost for senior developers and translates into better economics and a better time-to-market for any project.

# 3

## Transparent results are the only measure of progress

### GUIDING PRINCIPLES

---

- ▶ The client experience is the same and the client has as much control as working with a local team
- ▶ Metrics: The same complete and transparent insight for client, team and Xebia
- ▶ Working, tested software at every iteration is the only real proof that work is completed

The best way to gain accurate insight into development is to see the working product. The most important rule of project management is to avoid surprises. Working in iterations of between two and four weeks allows us to see the actual progress in short intervals. This makes it impossible for development to ‘submerge’ and come up later with unpleasant surprises.

Delivering working, fully tested software every iteration makes the process easy to measure. These measurements are shared among all stakeholders, clients and suppliers. There is no hiding when the only measure of progress is working product increments at short intervals.

By measuring the actual amount of work done at every iteration, a realistic plan is extrapolated. To monitor the progress reliably, the product increments have to be fully tested and working. This also avoids the usual rush at the end of the project to stabilize or bug-fix.

This pattern applies all the more to offshore development, where progress and quality are often unclear due to distance and communication issues. By working in a single distributed team, the client can monitor progress and interact with the developers just as well as if the team is local. Providing this transparent experience is a golden rule at Xebia.

---

*‘FACE REALITY AS  
IT IS, NOT AS IT  
WAS OR AS YOU  
WISH IT TO BE.’*

JACK WELCH

# 3.1

## The client experience is the same and the client has as much control as working with a local team

With a good local agile team, clients have direct access to the engineers. They can interact first-hand to communicate their outlook and give feedback. They have complete insight into progress on a daily basis. The language used is the client's language. By creating a distributed team with part of the engineers at the client's location all these benefits apply even though the bulk of the work can be done at an offshore location. Clients have direct access to and interactions with all team members and see detailed progress reports with working software every two to four weeks. Difficult nuances are discussed with local team members in the client's language.

To implement this principle we use the following practices:

**Knowledge is always available locally because both sides of the team work on the same parts of the system**

Both sides of the distributed team work on the same parts of the system. There is no separate code ownership. Team members regularly extend functionality implemented by other team members when later user stories require them to do so. Team members across locations talk face-to-face each day using audio and video channels to discuss design. Having knowledge of the system at both locations enables the client to scale up or down without a knowledge lock-in and prevents unproductive dependencies.

**Onsite team members interface with the client in the local language.  
Translations are not the client's problem**

The client always enjoys the benefits of the local language with the onsite team members. For the distributed team members the communication is either held in English if the client agrees or the onsite team translates and explains the stories and discussions after the planning meeting. The client is not bothered with the translation process and can still enjoy the benefits of a distributed team.

**Remote staff can be scaled up or down without impacting client local organization and interfaces**

Depending on the amount of work and other issues such as time-to-market and desired productivity, the client can scale the team up or down at both the local and the remote location. Since the same amount of knowledge is available across locations, it is easy to scale up a team when desired and to scale down after goals have been met. By retaining local engineers, the client can interface with the same people the same way even after scaling down. By retaining remote engineers the client gains major financial benefits. Because there is no knowledge lock-in, the client actually has access to these strategic options.

## 3.2

## Metrics: The same complete and transparent insight for client, team and Xebia

There is no hiding in agile development. Detailed measurements on tasks completed provide daily updated progress reporting. The intense interaction between team members reveals personal performance. Measuring quality is automated using static code analysis and extensive automated testing. The client, the team and Xebia get the same dashboards and progress reports. There is full transparency!

To implement this principle we use the following practices:

**During the iteration the team creates a public graph showing daily progress toward the iteration goal**

Every day during the short daily standup meeting, the team updates progress information based on the size of the tasks completed. This results in a “burn down” graph indicating team progress toward the iteration goals. The graph helps the client track the teams progress on a daily basis and works as an early warning system. This enables the team to take corrective actions when necessary to meet the iteration goals.

**Actual speed of progress and total work supply are measured, creating a projected planning based on reality**

The client maintains a single estimated and prioritized list of all work for the project. By measuring the amount of work the team performs in every iteration, and by plotting this against the total estimated workload, the team produces a projected planning. This is fundamentally different from drawing up a detailed Gantt chart at the start. The motto is, ‘plan the work, don’t work the plan.’ This accurate, evidence-based prediction

allows the client to steer the project according to all aspects of the Project Management triangle: Cost, Time, and Scope.

### **Quality is measured using real-time dashboards**

Every project has a project dashboard which not only tracks bugs but is a comprehensive collection of reports which depict the health of the project and status of the work. This includes iteration 'burn down' charts, team speed, static code analysis, test coverage reports, performance and scalability reports, build status reports etc. All this information is aggregated on a dashboard which is transparently shared with the entire team and the client. This makes quality very clearly measurable and automated alarms are triggered as soon as a quality aspect, such as one of the many automated tests, fails.

## 3.3

# Working, tested software at every iteration is the only real proof that work is completed

The proof of the pudding is in the eating. No amount of documentation or analysis can tell us more than actual working, fully tested software. Working software provides proof of progress and enables the client to offer feedback on functionality and set the project's future direction.

To implement this principle we use the following practices:

### Every iteration delivers a working product

Following the initial project setup, the team produces working software every iteration without exception. Work is only considered done when it meets stringent quality standards and is of production-ready quality. Even when an iteration focuses on architecture, risk management or candidate design identification, it has working software as its end product. This constant visibility means there can never be big unpleasant surprises when it comes to developers' quality or progress.

### The team presents the results to the client at the end of the 2-week iteration

During the demo at the end of every iteration, the team shows all newly delivered functionality to get immediate feedback from the client. This keeps the team in sync with business expectations and shows the client that the team is building the right product. Frequent demos allow for maximized learning, and ensure that the development team keeps adding the highest value at every iteration.

### Only tested features are included in the demonstration to show real progress

Testing is an essential component built into the fabric of development. If a feature is to be included in the demonstration, and if it is to count in the progress metrics, it has to have passed the most stringent and extensive tests. This ensures that all delivered work is of production ready quality. All 'hidden' work is removed from the project by working this way, and the entire system can go into production without further unexpected bug fixing or 'stabilizing' stages. You can only track real progress by delivering production ready quality at the end of every iteration.

# 4

## Building quality in, results in lowest development costs

### GUIDING PRINCIPLES

---

- ▶ High quality is the secret ingredient for faster delivery
- ▶ Use the same standards and tools so that it does not matter where you work
- ▶ Improvement is a continuous and relentless process

After a few years in IT, there's a good chance that you will have seen a variety of development projects come to a grinding halt because they are suffering from poor quality. Growing numbers of defects and poor code can become so entangled that it is impossible to modify the codebase effectively, causing projects to derail after the initial stages.

Agile development has shown that writing code the right way does not slow you down, even on the short term. Instead it actually speeds up development; it just requires greater discipline and engineering skill. Development costs are reduced significantly because of the short feedback loop that integrated functional testing provides. Creating a proper testing basis and a healthy clean architecture provides flexibility and the assurance that allows you to take the product where the business goes, not the other way around.

Greater speed and flexibility through quality is only achievable if you and your offshore provider share the same level of quality. Synchronizing according to the same standards and tools for development, collaboration and knowledge-sharing is an important basis.

The best way to control quality and stay on top is through a continuous process of evaluation and improvement. The principles of lean manufacturing that made Toyota the world leader that it is today are the proper guidelines for this ongoing improvement process.

---

*'QUALITY MEANS  
DOING IT RIGHT  
WHEN NO ONE  
IS LOOKING.'*

HENRY FORD

# 4.1

## High quality is the secret ingredient for faster delivery

It seems counterintuitive – higher speed through higher quality. We are used to thinking in terms of trade-offs. The secret is that by not compromising on quality you create maintainable software that is easy and cheap to modify. This results in higher productivity and faster delivery. Often the best way to speed up a project is to invest in reducing code complexity and refactoring the architecture.

To implement this principle we use the following practices:

**A feature is only finished when it matches the ‘Definition of Done’, including any architecture refactoring, tests and documentation**

The team uses an elaborate list of criteria which it applies to every feature to determine if it is done. This ‘Definition of Done’ (DOD) is a fixed list that applies to all work. It includes quality considerations such as test coverage, functional correctness, completeness, complexity, readability, performance, bugs, architecture and documentation. Once a story satisfies these criteria, is it deemed complete.

**All technical and functional tests are automated so they can be performed continuously throughout future iterations.**

Before a team starts work on a new functionality, the members decide how to automate the tests. The developer’s first responsibility is to enable the tester to start work by making any necessary extensions to the testing framework. The tester and developer then begin work in parallel. The tester writes automated tests and the developer creates the functionality. This test automation is done using tools such as Selenium, Fitnesse,

Watir, etc. Once both are complete, the teams run the tests and resolve any issues that might arise. From that moment on, the tests are included permanently in the automated testing process as part of the continuous build.

**Requirements are made mature just in time for the iterations – requirements that do not match the checklist are not accepted into the iteration**

In keeping with the lean manufacturing philosophy of maximizing flow, requirements are detailed just in time. Before requirements are accepted into an iteration, they have to be in the ‘ready’ state. In other words, they should be in a state that allows the development team to start work on them. Requirements that are not in ready would add waste to the development process as time would have to be spent detailing them during the iteration, thereby holding up other activities.

## 4.2

# Use the same standards and tools so that it does not matter where you work

Common development standards and toolkits across the distributed team improve productivity and communication. Teams can talk to each other using the same terminology as they resolve common issues. The use of common standards and tools increases the OneTeam<sup>®</sup> feeling. It does not matter if a developer opens his laptop in the Netherlands or in India; his working environment is the same.

To implement this principle we use the following practices:

**The team holds a norming session at project start to make explicit agreements concerning technique, quality and process**

At the start of the project the team decides on the details of the standards and procedures to follow. All team members consciously agree with the choice of tools and processes and commit to it. This conscious agreement is the basis for all future discipline. Since each team member has a say in the process, the final agreement on tools, techniques, quality and process is something that everyone on the team must adhere to. Examples of decided standards are the development environment, when to check in your code, what the time of the stand-up meeting is etc.

**The team works on a single technical platform for version control, software-building, continuous integration, etc.**

The team sets up a shared infrastructure at a single location, which all team members use across all geographies. This includes a common server for version control such as subversion, where all code is checked in from all locations. Common automated tools are used to build the software

and continuously integrate the software committed from various locations. This shared setup ensures that team members work together as a team and are more effective.

#### **Online collaboration tools provide close contact between team members at different locations**

All team members have Instant Messaging accounts which are shared with the rest of the team. All team members are readily available online for discussions, brainstorming sessions, distributed coding and working together during the overlapping working hours across geographies. Teams have always-on video links that offer a virtually co-located feeling by enabling them to share live video feeds from other locations in their team room. Other shared tools like issue tracking software and a Wiki ensure that everyone on the team is on the same page. Finally, the rule is to just pick up the phone whenever technology fails. Effective human contact is always more important than saving a few bucks on a phone bill.

## 4.3

## Improvement is a continuous and relentless process

One of the most valuable aspects of Distributed Agile Development is that it enables rapid learning and improvement through short iteration cycles. Evaluations are held and feedback is requested after every iteration, after every major release and after the project is completed. Concrete improvements are implemented every iteration. These improvements focus on hard factors such as technology and performance quality and on less quantifiable factors such as collaboration, team spirit and communication. Combined with a quality monitoring system driven by company best practices, this creates a hyperproductive environment after just a few iterations.

To implement this principle we use the following practices:

**The whole team evaluates each iteration looking to remove waste and to improve upon communication, process and technology**

After each iteration, the entire team does a retrospective evaluation to determine what went well and what can be improved. Everything that went well is carefully preserved and the team selects the top three improvements to be carried out in the next iteration. Making these structural improvements becomes part of the workload for the next iteration.

**The team sets working standards aiming to improve them during the project**

One of the principles of lean manufacturing states that if you don't have a standard, you can't improve upon it. At the start of the project, the team sets the standards to follow during software development. This includes

practices such as pair programming and definitions such as what adequate documentation looks like. Standards are also determined for quality metrics. These standards are a starting point and the team raises the bar after every iteration, rapidly streamlining the process.

**During regular project intervals, Xebia Quality Management checks for all previously encountered risks in projects**

Xebia maintains and updates a Quality Management System (QMS) that includes all the lessons learned and risks associated with previous projects. An internal quality consultant not working on the project performs periodic analyses. This analysis looks at more than technology and includes project management, client expectations, delivered results, etc. This experience driven risk assessment makes sure that all relevant risks are addressed, making the development process more efficient and predictable.

# Case studies

Distributed Agile Development delivers complete end-to-end projects as well as strategic offshoring partnerships to Xebia clients. In many cases we create a distributed team involving the client's on-site engineers and Xebia India's off-site engineers.

The following three cases are examples of the benefits achieved:

## Case 1: ProRail PUB System

|                            |  |
|----------------------------|--|
| <i>Client</i>              | ProRail (InfoPlus program)   |
| <i>Project description</i> | Winner Dutch Computable Best Business Case award. Build a new information system for the Dutch Railways to display train information on digital screens, provide operator interfaces, and play audio messages at all the train stations in the Netherlands. Project delivered at 6.6 hours per FP with 'excellent' independent audit remarks. <a href="#">Download the case for details.</a> |
| <i>Engagement type</i>     | End-to-end project delivery  |
| <i>Project size</i>        | 40+ man years, 20 people in 4 teams at project peak, half in the Netherlands half in India   |
| <i>Main driver for DAD</i> | Transparent progress with optimal control  |

### Case 2: AlbumPrinter (Albelli) partnership

|                            |  |
|----------------------------|--|
| <i>Client</i>              | AlbumPrinter (winner of the Deloitte Fast 50 2008, the Netherlands fastest-growing company)  |
| <i>Project description</i> | Extend the AlbumPrinter Agile Development team to provide speedy delivery of a new client platform based on Adobe Flex technology. |
| <i>Engagement type</i>     | Strategic partnership  |
| <i>Project size</i>        | 4 Xebia India engineers, long-term   |
| <i>Main driver for DAD</i> | Gain the same benefits of local agile development with spread risk and lower cost  |

### Case 3: TBD partnership

|                            |   |
|----------------------------|---|
| <i>Client</i>              | TBD social networking website   |
| <i>Project description</i> | Extend the development team based in San Francisco on a project basis. The DAD model was adapted to deal with the large (12.5 hour) time difference. See case download for details. |
| <i>Engagement type</i>     | Scaling up a project temporarily  |
| <i>Project size</i>        | 4 Xebia India engineers, 8 months   |
| <i>Main driver for DAD</i> | Agile quality and productivity with offshoring benefits   |

More information: <http://xebia.com/distributed-agile>  
or [distributed-agile@xebia.com](mailto:distributed-agile@xebia.com)

# About Xebia

We believe in principle centered leadership. The following mission, values and people philosophy are the same for all parts of the Xebia Group, India and Europe alike.

## Mission & Focus

We aim to be the leading authority in our markets. Doing the best possible job is what all Xebians have in common. When it comes down to it, we value quality over growth.

We focus on Java technology, agile methodologies and offshoring.

## Value System

Xebia's value system has 4 principles that are core to our success:

- People First
- Quality without Compromise
- Customer Intimacy
- Sharing Knowledge

Everyone in the organization understands how to apply these principles. This thorough understanding empowers all Xebians to take ownership and to independently make the right decisions on any level and in any situation.

## People

We work with people who want to be the best in their field and thrive in an organization of peers with the same ambitions and potential. Our thorough hiring procedure includes triple interviews, hands-on technical assessments and strong attention to soft skills and personal drive. We only hire people who can add to our authority and thought leadership.

This great group of people keeps each other sharp through intensive mandatory knowledge sharing and frequent sessions with external speakers. Everyone is expected to help shape our organization according to our value system.

## Key figures

Xebia was founded in 2001 and currently has operations in the Netherlands, France and India. Craftsmanship, productivity and quality have driven a steadily growing revenue resulting in over \$20 million in 2008.

Xebia currently employs 200+ people worldwide with an international customer base consisting of both large corporations and smaller highly successful companies.

Technologies complementary to Java such as Adobe Flex, Scala, Enterprise Portals & CMS are part of our competence.

“Distributed development is extremely challenging and prone to failure. Xebia’s approach, grounded in agile principles and practices, is one of the best that I’ve seen. This booklet shares the key strategies for how they do it.”

---

PETE DEEMER

CO-FOUNDER, SCRUM TRAINING INSTITUTE

“To build our website, we have chosen the Distributed Agile Delivery Model from Xebia because we were sure to have the best price, to drive the development by business value, shorten delays and end with high code quality.”

---

THIERRY PÉPIN

CEO COACHCLUB

“Anurag, Xebia India’s MD, has created an offshore team that lives and breathes Scrum, it’s deeply rooted their genes, and it positively affects your own team”

---

WINFRIED VAN HOLLAND

CTO HUMAN INFERENCE

“Not only did Xebia develop a working platform within three months, but it also removed our aversion to outsourcing and offshoring.”

---

ADRIAAN VAN DER HEK

COO COMPARE GROUP

“Xebia allows us to scale up at a high level of quality. It is even advantageous in terms of cost. But that is not the deciding factor. We would not have taken this opportunity had it not been for the quality Xebia can deliver with its specialists in India. Quality comes first.”

---

KLAAS WASLANDER

CTO ALBUMPRINTER (ALBELLI)

“Xebia Agility has been used in an extremely beneficial way in our project. Our project would not have been successful without it.”

---

JÉRÔME CHELLET

PROJECT MANAGER PUBLICIS GROUP

“Offshore software development has a much higher success rate with agile methods. In my experience Indian engineers are generally shy. Xebia India has a very strict recruitment policy with the same assessments as in the Netherlands. That is how I am sure to get the right people in the team.”

---

ANDREW DE LA HAYE

COO RIPE NCC

## XEBIA GROUP

---

XEBIA INDIA  
612 BPTP Park Centra  
Sector 30,  
122002 Haryana  
India

T. +91 (0124) 420 00 51  
F. +91 (0124) 420 00 52

[www.xebia.co.in](http://www.xebia.co.in)  
[infoindia@xebia.com](mailto:infoindia@xebia.com)

---

XEBIA NETHERLANDS  
Utrechtseweg 49  
1213 TL Hilversum  
The Netherlands

T. +31 (0)35 538 19 21  
F. +31 (0)35 538 19 22

[www.xebia.com](http://www.xebia.com)  
[info@xebia.com](mailto:info@xebia.com)

---

XEBIA FRANCE  
La Défense Colisée  
10/12 Avenue de L'Arche  
92419 Courbevoie Cedex  
France

T. +33 (0)1 469 176 16  
F. +33 (0)1 469 188 00

[www.xebia.fr](http://www.xebia.fr)  
[info@xebia.fr](mailto:info@xebia.fr)