Discovering Strategies to Improve Business Value in Outsourcing Projects

Maria Laura Ponisio, Pascal van Eck
University of Twente
The Netherlands
{m.l.ponisio, p.vaneck}@utwente.nl

Peter Vruggink
Logica
The Netherlands
peter.vruggink@logica.com

Abstract

This paper deals with the problem of leveraging client business value in a software development outsourcing relationship. We have observed software development projects from two different Dutch IT outsourcing companies and studied the approach they apply in their (successful) projects. The results show that they create a role dedicated to facilitate communication. This arrangement has the potential to put team members in a better position to communicate, facilitating the transfer of information supporting the rationale behind design decisions. Teams are thus better equipped to anticipate change and to react faster in solving everyday problems. This paper describes our observations and the practical implications we expect, such as the improvement of re-buy intention on the client’s side.

Keywords: outsourcing, software development, quality software system, project management

1 Introduction

Outsourcing of IT functions has been flourishing over the past decade and continues to do so. Yet, in many cases, outsourced IT projects fail [12], more than two thirds of IT projects do not deliver what was promised on time or within budget [18] and customers end up so unsatisfied that they would rather try to avoid asking the insourcing company as provider in another outsourcing project [21]. When the task being outsourced is the development of new software, IT practitioners need to produce high-quality software systems that leverage the client’s business value.

Our long-term goal is to improve the alignment of IT solutions and business goals. We are looking for guidelines that help IT providers to build solutions that leverage their customer’s business value. Thus, in this paper we take the provider’s point of view and we search for guidelines to tackle critical factors affecting success or failure of outsourcing projects.

It is clear that software systems developed by providers should help clients to improve their businesses. However, the client’s business domain is usually difficult to understand, communication is poor and devised solutions are difficult to implement. To deliver a software product that leverages client’s business value, the in-house team (client’s side) and the provider team need to anticipate challenges, share concerns early in the project, check satisfaction regularly and build the relationship, even when in-house team members may feel threatened by the decision to bring in an outside service provider [1]. It is necessary to ensure transfer of knowledge by opening communication channels.

Existing theory stresses the importance of such communication and suggests that companies devote employees and time to ensure and facilitate communication. However, there is little research explaining how this can be done. In this paper, we take the first steps to move from theory to practice and answer the question whether we can find communication facilitators empirically in software development projects and how they work.

Specifically, based on elements present in existing theory (Section 3) this paper reports their use as we have observed in practice in three different outsourcing scenarios (Section 4). Our findings relate to the implications of applying the theoretical elements present in the outsourcing theory. Our study confirms that communication facilitators (i.e., employees whose role in the team is to increase communication) is indeed used in practice, presenting several advantages in the context of outsourcing (Section 5).

At the level of software development practice improvement, the development team benefits from better knowledge of the final software environment. Developers have less doubts about requirements, which in turn has the potential to increase usability of the final software product [13]. In addition, requirement change is unavoidable, but the experts whom we interviewed had the certainty that their team had the flexibility to deal with it, because communication about these changes was facilitated.

At the level of project management, client and provider are in a better position to anticipate changes. Moreover, the responsible person in the in-house team feels reassured,
having at hand all the necessary information justifying every important decision.

We have observed these benefits in the cases studied. Although we cannot make claims about other cases without further research, we do believe that the potential benefits are also present when adding communication facilitator teams at higher managerial levels in the organisation. In that situation, instead of using a single person, the organisation would create an entire organisational unit dedicated to facilitating client-provider communication. More implications that we see for further research are presented in Section 6. In addition, we have analysed the validity of our study and confirmed our findings with experts. Section 7 discusses validity and includes the insight from experts from the outsourcing field.

2 Problem

Our long-term goal is to improve software development practice to maximise the client’s business value. By client’s business value we mean the value of the IT developed in an outsourced project to the business of the organisation that is outsourcing the service of developing software. All observations have been performed with this goal in mind. A key factor identified in theory influencing the success of outsourcing projects is communication between client and provider. Existing literature recommends managers to pay special attention to facilitate communication, a concept that we will refer to as the communication facilitator (CF) strategy.

The problem tackled in this paper is two-fold: can we find projects where the communication facilitator strategy is actually used in the current outsourcing practice? In addition, we need to learn how we can apply the communication facilitator strategy to improve the client’s business value and the quality of the software developed.

To do the empirical research required for this we have to design the research first. We have to decide if we are going to use, for example, questionnaires, focus groups or surveys and we have to decide how to validate the outcome of it [22]. In our case we decided to interview outsourcing experts, use questionnaires, analyse obtained data and draw conclusions. We validated the outcome via interviews with other experts in outsourcing projects in the same domain.

In this paper we focus on software development, and results obtained in this paper are validated for outsourcing IT where development of software plays a role. Our claims are thus constrained to this scope.

2.1 Terminology

We assume that there is an organisation (i.e., a certain group of people) that needs a particular software system to support them in their work. This organisation can be a department of a company, a collection of departments, or even an entire company or government agency. We call this organisation (which comprises direct users of the software program, but also their supervisors) the client organisation or client. The client organisation does not develop the software program itself: it commissions this to another organisation that we call the provider organisation or provider. The provider can for instance be the IT department within a large company, or a separate company dedicated to software development for its commercial customers.

The fact that a client organisation does not develop software itself gives rise to two phenomena that are sometimes mistakenly taken as synonyms: outsourcing and offshoring. With outsourcing, we mean that the client organisation commissions development of a software program to a provider organisation that is economically independent (outside of the hierarchy of the client); this creates a commercial relation between the client and the provider. With offshoring, we mean that the client and the provider are geographically far apart, e.g., the client is located in Western Europe, while the provider is located in a low-cost country in Asia. Outsourcing and offshoring are independent phenomena and therefore, all combinations are possible: Offshore outsourcing, Domestic outsourcing, Offshore in-house development, Domestic in-house development

Let us disambiguate the terms. First, in Offshore outsourcing the client organisation commissions software development to a provider organisation that is outside the client’s hierarchy and located in a distinct geographical location, e.g., a Dutch bank outsourcing software development to an Indian developer. Secondly, in Domestic outsourcing, the client organisation commissions software development to a provider that performs software development in the same geographical area as where the client is located. This is the case in one of our examples. Then, in Offshore in-house development a large multi-national organisation locates its software development department in a remote location. This is the case in another of our examples. Finally, in Domestic in-house development an organisation has its software development located geographically close (in the same country) as its users.

The results presented in this paper apply to all four combinations of offshoring and outsourcing listed above.

2.2 Research Method

In order to discover whether the communication facilitator strategy is used in practice and to learn from practice beneficial ways of applying it to improve the client’s business value, we need to follow some steps.

First, we need to identify references to this idea in existing literature. Secondly, we need to make sure that we are
talking about the same thing and that the elements found actually help to solve our problem. Then we need to eliminate ambiguities. For instance, we can aggregate solution elements into a definition of communication facilitator. Perhaps we will have to increase the level of abstraction, but this will be justified by the fact that we need to have a clear definition that is abstract enough for everybody to recognise what we mean by communication facilitator. The objective behind this is to eliminate ambiguity, which we need to discuss with members of different disciplines [19]. Finally, we need to evaluate whether the theoretical concepts appear in practice, and if they do appear, we hope we can learn something by observing how they are used.

The research method we chose does not allow us to make any statistically significant claim for the whole population of outsourcing relationships based on the few relations studied. Case study research is not the same as statistical sampling; generalization in case study research follows other mechanisms [23, 14]. In this paper, we derive empirical statements from existing theory, this is what Lee and Baskerville call ‘type TE generalizability: generalizing from theory to descriptions’ [14, p. 237].

We have identified the problem, long-term goal and scope boundaries. The rest of the paper follows the structure of our research. This structure draws from established processes of research [22, 7, 23], and is as follows: we perform a literature review and aggregate related concepts into a conceptualisation of communication facilitator (Section 3), we observe outsourcing practice, performing interviews in selected cases of practice (Section 4), we compare theory with data gathered from practice, deriving and reporting practical implications (Section 5) as well as implications for further research (Section 6), and finally, we verify our findings with experts and report their insights (Section 7).

3 The communication facilitator in theory

Not surprisingly, it is generally accepted as good practice to implement means that facilitate communication between client and provider. From behavioural models aimed at improving the quality of large software systems and from our own experience, we know that “developing large software systems must be treated, at least in part, as a learning, communication and negotiation process” [3]. This seems to be the way to deal with well-known problems such as low application domain knowledge, fluctuating and conflicting requirements, and communication and coordination breakdowns. From this perspective, previous research claims that, in general, introducing a communication facilitator is beneficial.

We consulted relevant literature in search for concepts related to improving outsourcing practice from the perspective of service providers and project management. From this perspective, the communication facilitator role was first identified by de Meyer [4], and it re-appeared in research studies ever since. For instance, a recent empirical study of elements that contributed to the successful outcome of projects involving virtual project teams observed the project leader taking information from internal and external sources and directing it to the appropriate persons. In this study, the project leader has a caretaker/communication facilitator role, taking particular significance when the task is to overcome problems of complexity [20].

We have found related guidelines aimed at improving communication flow between in-house personnel at the client side and developers at the provider side that did not mention explicitly the word communication facilitator. Examples of those are up-front sharing of concerns and potential solutions, early meetings with stakeholders, check satisfaction regularly, share project vision, discuss project scope and build up the relationship [1]. In short, have strategies to improve communication.

To succeed in outsourcing, practitioners recommend to follow guidelines along the lines of “have your own employee at the vendor’s site” [12]. Another guideline proposed recently is to “appoint a contact person per remote team” [16]. This guideline supports the formation of social ties from the beginning onward and at the level of team activities.

Work on requirements transfer during outsourcing detected the use of a communication facilitator among other factors relevant to success of transferring requirements [11]. In order of importance, the communication facilitator was found to be the sixth most important quality attribute. According to this study, more important quality attributes were consistent information, reduced ambiguity, required skills and experience in development team, verification of requirements, and having a minimum requirements document needed for design.

3.1 IT governance

From the IT governance perspective the need has been recognised for a counterbalance to the interests of the software development organisation, which a communication facilitator can provide. The primary concern of the project leader (and, therefore, of the other project members such as the developers and testers) is that the contracted product is developed on-time and within budget, and that risks are properly managed. In other words, the primary concern of the project leader is the question “Are we getting the project done well?”. The focus is on the process. To the contrary, the primary concern of the communication facilitator is the question “Are we doing the right things [for the client]?”
3.2 Inter-organisational networks

The inter-organisational network perspective extends existing software development methodologies by taking into account the inter-organisational network and (when appropriate) even commercial context in which the client and the development organisation are embedded. In a software development relation, the client organisation wants to avoid situations in which the development organisation can exploit a power imbalance.

The client organisation will thus favour development organisations for which the power imbalance is lowest. It is therefore in the interest of a development organisation to take actions to lower the power advantage it may have over the client. One way to do so is to materialise the interest that the development organisation has in business value of its client. For instance, to become a stakeholder in one or more goals that are under control of the less powerful organisation (the client) [8].

Take for example a program that works as intended by the programmers (i.e., has a high quality from their perspective), but does not match the environment in which it will be embedded (i.e., low quality from the perspective of the client). Hence, we have an unhappy client. At this point in time, the software development organisation usually has great power over the client, as it is for example too late to switch to another vendor. The software development organisation can exploit this power imbalance [8] and insist that it did its job and sees no reason to invest more work in the product. The client organisation may have foreseen such opportunistic behaviour and may have created safeguards in the contract (e.g., limited upfront payments). However, no quality improvement whatsoever is obtained if the software development organisation takes this road, and this is counter-productive given that the software development organisation usually also has objectives that surpass the project [15]. What is needed instead is a strategy that does result in quality improvement: the communication facilitator strategy.

3.3 The conceptual communication facilitator

All in all, the existence of different definitions of communication facilitator (or mention of this concept at a very abstract level) calls for aggregating theory into a conceptual definition that can actually be of use for outsourcing practitioners. To solve that problem we signalise the communication facilitator, i.e., the conceptual communication facilitator, as a role and we define it as follows:

A communication facilitator is a role in the software development organisation that has the responsibility and authorisation to ensure that there is fluent communication with the client organisation to address issues that arise during software development.

The communication facilitator role interacts with several different groups of stakeholders, as depicted in Figure 1. In this figure, the communication facilitator role is in the center, as an interface between the client (left-hand side) and different groups of persons in the development organisation (right-hand side), such as project leaders, architects or lead developers, software designers, programmers and testers.

4 Communication facilitator in practice

We observed the communication facilitator role in three instances of outsourcing relations in two Dutch software development companies, which we will call Medium Company and Big Company. These particular instances are neither toy examples nor hypothetical scenarios. Rather, they refer to real outsourcing cases that we have found in practice.

In the Netherlands, with several thousands of employees, Big Company is in the top-ten of IT service providers ranked to number of employees and revenue. The company offers a wide portfolio of services, such as IT infrastructure management and consulting. Medium Company is a medium-sized (100-200 employees) developer of embedded systems solutions and e-business systems in the Dutch market.

The cases in our examples complement each other describing typical outsourcing scenarios. They are representative because the services these companies offer are similar to the services offered by the rest of these kind of companies in the market.

Figure 1. The conceptual communication facilitator.
4.1 Case description

Case 1: a domestic outsourcing scenario. Our first example focuses on Medium Company. Table 1 shows the data of the projects investigated [5]. The size of the project (i.e., large, medium or small) is relative to the company size. The company performs, in The Netherlands, software development projects for mostly Dutch clients (domestic outsourcing).

<table>
<thead>
<tr>
<th>Project</th>
<th>Project size</th>
<th>Team size</th>
<th>Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>large</td>
<td>10</td>
<td>1 M euro</td>
</tr>
<tr>
<td>B</td>
<td>medium</td>
<td>8</td>
<td>0.5 M euro</td>
</tr>
<tr>
<td>C</td>
<td>small</td>
<td>3</td>
<td>0.2 M euro</td>
</tr>
</tbody>
</table>

Table 1. Medium Company’s studied projects [5]

Case 2: a multi-national outsourcing and offshoring scenario. Our second example focuses on Big Company, a large (several tens of thousands of employees) multinational IT service provider, with offices in many countries. In this example, we zoom in on the relation between the domestic (on-site) team and the offshore team (both teams belong to Big Company). This is thus an example of offshore in-house development of software that Big Company develops for its customers (outsourcing). In this example, we focus on the software development services provided. Contrary to the previous case, Big Company uses geographically distributed teams to carry out the software development projects.

<table>
<thead>
<tr>
<th>Project</th>
<th>Size</th>
<th>Team size</th>
<th>On-site</th>
<th>Off-shore</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>large</td>
<td>40</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>B1</td>
<td>large</td>
<td>24</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>B2</td>
<td>large</td>
<td>22</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>C</td>
<td>small</td>
<td>10</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2. Big Company’s studied projects [11]

4.2 Results

We have found that a communication facilitator was used informally in all the projects. The person in that role had been given, perhaps informally, the power that the situation demanded. Therefore, we conclude that our results confirm existing theory associated with the conceptual communication facilitator.

Our intention was also to learn about how we can apply the communication facilitator strategy to improve the client’s business value. At Medium Company we found that the role of communication facilitator (CF) was enacted by the project leader. The project leader was the informal CF, spending as many resources as needed to be closer to the client. However, that came at a price. Provider’s team members cannot expect to have the project leader available at any time to answer their questions. The project team was adapted to compensate the fact that the project leader had to be away frequently. In particular, the team had an extra member, named the lead developer, to answer developer’s questions faster whenever they occurred.

Transfer of information between Big Company’s on-site and offshore teams was managed by informal CFs. Team members enacted the role of CF informally. In one project the project leader was the CF, while in the other two projects, the CF role was performed by developers. The particular outsourcing conditions of Big Company’s projects made them to have two CFs: one on the on-site team that was acting as a client and another on the offshore team that was acting as a provider.

The typical problems that were mentioned in the interviews (such as conflicting and fluctuating requirements, need for re-negotiation with the client, lack of usability or requirement understanding from development staff and tensions between two contending groups) have been detected long ago [3] and continue to appear. In the discussion of the practical implications that follows we focussed on finding out what is it that keeps developers and managers from acting on those factors that, and this is something they already know, lower the quality of software.

5 Practical implications

Practical implication 1: Software development team structure. Figure 2 depicts the communication facilitator in the scenario of the projects observed in Medium Company. The structure of the software development team accommodates itself to match the use of the communication facilitator strategy. In order to give the project leader enough freedom—they need to be close to the client—a new figure appears: the lead developer. The structure is as suggested by current software development methods such as DSDM [6].
Practical implication 2: the communication facilitator role needs managerial support for its freedom. In Medium Company, project leaders (who have the communication facilitator role) have complete authority to invest time in the client, and management fully supports this. In particular, on the one side, to give extra freedom to the project leader for her communication facilitator role, there is a lead developer, who stays all the time with the developers, answering (for instance) their questions about requirements and product usability. On the other side, in the event that the project leader detects the need for a change, they communicate it to the lead developer (as we have noted above, the team structure is adapted accordingly to include this member), who will take care that the team implements it.

Practical implication 3: project leader role needs authority and managerial support for their decisions. In Medium Company, the project leader role, not the communication facilitator role, is responsible for the team. The project leader of the development team is empowered to add and remove people to their project ("hire and fire"). This is necessary when using the communication facilitator strategy. If the project leader is not sufficiently empowered, the communication facilitator will find their hands tied in negotiation with the client. The client will soon feel that client-provider communication, while open and well established, in the end does not have any consequences. The project leaders add or removes people from their teams if the project is more complex than expected, or if a particular developer turns out not to fit in the team. It is also important that they know that they have managerial support for their decisions.

Practical implication 4: Better understanding of the deployment environment. Having an idea of the way the software product will be used (e.g., building prototypes involving the client), helps developers to make the right assumptions to compensate for the tacit requirements; which potentially gives software with higher usability [13]. This suggests that the communication facilitator strategy, facilitating communication, might improve in the long term the quality of the software. Further empirical research can confirm or deny this.

Practical implication 5: no fear of requirement change. The projects were carried out in a surprisingly stress-free environment. It is not the case that the projects were problem-free. Private communication between the authors and the experts of Medium Company revealed common but specific problems in the software development process, such as fluctuating and conflicting requirements, but Medium Company’s project leaders did not complain of changing or conflicting requirements. Rather, they trusted their people’s ability to solve problems. Borrowing from Curtis et al. we can say that they simply “accommodate to change as an ordinary process” [3]. Moreover, they sought
It is not surprising, then, that the projects of the outsourcing team reacted fast to everyday problems. They used modern coordination practices [10]. The project leader (who has the communication facilitator role) knew that they could anticipate change if they kept communication between two contending groups (client and provider developers) then to fear requirement change.

**Practical implication 6: client return (client re-buy)** In Medium Company the responsible person of the project in the client team felt reassured, both teams (client and provider) knew that they could anticipate change if they used modern coordination practices [10]. The project leader of the outsourcing team reacted fast to everyday problems. It is not surprising, then, that the projects of Medium Company ended on time and within budget. Sometimes they had problems, e.g., a developer not performing well. However, all the issues were resolved by Medium Company experts surprisingly fast. In general, clients appreciated this quick reaction to everyday problems.

All in all, we interpret our observations in Medium Company and Big Company as follows. We think that the communication facilitator strategy explains in part their success. However, further research is needed to determine how the influence of the communication facilitator on project success compares to the influence of traditional aspects of software engineering, such as the use of a particular programming language, a particular management process or the latest notation. Other implications for further research are presented in the next section.

### 6 Research implications

Apart from practical implications, we also found questions that still have to be answered by further research. In this section, we present such questions, as well as a further elaboration of the communication facilitator concept that our findings suggest. These questions fit in a more general program of empirical validation of the many factors that influence outsourcing and the quality of the developed software. The overall aim is to find pragmatic guidelines that help practitioners to produce software of higher quality (in terms of the client’s business value) without constraining the particular way of working of partitioners or even of organizations.

We identify two questions that call for a more elaborate conceptualisation of the communication facilitator. First, we note that the communication facilitator is a role, not a person. However, this role has to be assigned to a person. The question is then whether it is preferable to combine the roles of communication facilitator and project leader in one person, or to appoint a dedicated communication facilitator, *i.e.*, employ a person whose sole role is the communication facilitator role. Second, regardless of the answer to the previous question, what should be the exact responsibilities and authorizations of the communication facilitator, how does this relate to the responsibilities of other roles in the software development organisation, and what is the communication facilitator’s place in a software development team?

Which choices to take depends on properties of the software development project at hand. For instance, in making these choices, the responsibilities associated with the communication facilitator role and other roles, especially the project leader role, have to be balanced, whether these roles are combined in one person or not.

In Figure 4, we further specify the role of the communication facilitator and contrast this to the project leader role by presenting responsibilities and authorizations of both roles. The question to be answered by further research is whether these responsibilities and authorizations indeed appear in practice. The specific responsibilities and authorizations that we suggest are:

- **communication facilitator responsibility**: The communication facilitator role has to focus on client satisfaction. That is, the communication facilitator’s sole concern is the question “Are we doing the right things?” [9] (for our client) and the communication facilitator could help the client to determine what is really needed.

- **communication facilitator responsibility**: Requirements change over time. The communication facilitator role has to detect requirement changes that are needed and make sure that they are addressed properly by the development organisation.

- **communication facilitator authorization**: The communication facilitator role should be fully empowered to speak (face to face) with any person both in the client organisation as well as in the developer organisation.

- **Project leader responsibility**: The project leader is responsible for finishing the project on time and within budget, and for managing project risk.

- **Project leader authorization**: The project leader role is authorized to make decisions with respect to how the project budget is spent, to hire and fire project team members, and to which artifacts are created during the software development process.

More generally, software developing organisations cannot afford not to optimize their software development practice. This has become evident with the emergence of many frameworks and methodologies (*e.g.*, TQM, Six Sigma,
Business Process Management, CMMI) and management process models (e.g., requirement change management process models [17]). However, the profusion of methods generates considerable confusion for practitioners, as to which is the best method in a given scenario and therefore research should provide adequate support for dealing with these challenges.

7 Validity

7.1 Internal validity

Internal validity concerns the question whether the research method chosen leads to the right conclusions in the case studies. To address this concern, we use feedback from two experts. One of the experts, who has nine years of experience in outsourcing, is a member of one of the case study organizations, but he is independent from the projects studied. He assessed internal validity as high. He indicated that the benefits of the communication facilitator approach suggested by this study can be recognized in practice. According to the expert, applying the communication facilitator strategy improved information and knowledge exchange between client and development team members. Moreover, it improved the transfer of tacit knowledge. This is very important, as it reassures the client that the project is under control. Ultimately, it is the client who is responsible for success of the project in his own organisation, whether or not part of this responsibility has been delegated to the provider.

The other expert has an academic background in software engineering. He generally agrees with the approach taken at this stage (exploratory), but suggests that further research is needed to exclude any competing explanation of our findings.

7.2 External validity

External validity concerns the question whether results can be generalized to other cases. In this paper, we confirm existing theory about outsourcing and software development in the scope of two current outsourcing scenarios (with a few projects each) in two organizations. By showing that the phenomena predicted by theory do indeed appear, the validity question boils down to the question whether these phenomena would also appear in additional, similar scenarios. Based on discussions with experts (see the previous subsection), we conclude that this seems to be the case: there is nothing special about the scenarios studied. To the contrary, both organizations base their software development processes on common standard methods such as DSDM [6] and CMMI [2]. One of the experts noted that both methods in fact strongly suggest to apply the communication facilitator strategy, without using this term.\(^2\)

8 Conclusion

With the advent of global outsourcing, a new premium is being placed on those IT solution providers that are able to respond to the real needs of the client, operating under conditions of uncertainty and rapidly changing requirements.

However, knowledge about how do IT solution providers manage to accomplish their work remains relatively lim-

\(^2\)In CMMI 1.2, this suggestion is no longer present in the core of the CMMI. Instead, it has moved to the acquisition module.
ited. In particular, little empirical research has focused on how software development teams organise themselves to facilitate adaptability, react speedily to change, and program the software that the client actually needs in the outsourcing context. This work aims to shed light on how software development teams adapt to maximise communication between client and providers in outsourcing projects.

We have observed projects from two Dutch IT companies to find that they take actions to ensure the flow of important information between people from the client and the development team. These projects make one team member act implicitly as a communication facilitator. In some projects, we have found this role performed by a project leader and in other projects, by a developer. The observations suggest that it is important that the communication facilitator role is equipped with the right authorisations, and that there is a good balance of responsibilities and authorities of the communication facilitator role and those of the project leader.

The observations suggest potential concrete benefits of using the communication facilitator strategy. For instance, the arrangement favours that teams applying the communication facilitator strategy may become potentially more flexible toward unanticipated challenges.

We are very positive (but we leave collecting evidence confirming that for future work) about the potential of the communication facilitator strategy when scaled up to the organisational level, i.e., having a whole organisational unit devoted to ensuring that the needs of the client’s businesses are met, rather than leaving that challenge to the effort of individual employees.

ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support of the Dutch Jacquard program for the project “QuadREAD”. Thanks to the two experts that confirmed our findings, helping us to validate the results presented in this paper. We are also grateful for the helpful comments we received from the reviewers of this paper.

References