

Using Property Rights Theory to Overcome Success Barriers to Software Development Project: Protection of Contractors' Knowledge

Cornelia Gaebert

*Research Group on Strategic Information Management, European Research Center for Information Systems,
c/o University of Muenster, Leonardo Campus 11, D-48149, Muenster, Germany*

Keywords: Software Development Project, Information, Knowledge, Intellectual Property Rights, Property Rights Theory.

Abstract: A fundamental tenet of the information systems discipline holds that: (a) changing requirements in software development projects (SDP) are the main reason for failure; (b) therefore, in case of such uncertainties, fixed-price contracts (FPC) are not suitable for success. Our research, informed by economic theories, compellingly illustrates that among other things changing requirements stems from missing protection on knowledge. In this paper, we present an analysis of knowledge difficult to protect. Both parties must share knowledge during the specification of requirements. However, this business knowledge is an essential intellectual property, and thus needs protection for misuse. We enact a strategy to achieve SDPs success despite these barriers. Our theoretical and empirical analysis also found that SDP success is largely an uncertainty problem between the contractors on the management level, and thus technical-organizational approaches alone are inadequate for achieving success. Based on property rights theory, we introduce two models for protecting knowledge depending on uncertainties. Our findings offer managers important insights into how they can design and enact FPC for effectively manage SDPs. Further, we show how the economic theories can enhance understanding of SDP dynamics and advance the development of a theory of effective control of SDP success.

1 INTRODUCTION

Despite project management improvements and professionalization of the software development process, the number of failing software development projects (SDP) has remained high for decades (Standish Group, 2010; El Emam and Koru, 2008; Dijkstra, 1972). Changing requirements are considered as the main reason for failure (cf. Al-Ahmad et al., 2009; Dwivedi et al., 2013). Organizations expect to mitigate this risk by outsourcing (Chua et al., 2012). They expect the supplier to take the risk for the project failing when working independently. Researchers claim that in this situation a predetermined price for the software system - defined in a fixed-price contract (FPC) - is not suitable for completion of the SDP in line with the expectations of the contractors (Dey et al. 2010, Fink, and Lichtenstein 2014). Indeed, the contract influences the success or failure of SDPs (Chen, and Bharadwaj 2009). This is not a question of trust, but an economic issue of uncertainty due to incomplete information on both sides (Williamson, 1985).

In this context, we claim that ineffective contract provisions regarding the protection of property rights (PR) on information are at the root of causes for ineffective FPCs. We claim that among others the inadequate protection of information causes changing requirements. We argue that to be successful in software development outsourcing (SDO) the contractors must share business information and knowledge (I&K) at least if the project regards a novelty (Tiwana, 2004). However, this I&K belongs to the organizations' intellectual properties (Teece, 2000). Therefore, both sides have an interest in protecting their I&K for securing their intellectual PR (cf. Norman, 2002). Problems arise if the SDP contract does not contain effective instruments securing these PR.

The available rights on the exchange object are in the focus of the PR theory, a central approach of new institutional economics (Hart and Moore, 1990; Cole and Grossmann, 2002). The research focuses on the institutional framework and the incentives triggered by regulatory and ownership; the use of economic theories proved to be appropriate (cf. Benaroch et al.,

2012; Aubert et al. 2003; Lichtenstein, 2004; Beulen and Ribbers, 2003). Methodically, the aim is to provide a theoretical ex-ante investigation of the contract impact on reaching customer and supplier-specific economic outcome goals.

In this paper, we will investigate how the PR theory can improve our understanding of how the right protection of PR can attenuate reasons for requirement change and therefore, reasons for failure of SDPs. However, there is a paucity of research on how contractors can improve SDP performance or outcome by protecting PR on information under FPC.

Therefore, we have conducted an empirical investigation regarding the role of sharing I&K in SDPs. The collected empirical data will guide our theoretical considerations. After a brief description of our empirical study in section 2, we will discuss the kind of knowledge and information customers and suppliers must share and protect within the project in section 3. We will distinguish this from other kinds of information needed to share between the parties without the problem of protecting it from misuse. In addition, we will tell it apart from intellectual PR.

The conflict description between the need of sharing the knowledge and the necessity of protecting it by means of the PR theory is in the focus of section 4. We will develop a rent and a wage model concerning the PR on I&K in SDPs under FPC, depending on the net gain of the cooperation.

In section 5 we will finally discuss possible decision criteria regarding the derived. Above all, we will suggest instruments for securing PR on information while sharing it during the project phases.

2 THE EMPIRICAL SUPPORT

Throughout this paper, we will be using an abductive research approach (Osei-Bryson and Ngwenyama, 2011; Schurz, 2008). We are “in the discovery stage of scientific hypothesis formation and testing” (Walton, 2014). For the collection of empirical data on the I&K role in SDPs under FPC, we conducted a two-step evaluation. First, we developed a questionnaire in the form of a standardized online survey as a special kind of standardized survey (Klammer, 2005). Next, we conducted personal interviews to deepen our understanding of the questionnaire results. The evaluation period lasted one year.

For the questionnaire, we chose the standardized online survey to give the respondents an opportunity to reflect and to question their own companies

(Schnell et al., 2011). The format of the online survey itself was legitimate because the interviewees were an IT-savvy group. Open answers supplemented closed questions so the questionnaire would not be too restrictive. Also, they helped gather the covered information (Mayer, 2012). In the following, we will analyze and interpret the results descriptively.

Experienced project participants on both sides (customer and supplier side) were interviewed. The questionnaire had to take into account the management perspective as well as the project management’s view. Since it is not possible to trivially address the population of all SDP customers and suppliers, and given that questioning the population about any associated and unacceptably high cost is not realistic, we chose a smaller population. Therefore, we could not achieve complete representativeness (Schnell et al., 2011). For practical reasons, we addressed the 45 members of an IT companies network in Germany. Fifty additional addressees were available from other contacts. To expand the circle of respondents and to amplify the customer side, we used contacts in social networks such as Facebook (approximately 30), Xing (approximately 20), and Twitter (approximately 50). This ensured that the respondents had experience in different possible project contexts. Of the 200 addressees requested to participate in the survey, 29 actually completed the questionnaire (14 suppliers, 5 customers, 9 suppliers *and* customers, and 1 other).

An independent survey evaluating the willingness to participate in the survey suggested a conscientious answering of the questions. A total of 48.3% of the respondents indicated that they belong to management and have contract responsibility; 27.6% are project managers; 6.9% are employees at the working level, and 17.2% perform other activities, such as consulting. A total of 89.7% of the respondents had 10 or more years of SDP experience. The participants represented a broad range of project sizes with regard to duration and number of employees.

For the exemplary and in-depth interviews, we conducted semi-structured expert interviews. On the one side, we questioned a consultant with an SDP experience of approximately 15 years. He supports big companies in defining and organizing the contractual issues of SDPs. On the other side, we spoke with a supplier having an SDP experience of approximately 20 years. He is the owner of a software development company employing 10 programmers. Considering the sensitivity of failure research and the resulting difficulty in gaining access to project details, this methodology was most appropriate. The

incomplete script of the semi-structured interview format left room for improvised questions (Myers and Newman, 2007). The first interview lasted approximately 3 hours; the second one, 1.5 hours. We made extensive notes during the interviews, which we evaluated afterward through a qualitative content analysis. Since we demanded appointed circumstances and facts, we were able to avoid free interpretation problems (Gläser and Laudel, 2009).

In the next sections, we will present some study results in connection with the corresponding theoretical considerations.

3 KNOWLEDGE AND INFORMATION IN SOFTWARE DEVELOPMENT PROJECTS

The requirement specification is a fundamental document for the contract and the SDP itself, and it consists of I&K. We claim that not all I&K are worthy of protection. Therefore, we will analyze I&K in this section. First, we will discuss the difference between knowledge and information. Second, we will detect which kinds of I&K are worthy of protection in the context of SDPs. Finally, we will summarize the chain of reasoning derived from that.

3.1 Knowledge and Information

It is difficult to distinguish between *knowledge* and *information* in a way that it can be clearly answered, for every single situation, if some set of statements, documents, or data contains knowledge or information (Rowley, 2007). For systematic reasons, we will clearly differentiate between information and knowledge as follows.

According to Ackoff (1989), an information contains useful data and provides answers to questions like *who*, *where*, *when*, and *what*, whereas knowledge is an application of information, which answers *how* questions. In addition, Zeleny (1987) describes information as knowing *what*, whereas knowledge is knowing *how*. One can describe a business process by saying *who* is the process actor, *what* the actor does within this process, and in which situation (*where* and *when*) he does it. These descriptions are information about the business process. On the other hand, one can describe *how* the actor meets a business goal by carrying out a business process. This provides knowledge, the know *how* to reach the goal. In order to have knowledge it is therefore not enough to be able to describe process

details, but you must also know the goal, for which the process is a tool on the way to reach it.

Both Ackoff and Zeleny point out further kinds of knowing something. If one not only knows *what* (*where*, *when*, *who*) and *how*, but also has the answer to the *why* question, then, according to Ackoff, the person has got *understanding*. Zeleny calls the ability to provide answers to *why* questions *wisdom*. As knowledge is inherited from information, understanding and wisdom are inherited from knowledge. Thus data is the quasi common denominator of information, knowledge, understanding and wisdom.

There is a broad discussion going on regarding the differences between information, knowledge, understanding, and wisdom (Rowley, 2007; Swigon, 2013). It is not the supplier's task to understand the customer's business model in the broadest sense. However, there is no clear divide between knowledge and understanding. The purpose of a software system is in some sense connected to the customer's business goal. For our purposes, it is therefore sufficient to clearly mark the difference between information and knowledge. For that reason, we will use the notion of information for data regarding processes, events, objects, and methods without answering how these described data are used to reach a goal, and why these data are necessary as a means for someone's purpose. On the other hand, knowledge offers us such answers. He who has knowledge knows the purposes the data are used for. He knows how these data should be used to reach these purposes, and why these data are suitable for the purposes.

In the next section we will consider I&K in an SDP context.

3.2 Knowledge and Information in Software Development Projects

We aim to clarify which kind of I&K is a property the contractors wish to protect against misuse by the other side. During SDP preparation and execution, the customer and the supplier must share different kinds of I&K. Our empirical study already shows that during the proposal phase and during contract negotiations both sides must provide detailed information on the customer's requirements and the supplier's abilities.

For analyzing purposes, we will first take the customer's perspective. There is broad research in the field of software engineering regarding I&K provided by the customer in SDPs. After that, we will be able to develop a consistent view of the supplier's perspective. From this we will summarize which are

the kinds of I&K worthy of protection in the context of an SDP.

3.2.1 The Customer’s Perspective: Requirements

The research about customer-delivered information focuses on requirements representation and specification (Pohl, 2013). Researchers have been distinguishing for long three kinds of requirements: project requirements (a), process requirements (b), and system requirements (c) (IEEE, 1998; Glinz, 2007).

(a) We have to consider information on the project’s timeline, the budget, and milestones to be met (what, when, where, and who). These are the project requirements. Obviously, the customer must share this information, because the supplier will set up the project plan based on it. In our expert interview, the business side consultant stated that the customer shares this information in the early phases, and mostly long before signing the contract. Often the customer delivers such information to more than one potential supplier. Obviously, there is no problem as long as it is only information as defined in the previous section. However, problems arise during the project if it gets impossible to achieve the needed milestones. Due to external reasons, meeting a milestone is sometimes prioritized, but in other cases, it may be possible to shift the timeline if changes in requirements or other problems arise. To find the right decisions, the customer must share more than information with the supplier. It is necessary to deliver the answer to the question why a milestone was set to a special due date. According to our definition in the previous section, this is part of the customer’s knowledge. This knowledge may be connected with other knowledge from the customer’s side, for instance a marketing strategy for a new product launch. There is good reason to hide this knowledge from external suppliers.

(b) There is information needed directly for work processing and for the coordination of the involved

staff’s activities. From the customer’s perspective, such information is part of process requirements. Due dates, contact persons’ names and data as well as the meeting schedule are examples of such information, which controls the process. This kind of information is not in the focus of our study. There is no reason to hide such information, because it is only valuable during the project and in case the contractors share it. (c) When we use the notion of requirements during SDPs, we refer to system requirements. Requirement specification is mostly the first part of each project. Since changing and ambiguous requirements are one of the main reasons for project failure (cf. Introduction and our empirical investigation), researchers have been focusing on methods of software requirements engineering during the last decades.

System requirements are divided into functional and non-functional requirements, and constraints. Functional requirements define what the system is bound to do. Obviously, this information is crucial for producing the right software system. On the other hand, the customer is often not able to define in a clear and detailed way how the system should look like, how it should react to user input, or how it should process data. As the manager from the supplier’s side stated in our interview, in order to design an appropriate system it is useful to share knowledge about how the users do their work, about what the business processes are, and why they need support from the required software system. This knowledge is the know-how of the customer’s organization. There are good reasons for sharing this knowledge with the supplier, but on the other hand there are also reasons to hide this know-how: it is the internal knowledge of an organization and it is its competitive advantage.

The customer may also define non-functional requirements as information. Requirement specifications must contain statements about needed response times or data volumes. Sometimes it is necessary to prioritize non-functional requirements. For this, it is useful to know why the customer needs the defined system parameters, and why it is

Table 1: Kinds of information and knowledge provided by the customer.

| | Information | Knowledge | Protection-relevant |
|----------------------|---|---|--|
| Project Requirements | Project’s due dates | Project Business Goals | Relevant, maybe interesting for competitors |
| Process Requirements | Project internal due dates Availability of resources | Current business processes and activities outside the project | Not relevant, because only valid during a short time period |
| System Requirements | Functional and Non-functional Requirements, Constraints | Business Goals, Business Knowledge | Relevant, interesting for competitors, usable by the supplier in other projects, define the customer’s competitive advantage |

important to meet these requirements. Then, it is not enough to deliver information, but it is also useful to deliver knowledge about the business know-how.

Last, we have to consider constraints. Constraints are legal or cultural restrictions which the supplier must take into consideration during the software system development. Such information is mostly publicly available, and is not the customer's private property. Therefore, we must not examine it in detail in this study.

Table 1 shows an overview of the I&K to be provided and shared by the customer during an SDP. The customer may share nearly each input either as information or as knowledge. At the knowledge level, the customer may have an interest in protecting it from usage by external parties (Norman, 2002). Therefore, we can depict this knowledge as customer property. Properties are worthy of protection. We will further develop this idea and its consequences in section 4 of this paper. Before, we will take the supplier's perspective.

3.2.2 The Supplier's Perspective: Abilities

In order to achieve a suitable and systematic description of I&K provided and shared by the supplier, we will describe this in analogy to the kinds of I&K found for the customer. In the previous section, we have identified three kinds of I&K for the customer to deliver during the SDP. These are (a) project requirements: organizational project embedding within the organization's processes, its connections with customer strategies and business goals, (b) process requirements: project management and organization, (c) system requirements: the organization's business know-how as far as it is relevant for the development of the needed software system.

On the suppliers' side, we can identify three corresponding kinds of I&K. (a) Project requirements: The supplier has to embed the single project within his production processes. He must answer the following questions: When is the needed staff available? Which experts are necessary? When will they be able to work on the project? Which other

resources, like development and test systems, are needed for the project?

This information influences the overall project plan. The supplier must deliver these data to the customer by at least partly committing to timelines and milestones. Sometimes, the sequence of the project's working steps will depend on such external basic conditions. If the supplier does not just deliver the plain information, but also the reasons underlying the decisions for a certain setting, the supplier shares organizational knowledge with the customer. This knowledge concerns the supplier's organization ability to handle and process projects as needed in order to produce the desired software system.

(b) Process requirements: As in the customers' case, sharing project management information is not worthy of protection because it has no value outside the project. This information is in no way connected with the organization's knowledge. Therefore, we will not consider this information in our study.

(c) System requirements: We must consider information regarding the core of the supplier's business expertise. During the proposal and negotiation phase, the supplier must deliver information to show that the organization is able to produce the needed software system. Therefore, the supplier shares relevant parts of the business expertise. Furthermore, during system development of the system, it may be useful to share information on the used employed tools and methods. In addition, the supplier has to share technical implementation details to justify prioritizing the implementation of non-functional requirements. Often, the customer and the supplier must make such decisions in common. In order to make possible such decisions, the supplier must share the information in a way that the customer can generate knowledge regarding the underlying technical facts, methods, and constraints. This knowledge is worthy of protection for the supplier, because the generation of such knowledge makes the customer more independent from the supplier's services (Gassmann et. al, 2010). Furthermore, the customer may use this knowledge in other projects and may share it with the supplier's competitors

Table 2: Supplier's knowledge and information.

| | Information | Knowledge | Protection-relevant |
|-------------------|---|---|---|
| Project Abilities | Commitment of Project's due dates | Supplier's overall strategy | Relevant, maybe interesting for competitors |
| Process Abilities | Project-internal due dates Availability of resources | Current business processes and activities outside the project | Not relevant, because only valid during a short time period |
| System Abilities | Development methods, Used tools and frameworks | Know-how in development, experiences, problem- solving strategies | Relevant, interesting for competitors, make the customer independent, define the supplier's competitive advantage |

(Paasi et al., 2010). We have summarized the supplier's perspective on I&K he has to share in Table 2.

3.2.3 The Character of Information and Knowledge as Properties

Summarizing the theoretical analysis of I&K needed in SDPs, we have to accept the fact that both parties have reason to protect knowledge from misuse by the other side. There is knowledge regarding the internal business processes, regulations, experiences, and goals making the organization powerful and successful in the market and in the context of competition with other firms. These are intellectual properties of the firm (Teece, 2000).

Consequently, this knowledge is an important asset for the firm as long as the organization protects it against competitors. The customer possibly maintains a relationship with the supplier's competitors and vice versa. Therefore, both contractors have reason to protect their own knowledge. Both contractors may withhold the needed I&K as long as possible.

On the other hand, sharing this knowledge is crucial for a successful project. Knowledge is needed to make the right decisions in designing the right system. However, at the beginning of an SDP both parties do not know the exact I&K is needed. Therefore, both parties deliver I&K as late as possible, a reason for changing requirements.

4 THE PROTECTION OF KNOWLEDGE: THE PROPERTY RIGHTS MODEL FOR THE SDP

In this section, we will provide the announced theoretical ex-ante investigation of the contract impact on reaching customer and supplier-specific economic outcome goals in an SDO project. We will investigate how PR theory can improve our understanding of how the right protection of PR can attenuate reasons for changing requirements and therefore, reasons for failure.

First, we will give a summary of our literature review on efforts made to date. We claim that the protection of knowledge needs further research; this paper will contribute to that. Second, we are adopting a classical model from literature on our situation under investigation. We will use Barzel's (1997) landowner model.

4.1 Excursus: Protection of Knowledge by Credible Commitments

Copyright or patent laws are formal protection measures, yet they are not suitable for protecting knowledge in an SDP context (Liebeskind, 1996; Friesike, 2011). As shown by Liebeskind (1996), the firm itself has the power to protect the knowledge, making it invisible from the outside. Friesike (2011) calls these options "informal protection measures". However, contractors have to share knowledge in an SDP. Consequently, preventing visibility is not useful.

In long-term collaborations, contractors can reach trustworthy behavior by credible commitments (Williamson, 1983; North, 1990; Ebers and Sen, 2015). For that, reciprocal specific investments are most effective. Reciprocal specific investments create a mutual hostage situation that serves as a safeguard against the misuse of knowledge. However, these investments are not suitable for a single SDP. This also applies to non-disclosure agreements, they are effectively reachable only in long-term collaborations (Craswell, 2006; Bogers, 2011). Thus far, the contractors are aware that the SPD's end is within the range of vision. Therefore, trustworthy behavior is hardly to ensure, especially after the project's end.

Consequently, we must search for alternative incentives to secure knowledge protection during an SDP. In the next section, we will find such options by starting from models of the economic PR theory.

4.2 Protection of Knowledge by Property Rights: The Landowner Model

A classic model introduced by Barzel (1997) analyzes the contractual situation between a landowner having PR on land and a worker having PR on labor. In order to produce goods, at least one of them must assign the PR to the other. They have three options: first, the landowner may assign the land-using right to the worker through a fixed-rent contract. Second, the worker may assign the labor-using right to the landowner via a wage contract. Finally, they can sign a shared tenancy contract to share the profit from the crop sale. In each case, both parties will overuse the contractual partner's resources, which they control themselves. On the contrary, each party will reduce its support for resources, which are now under control of the other side. Therefore, the decision for a contract type depends on the net gain of the cooperation. No contract type will always be the best under all

circumstances. With a fixed-rent contract, the landowner will maintain his land less improved; with a wage contract, the tenant will shirk more. Half way between both of them, there is a bit of everything. In this case, output specification and monitoring will be additionally necessary.

I&K does not need maintenance in the sense of physical wear and tear. At first sight, I&K cannot be overused, because it cannot be damaged by using and sharing. Nevertheless, we can interpret it as an overuse when a party, having received I&K from the other side, shares this with a third party or uses it in another project. In such cases, I&K as a property of the owner may lose its value.

The fixed-price contract, as under our investigation, is the most commonly chosen contract design (Badenfelt, 2011) in SDPs; our empirical investigation supports this. However, we must be careful: using the fixed-rent model for managing the usage of I&K in an SDP does not imply a FPC for the SDP, such as a wage model does not imply a time and material contract. We will see as follows that the crucial question does not relate to the price model of the underlying contract. It is the question who controls the usage of I&K and labor.

However, with respect to SDPs, the landowner model has some weaknesses. First, the I&K needed by the other party plays the role of the land. This raises the question whether someone is willing to pay for the use of a requirement specification. Second, in our case landowner and crop buyer are the same party. Consequently, no independent third party (the market) will validate the value of the I&K. Incidentally, this explains why shared contracts are unusual in SDPs.

Therefore, we will analyze the consequences of these problems in the following, and discuss the fixed-rent model as well as the wage model.

4.3 The Fixed-rent Model

In the most obvious application of Barzel's model to the situation of an SDP, the customer plays the role of the landowner, whereas the supplier is the worker. Requirement specification and the business knowledge behind it are customer properties. In case of a rent model, the supplier rents this knowledge. After finalizing software development, the supplier sells the software solution to the customer.

However, according to section 3, we must consider the opposite situation, too. We will look for situations in the SDP, when the customer uses supplier's knowledge to contribute to the project success.

In the next section, we discuss the following questions: is it possible to rent the knowledge needed by the other party inside the SDP? How can we interpret the model in this situation, and what are the consequences of that? In the following two sections, we will analyze both cases where supplier or customer rent knowledge.

4.3.1 Knowledge regarding Requirements

At first glance, it seems to be an unusual idea for a supplier in an SDP to rent knowledge from the customer. Two questions arise: (a) Should the supplier pay a rent for the knowledge, which is necessary for the SDP? (b) How it is possible to give the knowledge back to the owner after finalizing the project?

Concerning the first question (a), we can take the view that the parties offset the knowledge rent against the price for the software solution delivered at the end of the project. Customer information on business goals and the business knowledge lying behind this information is valuable for the supplier. This knowledge helps the supplier in understanding and thus developing the required software. Consequently, customer knowledge is valuable for the supplier, and the supplier can rent this knowledge. The contractors can offset this rent with the calculated overall SDP fixed-price.

Nonetheless, we must discuss the possibility of project failure. The supplier does not deliver any software and the customer will not pay the agreed price. Should the contractors stipulate a penalty regarding the delivered knowledge? We are facing the same question as in the landlord-worker model: should the worker pay a rent after crop failure, if he cannot harvest grain? The answer the rent model gives us is yes, because the landowner does not know the reasons for the failure. Therefore, it is the worker's risk. Dispensing the rent would be only possible in the case of a sharing model. Thus, also in an SDP, the parties should agree on a penalty in case the customer shares business knowledge, but the supplier delivers no software system.

We are now coming to the more interesting second question (b). Here, the parties must avoid that the supplier integrates the knowledge rented from the customer into the knowledge of his own organization. As our empirical study shows, 77% of the suppliers admit using received information outside the project. If the knowledge given from customer to supplier becomes a part of the suppliers' business knowledge, the supplier cannot give back this knowledge when the rent comes to an end. We can interpret this as an

overuse of information as defined by Barzel (1997). If the supplier uses the customer's business knowledge in other projects, namely to support competitors of the customer, the value of this knowledge decreases, as does the value of overused land.

One can argue that the supplier must integrate knowledge into his own knowledge in order to develop the required software. Nevertheless, we must distinguish three cases: integration of knowledge into the knowledge of the supplier as a knowing organization (i) (Choo, 1996), integration of knowledge into the knowledge of a project team (ii), and integration of knowledge into the knowledge of a single person (iii), for instance the requirements engineer or solution designer. (i) For software system development, it is not necessary for the supplier to integrate knowledge about requirements into the knowledge of his own organization. The contract should contain rules to avoid this. (ii) In our empirical online survey, 77% of the interview partners say that such knowledge will be discussed in joint project teams. The project team knowledge disappears when the project ends. (iii) Last, a person's knowledge about complex facts often requires this person to have access to detailed documentation. We can interpret this documentation as the expert's extended mind (Clark and Chalmers, 1998). For proper use of the knowledge, the expert needs access to the documentation. If this access is denied after the end of the project, the value of such knowledge decreases within short periods of time.

Furthermore, we must take into account the different kinds of information as discussed in the previous section of this paper. As we have seen there, two kinds are likely to raise problems: the knowledge behind the information regarding project requirements, and the knowledge behind the information concerning functional and non-functional requirements.

First, the knowledge regarding project requirements, such as reasons for timelines or customer business goals, which the customer will only deliver to project leaders. This knowledge should remain inside the project management team. These people should sign a non-disclosure agreement regarding this knowledge. Since the value of such knowledge decreases very fast after the project is finished, it is possible to supervise this agreement and to demand a penalty in the case of knowledge misuse.

Second, functional and non-functional requirements and the associated knowledge are mostly very complex and recorded in voluminous documents. Making impossible to copy these

documents is a technical issue. Consequently, the experts from the supplier's side can only use this knowledge without integrating it into the knowledge of their organization. At the end of the project, the customer must prevent the supplier's experts from having access to the documents.

Therefore, it is possible to rent relevant knowledge to the supplier. A prerequisite for this lies in agreeing on some special contract regulations and in securing that the documents delivered be difficult to copy. This is just a technical issue (Krogh, 2012).

4.3.2 Renting Knowledge regarding Abilities

Following the same arguments, we can apply the rent model to knowledge about the abilities of the supplier as analyzed in the previous section. First, the parties may also offset the rent with the delivered software price. We suggest calculating the rent within the project price. If possible, the parties should agree that the customer pays a certain amount of the overall price for this knowledge. It is particularly worth noting that this amount is even due if the customer cancels the SDP.

Nonetheless, we have to consider knowledge regarding the supplier's technical abilities, especially regarding the core of business expertise. The customer may be interested in this information for two reasons. First, it may help in negotiations and cooperation with other suppliers, namely with direct competitors of the supplier who delivers the knowledge. Second, the customer may use it to be more independent from the supplier after finishing the project, especially in the field of maintenance and when it comes to developing future software releases. In our empirical survey, 50% of the interview partners, acting at least partly as customers, admit using received knowledge outside the project.

Especially in the case that the customer cancels the contract, the supplier will fear that the customer uses the knowledge he has got during the project for future developments. Therefore, the parties should agree on a price for the delivered knowledge, which must be paid by the customer also in case of cancellation.

However, the supplier can protect his knowledge, too. This knowledge about the supplier's abilities is also very complex and mostly documented in repositories and libraries. In order to use this information outside the single project, the customer must get permanent access to these documents. The supplier should present this knowledge without really sharing it. It is possible to show the needed knowledge proving that the needed knowledge is

available, and without delivering all the documents specifying details of the supplier's business know-how.

4.3.3 Consequences of Renting Knowledge

As shown by Barzel (1997), the rent of a property is connected with the problem of overusing this property. In the case of knowledge, this means that using it outside the project may reduce its value dramatically. Therefore, both parties, customer and supplier, should assess the situation and the potential for such misuse before signing the contract. If they know the value of their business knowledge, they can agree on a rent model on knowledge as described before.

To sum up the sections 4.3.1 and 4.3.2, both parties signing a fixed-price SDP contract must be aware of the fear of delivering part of one's own business knowledge to the other side. As we have shown, it is possible to offset a rent of this knowledge with the agreed fixed-price. For some kinds of knowledge, a penalty can be agreed upon in case of misuse or if the project fails and the customer does not pay the agreed price.

To implement a rent model regarding this knowledge it is necessary to secure that the other party has no permanent access to the documents containing the knowledge. This is just a technical issue. "Show and present, but do not deliver your documents" may be seen as the golden rule of implementing the rent model for PR on business knowledge.

We will now discuss the opportunity for a wage model agreement under FPC.

4.4 The Wage Model

Let us recall that the parties sign a fixed-price contract in most SDP cases. Therefore, at first glance, there is no place for a wage model regarding knowledge. However, we claim that this view is wrong in connection with knowledge in SDPs.

First, we must face the problem that also in the case of a time and material contract the factual PR on knowledge may go over to the other side. If the customer completely delivers all the documents to the supplier, the sender cannot be sure that the receiver might use it in other projects. Therefore, also in this case the customer may lose PR on own knowledge. This is the same problem as in the case of the fixed-price contract.

Second, we can also apply Barzel's wage model to the FPC in SDPs. It is not about an overall fixed-

price connected with the project result, it is about knowing if the PR on I&R are rented to the other side, or if the owner uses the other side's labor just to produce some goods.

The goods produced by using the customer's business knowledge are the technical specifications and the software designs needed by the programmers for software realization. In an SDP, it is possible to integrate the business analysts, the requirement engineers, and the system designers needed for the time work into the customer's organization for the time required to produce these specifications. It is not important whether the work of these experts is paid by time and material or if the wage is part of the overall fixed-price. The crucial fact is that they lose the PR on used knowledge when they leave the location where they had had access.

One might object that they will mentally take this knowledge with them. Nonetheless, as discussed before, in cases of complex business information the value of the individual knowledge decreases fast if this person has no longer access to the documents containing knowledge details and describing information connections and relations.

Consequently, if the supplier's business analysts are working under the control of the customer, with no option of taking away business information from the customer's offices, we can see this part of the project as work done with a wage model, and with or without payment of an overall fixed-price.

Following Barzel (1997), the customer must be aware of possible shirking from the supplier's side in this case. Given that the supplier's experts produce technical specifications for the software and even software architecture under the customer's control, the customer is responsible for the quality of these documents. It is difficult to identify the origin of software development result problems: they might be due to low information quality delivered by the customer, or to poor work of the experts producing the specifications, or to that of developers.

On the other hand, the supplier must face potential misuse of the intellectual property under the customer's control. In the case of an SDP, this means that the customer might try to use the abilities. There can especially be a transfer of the experts' knowledge to the customer. Consequently, the customer will be more independent as foreseen by the supplier.

According to Barzel's conclusion, the customer should only use the wage model if the quality of his own property is unclear. In the SDP case, this means that the customer is not sure about the quality of the specified requirements and the availability of the needed information and business know-how.

However, the customer must in this case be capable of effectively controlling the work of the supplier's experts and of evaluating the quality of the produced results. However, the last mentioned issue will be difficult because specifications and designs are not testable in the same way as developed software (Wieggers and Beatly, 2013).

Finally, we will discuss the case of a wage model regarding the supplier's business knowledge. This is possible when experts from the customer's side are working under the supplier's control, using the supplier's information and knowledge. In an SDP, there are a few cases where such a setting can happen, for instance during the design of user interfaces or during software system integration tests. We can interpret this part of the project as a wage model, in which the supplier must pay the customer a wage. We can expect the parties to offset this wage with the overall fixed-price. We suggest that the parties evaluate the value of this work during the contract negotiation.

5 CONCLUSIONS

In this paper, we have shown that software development projects require the customer and the supplier to share different kinds of I&K. The customer must deliver I&K derived from business goals (project requirements), information coming from external constraints (process requirements), and system requirements. On the other side, the supplier must deliver I&K regarding his own abilities. Sometimes, delivering information is sufficient. But the parties must often share their business knowledge, which can lead to misuse outside the project by the party receiving the knowledge. Our empirical study shows that this risk actually exists. In addition, at the beginning of an SDP both parties do not know the exact I&K is needed. Therefore, the contractors deliver worthy of protection knowledge as late as possible, requirements change.

We have described this situation in terms of the PR theory, claiming that the knowledge of an organization is an intellectual property of its owner. However, the owner cannot protect by copyright, patent, or trademark most of the knowledge needed in an SDP. Consequently, we have developed options for the protection of PR on knowledge straight from the classic models of the PR theory.

We have considered two options: rent model and wage model. In both cases, we face two issues: how to pay the rent respectively the wage, and how to protect PR, especially after the project is finished.

In the case of knowledge renting, we suggest offsetting the price for the delivered knowledge with the overall fixed-price. Therefore, the parties should consider the value of the needed knowledge during contract negotiation. The price should take into account the risk of knowledge misuse outside the project. If both parties are aware of these problems, they are able to arrive at a satisfactory solution.

For the protection of PR, the parties should prefer technical solutions, granting knowledge access just for the time needed. Such technical restrictions are possible by the use of document management systems and the implementation of knowledge management software. Considering the fact that knowledge about complex business processes rapidly decreases if the access to documentations is denied, an effective rent system is possible if such a system is used.

Whilst a rent model option is a possible model for using both customer and supplier knowledge, the wage model is just an option for the use of customer knowledge during technical specification and system design. The customer might pay the wage as part of the overall fixed-price. Analysts and designers can work under the customer's control and use the customer's facilities. This makes it easy to protect knowledge from misuse, because controlling the access to customer documentations and knowledge management systems is easy to organize.

The issue which option the parties should prefer will depend on the quality of the needed knowledge and on the possibilities of securing it by technical means. If knowledge is well-described in documents and if it is possible to protect the delivered documents containing knowledge by technical means from misuse, the parties should prefer the rent model. Otherwise, the wage model is a better way to share knowledge. In both cases, the parties should be aware of the following three principles.

- (a) Both sides have protection-worthy knowledge.
- (b) Both sides have to set a price for this knowledge. They should agree upon the value of the shared knowledge during contract negotiation and offset it with the overall price.
- (c) The parties should agree upon payment for received and used knowledge in case the project fails and the customer does not pay the negotiated price for the software system.

Following these principles, customer as well as supplier can minimize the risks resulting from sharing knowledge in SDPs. Further research can directly pick up here.

REFERENCES

- Ackoff, R.L. 1989. From data to wisdom. *Journal of Applied Systems Analysis* 16, pp. 3-9.
- Al-Ahmad, W., Al-Fagih, K., Khanfar, K., Alsamara, K., Abuleil, S., Abu-Salem, H. 2009. *A Taxonomy of an IT Project Failure: Root Causes*. International Management Review 5(1), 93-104.
- Aubert, B.A., Patry, M. and Rivard, S. 2003. *A tale of two outsourcing contracts. An agency-theoretical perspective*. Wirtschaftsinformatik 45, 181-190.
- Badenfelt, U. 2011. Fixing the contract after the contract is fixed: A study of incomplete contracts in IT and construction projects. *International Journal of Project Management*, 29, 568-576.
- Barzel, Y. 1997. *Economic Analysis of Property Rights*. Cambridge University Press.
- Benaroch, M., Lichtenstein, Y., Wyss, S. 2012. *Contract Design Choices in IT Outsourcing: New Lessons from Software Development Outsourcing Contracts* (April 20, 2012). Available at SSRN: <http://ssrn.com/abstract=2137174> or <http://dx.doi.org/10.2139/ssrn.2137174>.
- Beulen, E. and Ribbers, P. 2003. *IT Outsourcing Contracts: Practical Implications of the Incomplete Contract Theory*. Proceedings of the 36th HICSS.
- Bogers, M. 2011. The Open Innovation Paradox. Knowledge Sharing and Protection in R&D Collaborations. *European Journal of Innovation Management* 14(1), pp. 93-117.
- Choo, C.W. 1996. The Knowing Organization: How Organizations Use Information to Construct Meaning, Create Knowledge and Make Decisions. *International Journal of Information Management* 16(5), pp. 329-340.
- Chen, Y., Bharadwaj, A. 2009. An Empirical Analysis of Contract Structures in IT Outsourcing. *Information System Research* 20(4), 484-506.
- Chua, C. E. H., Lim, W. K., Soh, C., Sia, S. K. 2012. Client strategies in vendor transition: A threat balancing perspective. *The Journal of Strategic Information Systems*, 21(1), 72-83.
- Clark, A., Chalmers, D. 1998. The extended mind. *analysis*, pp. 7-19.
- Cole, D. H.; Grossman, P. Z. 2002. The Meaning of Property Rights: Law versus Economics? *Land Economics* 78(3), pp. 317-330.
- Craswell, R. 2006. Taking Information Seriously: Misrepresentation and Nondisclosure in Contract Law and Elsewhere. *Virginia Law Review*, pp. 565-632.
- Dey, D., Fan M., Zhang, c. 2010. Design and Analysis of Contracts for Software Outsourcing. *Information Systems Research* 21(1), 93-114.
- Dijkstra, E.,W. 1972. The Humble Programmer. *Association for Computing Machinery* 15(10), pp. 859-866.
- Dwivedi, Y.K., Ravichandran, K., Williams, M.D., Miller, S., Lal,B., Antony, V., Muktha, K. 2013. IS/IT Project Failures: A Review of the Extant Literature for Deriving a Taxonomy of Failure Factors. *IFIP Advances in Information and Communication Technology* (402), pp. 73-88.
- Ebers, M., Semrau, T. 2015. What drives the allocation of specific investments between buyer and supplier?. *Journal of Business Research*, 68(2), 415-424.
- El Emam, K., Koru, A. G. 2008. *A Replicated Survey of IT Software Project Failures*. IEEE Software 25(5), pp. 84-90.
- Friesike, S. 2011. *Profiting from Innovation by Managing Intellectual Property*. Doctoral Dissertation, University of St. Gallen.
- Fink, L., Lichtenstein, Y. 2014. Why Project Size Matters for Contract Choice in Software Development Outsourcing. *The DATA BASE for Advances in Information Systems*, 45(3).
- Gassmann, O., Kausch, C., & Ellen, E. 2010. Negative side effects of customer integration. *International Journal of Technology Management*, 50(1), pp. 43-63.
- Gläser, J. and Laudel, G. 2010. *Experteninterviews und qualitative Inhaltsanalyse*. VS Verlag, Wiesbaden, , 4th edition.
- Glinz, M. 2007. On non-functional requirements. In *Requirements Engineering Conference 2007*. pp. 21-26.
- Hart, O., Moore, J. 1990. Property Rights and the nature of the Firm. *Journal of Political Economy* 98(6), pp. 1119-1158.
- IEEE 1998. *IEEE Recommended Practice for Software Requirements Specifications*. Institute of Electrical and Electronics Engineers.
- Klammer, B. 2005. *Empirische Sozialforschung. Eine Einführung für Kommunikationswissenschaftler und Journalisten*. Utb, Konstanz.
- Krogh, G.v. 2012. How does social software change knowledge management? Toward a strategic agenda. *Journal of Strategic Information Systems* 21, pp. 154-164.
- Lichtenstein, Y. 2004. Puzzles in software development contracting. *Communications of the ACM*, 47(2), 61-65.
- Liebekind, J. P. 1996. Knowledge, strategy, and the theory of the firm. *Strategic management Journal* 17 (S2), pp. 93-107.
- Mayer, H., 2012. *Interview und schriftliche Befragung. Entwicklung, Durchführung und Auswertung*. Oldenbourg Wissenschaftsverlag, München.
- Myers, M. D. and Newman, M. 2007. The qualitative interview in IS research: Examining the craft. *Information and Organization* 17(1), 2-26.
- Norman, P. M. 2001. Are your secrets safe? Knowledge protection in strategic alliances. *Business Horizons*, 44(6), pp. 51-60.
- North, D. C. 1993. Institutions and credible commitment. *Journal of Institutional and Theoretical Economics (JITE)/Zeitschrift für die gesamte Staatswissenschaft*, 11-23.
- Osei-Bryson, K. M., Ngwenyama, O. 2011. Using decision tree modelling to support Peircian abduction in IS research: a systematic approach for generating and evaluating hypotheses for systematic theory

- development. *Information Systems Journal*, 21(5), pp. 407-440.
- Paasi, J., Luoma, T., Valkokari, K., & Lee, N. 2010. Knowledge and intellectual property management in customer-supplier relationships. *International Journal of Innovation Management*, 14(04), pp. 629-654.
- Pohl, K. 2013. The three dimensions of requirements engineering. In *Seminal Contributions to Information Systems Engineering* pp. 63-80.
- Rowley, J. 2007. The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science* 33(2), pp. 163-180.
- Schnell, R., Hill, P. and Esser, E. 2011. *Methoden der Sozialforschung*. Oldenbourg Wissenschaftsverlag, München, 9th edition.
- Schurz, G. (2008). Patterns of abduction. *Synthese*, 164(2), pp. 201-234.
- Standish Group 2010. *CHAOS MANIFESTO, The Laws of Chaos and the CHAOS 100 Best PM Practices* The Standish Group International.
- Swigon, M. 2013. Personal knowledge and information management – conception and exemplification. *Journal of Information Science* 39(6), pp. 832-845.
- Teece, D. J. 2000. Strategies for managing knowledge assets: the role of firm structure and industrial context. *Long range planning*, 33(1), pp. 35-54.
- Tiwana, A. 2004. Beyond the black box: knowledge overlaps in software outsourcing. *Software, IEEE*, 21(5), pp. 51-58.
- Walton, D. 2014. *Abductive reasoning*. University of Alabama Press.
- Wieggers, K., Beatty, J. 2013. *Software Requirements*. Pearson Education.
- Williamson, O. E. 1983. Credible commitments: Using hostages to support exchange. *The American Economic Review*, 519-540.
- Williamson, O.E. 1985. *The Economic Institutions of Capitalism. Firms, Markets, Relational Contracting*. New York, (1985).
- Zeleny, M. 1987. Management support systems: towards integrated knowledge management. *Human Systems management* 7(1), pp.59-70.