Contracting in Agile Software Projects: State of Art and How to Understand It

Shi Hao Zijdemans[1] and Christoph Johann Stettina[1,2]

[1] Leiden Institute of Advanced Computer Science, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
[2] Centre for Innovation The Hague, Leiden University, Schouwburgstraat 2, 2511 VA The Hague, The Netherlands

Abstract. The iterative nature of Agile methods paves the way for new and more dynamic contract arrangements in software development projects. However, while new types and adaptations of existing contract types have emerged in practice, a shared view on these arrangements is missing in literature. In this paper we review common contract types discussed in Agile and traditional project management. Based on existing literature and empirical data collected during a workshop and semi-structured interviews we present a preliminary framework to help understand and choose contracting practices in context.

Keywords: Agile project management, contracting, procurement practices.

1 Introduction

Agile project management is becoming increasingly popular in and outside the domain of software development [1]. While an Agile approach brings advantages to both the software supplier and the customer, it also introduces new challenges. Major challenges when implementing Agile methods in traditional software development organizations are caused by the necessity to adapt existing practices to fit the iterative nature of Agile projects in context [2]. Contracting is one of such domains of practice potentially affected by agile methods [3].

Contract negotiation is a crucial part of a software development project, because it has an influence on the entire project, its practices, the underlying patterns of action and governance structure. It can be challenging for a software supplier and customer to find a common ground of trust without prior collaborations. A good contract agreement is important for further relations [3]. Still, frequently we encounter (small) software companies that are unaware of existing contracting practices and the possibilities they offer.

Fixed-price contracts have been used frequently with Agile projects. Here, the high level of uncertainty in software projects, and the flexibility that is intended for the Agile approach conflict with the planning-driven nature of fixed-price contracts [3]. Therefore, the Agile contracting problem has gradually received more attention [4]. Alternative practices or contracting frameworks have been

G. Cantone and M. Marchesi (Eds.): XP 2014, LNBIP 179, pp. 78–93, 2014.

proposed in literature, but every project is different and therefore we believe that there is no one silver bullet for the Agile contracting problem.

Although an initial discourse can be found in literature, the discussion on contracting in Agile projects is scattered across practitioners literature and individual cases. Current contributions are limited to single case studies discussing individual frameworks in practice, and predominantly covering the perspective of software developers. This paper contributes an analysis of major contract types discussed in Agile and traditional project management literature. Empirically collecting the perceptions of different stakeholders such as legal advisors, Agile coaches and business owners we present a preliminary tool to understand and help choosing an appropriate configuration of contracting practices in context.

2 Related Work

In an attempt to counter contracting challenges in Agile projects, multiple contract types have been discussed in literature. In this section we review the contract types in Agile and traditional project management literature and their application in Agile contexts.

Fixed-Price Contracts Besides a prespecified price, a fixed-price contract contains a fixed deadline and an assumably complete specification of the software system in question. In practice a complete specification is rather difficult and costly to create (e.g. customer often cannot specify requirements accurately [4]) and hinders innovation (e.g. design changes in evolving system are only possible via expensive and often troublesome change requests). The supplier is financially responsible for any cost overruns. Therefore it is recommended that the supplier does not consider this contract type unless the system requirements are clear and unambiguous [5].

Target-Price Contracts In target-price contracts the risk of overruns are shared between the customer and the supplier [6]. The contract contains a target of effort (in man hours), a negotiated profit on the project and often a deadline. If the actual amount of hours exceeds the target, the customer pays 50% (or another agreed upon percentage) of the extra costs. And vice versa, if the actual amount is less than the target, the customer pays 50% of the difference between the actual amount and the target. This contract type can also be set up with a minimum and maximum amount of hours that can be charged.

Time & Material Contracts In Time & Materials contracts the supplier is paid based on its hourly rate [7]. At prespecified time intervals (e.g. monthly) a bill is sent to the customer, containing the amount of man hours and the total price according to the supplier's hourly rate. T&M contracts do not contain a complete specification of the system, and the project can be ended whenever the customer wants. The risk in T&M projects is carried by the customer; the supplier is paid fully for every hour, and therefore does not have an incentive to complete the project quickly [3]. The only meaningful incentive is high competition and the prospect of follow-up contracts [5].

Cost-plus contracts In cost-plus contracts, the customer pays for all the supplier's costs plus an additional fee that contains the profit [8]. There are

different options to construct the fee. *Cost-plus-fixed-fee* contracts have a pre-specified fixed profit fee [5]. In *cost-plus-incentive-fee* contracts, suppliers receive a higher profit fee if they meet or exceed a specific target performance, which is agreed in advance by supplier and customer. This introduces an incentive for the supplier to keep the costs down [5]. Whereas the incentive fees are based on objective calculations comparing certain measures (e.g. costs, delivery time), *Cost-plus-award-fee* contracts utilize a more subjective fee. Award fees can be earned when the supplier meets significantly higher levels of performance, quality, timeliness or responsiveness in the project [5]. Cost-plus contracts require that the supplier's books be audited [5].

We will now present frameworks dedicated to contracting in agile software projects available in existing literature.

Agile Fixed-Price Contracts This book specifically devoted to Agile contracts [9] provides a contract model for 'Agile Fixed-Price Contracts'. The book gives a detailed guide on the setup, the tender phase, and the practical aspects of project management for Agile fixed-price contracts. This contract type contains a fixed price ceiling and the scope of the software system is left variable. It incorporates interesting features, such as 'exit points' and a 'Checkpoint phase'. Exit points are predefined points in time where the parties may terminate the project in a controlled manner. A Checkpoint phase is a period of x sprints or a performance scope of y storypoints, that act as a test phase of the cooperation between customer and supplier.

Agile Contract Primer - Multi-phase variable-model The *Agile Contract Primer*[10] provides suggestions for better understanding of IT lawyers' perspective, and multiple suggestions for IT lawyers are given to get a better understanding of the implications of the Agile methodology and systems thinking. The article also discusses *Multi-phase variable-model frameworks* which take into account that the uncertainty and risk profiles of software projects change over time. In a multi-phase framework, any phase can use any contracting model. For example: choose fixed-price for the first phase, where the Product Backlog is created and hence the most uncertain phase of the project is traversed, and then switch to T&M.

adVANTAGE The 'adVANTAGE' pricing model combines elements of fixed-price and T&M contracting models [3]. In this contracting model, the software supplier is paid for their effort after each sprint. The contract provides an idea of the overall scope in terms of user stories, time and budget. The required effort for each user story is estimated in the beginning of the project. These estimates are used in all sprints as an orientation point for the bill, as the actual effort for each user story is compared to the estimated effort. The final price for each sprint is compensated according to any differences in the actual and estimated effort (like target-cost contracts). The customer can prioritize, eliminate or add user stories at the beginning of each sprint, and in doing so, has to take into account the supplier's price estimates and its own budget ceiling.

Collaborative Agile Contracts Thorup and Jensen (2009) [11] proposed and tested this model in two commercial projects. The biggest distinctive feature

of this contracting model is that the payment is delayed until a certain criterion is fulfilled [11]. In most contract types this criterion is a calendar date, however in Collaborative Agile Contracts, the criterion is to reach a predefined milestone where the customer is getting value from a specific delivered increment of the system. Generally both parties would like to reach these milestones as quickly as possible. The supplier's efficiency will be rewarded with a quicker payout. The customer will think more carefully when deciding what features he/she wants to have implemented, because all the increments will be paid for separately. Further, Concha et al (2007) [12] propose an approach called "Agile Commitments" which provides complementary practices for Agile projects. One of the objectives of Agile Commitments is to define and specify the commitments between the customer and supplier, which can provide a baseline for contract negotiation.

Current literature discusses a variety of contract types, however, the evaluation in projects following Agile methods is sparse. While contracting is discussed on the level of entire frameworks, our literature analysis points out that contracting types resemble a mix of distinct practices. Such concrete practices are fixed-price agreements or payments-per-sprint, often accompanied by further incentive and governance mechanisms. Based upon the literature reviewed above we find it appropriate to pose the following research question:

1. *What contract types and the contained concrete practices are suitable for different Agile project contexts?*

3 Method

Following our research goal we want to create an understanding of contracting practices in real world projects. To pursue this goal, we chose to conduct an inductive case study research approach as commonly proposed by literature [13,14]. Qualitative case studies allow to look at complex problems in context while developing rich and informative conclusions.

3.1 Data Collection: Workshop and Semi-structured Interviews

In order to establish an appropriate understanding of the topic matter we first conducted a scientific workshop for Agile professionals. The 45-minute workshop took place in context of a conference on Agile methods and was held with 8 participants. This pre-study allowed us to collect many perceptions on Agile contracting in a short amount of time, and gave us a general idea of current practices in use. Based upon the workshop results and existing literature we created an interview guide and conducted semi-structured interviews. The face-to-face interviews took place between April and June 2013 with five participants in different organizations. We interviewed participants from a variety of roles to ensure an inclusion of different perspectives on the topic. The interviews had an approximate duration of 60 minutes and took place at the participants' workplace. Before each interview, we asked the participant for permission to

record the interview. We mentioned that all data from the interview will be anonymized. All interviews were voice recorded and transcribed. We further asked participants to provide process descriptions if available.

The structure of the semi-structure interviews was as follows: We started with general questions about the participant, the organization, their knowledge on (Agile) software development and contracting. We then continued with questions related to their experiences with contracts in Agile projects and challenges they are facing. Example questions were: *How does the process of acquiring projects look like in your organization (from initial project request to start of development)? What are important aspects to consider when contracting Agile software projects? What are the challenges you are facing with current contracts in Agile projects? What do you think could be improved in your current way of contracting software projects?*

The interview questions were continuously revised in course of the study. Given the variety of backgrounds of participants, we altered the questions according to a given situation (e.g. special knowledge in a particular area or limited in another). For example, with one of the participants who is a jurist in R&D, we skipped some of the more technical software development questions.

3.2 Data Analysis

To analyze the data from the interviews, we applied a qualitative data coding technique similar to that of Grounded Theory (GT) by Glaser and Strauss [15]. We used the coding technique, in contrary to GT, however, we used semi-structured interview guides based on existing literature and findings from the previously conducted workshop. The method is especially suitable for areas of research that have not been studied in great detail before. Such would allow us to collect rich information while enabling direct exchange with the practitioners.

All interviews have been fully transcribed. After the transcription, we performed 'open coding' [15] using the tool *Open Code* [1]. There, the transcripts were analyzed line by line and inspected for recurring key themes. Each key theme was then assigned a 'code' (i.e. a descriptive label), which forms a level of abstraction in analyzing the transcripts. By constantly comparing the codes against each other (i.e. Grounded Theory's *constant comparison method*), we categorized the codes to extract underlying concepts. An example:

Quote: *"In calculating our price, we ensure that there is a certain financial buffer for us, we use it for contingency planning. We define that buffer upfront based on our expectations and if along the way the customer [makes change requests] then we can be easier about it.*

Key Theme: *"Contingency planning for fixed-price projects"*

Code: Fixed-price modifications

Category: Fixed-price

[1] `http://www.phmed.umu.se/english/divisions/epidemiology/research/open-code/`

4 Results

In this section we present the results collected. An overview of the participants is given in Table 1. In order to preserve the participants' confidentiality, we assign a an ID to each participant (P1-P5). All interviewees have several years of experience with contracting and contract negotiations in software projects. The majority of participants apply Scrum in their projects. P1 is a jurist in a legal department and does not apply any project management framework directly. P2 is familiar with agile methodologies, however, rather than implementing a particular method the company applies a few agile practices such as iterative development and frequent customer reviews. The size of the organizations varied heavily among the participants; from 10 to 145.000 employees. Supportive quotes will be provided, which have been translated from Dutch.

Table 1. Interview participants and descriptive variables

Participant	Type of Organization	Size of Organization	Function	Agile method	Experience
P1	R&D	Large	Jurist	-	Legal advisor in contract negotiations
P2	Web Development	Small	Co-owner	-	Contract negotiation with customers
P3	Auditing, Tax Advisory, Consulting	Large	Partner	Scrum	Contract negotiation with customers
P4	Consulting & Training	Large	ScrumMaster, Coach	Scrum	Training and consulting on business agility and performance improvement
P5	Software Development	Large	Agile consultant	Scrum	Research in Agile

In Figure 1 we present an overview of all the categories that we have distinguished from the different codes. To provide a quantitative view, all the categories were assembled and ranked by the the amount of codes they contained (Fig. 1). We divided the categories into two groups: *concrete practices* and *affecting factors*. As we can see in the figure, the majority of codes is related to concrete practices discussed by our participants. In total we found 330 codes distributed over 31 categories.

During the coding we identified the major challenges in Agile projects mentioned by our participants as:

1. *Lack of Customer Involvement: if the customer is not involved in the project enough, it will be more difficult for the supplier to develop the right system.*
2. *Difficulties in Scope Change: if the entire scope of the project has been completely specified in the contract, scope changes become very difficult.*

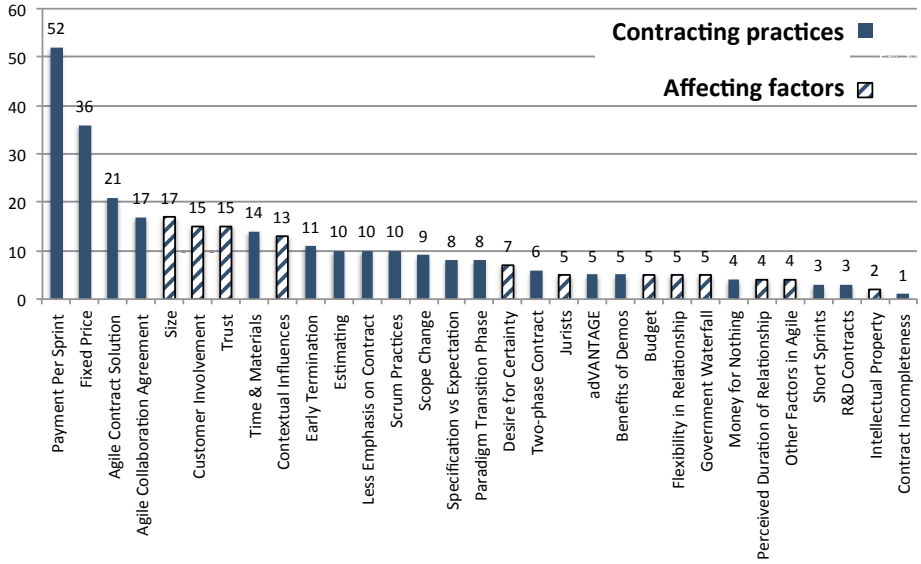

Fig. 1. Overview of the categories and the amount of codes they contain

3. *Lack of Trust: Before the project has started, customers tend to lack trust in that, by using the Agile approach, the supplier can deliver the right system without having to specify the system completely upfront.*
4. *Too much Trust: When the project is started, customers tend to have too much trust in that the supplier will deliver a good system, which in turn leads to less effort being put in the project by the customer.*

4.1 Concrete Practices in Use

Is this section we will report on the practices discussed by our participants. By such we mean observable patterns of human action such as 'payment per sprint'.

Fixed Price. 'Fixed Price' was mentioned many times. Besides the challenges regarding fixed-price contracts that have been described in related research literature (e.g. the customer's desire for certainty and upfront specification), another important distinction was often emphasized: the distinction between *specifications* and *expectations*.

> *"it's way more important to meet the expectations than the specifications. Of course you do need specifications, but those should be more like goals: 'what is the intended goal of the system?'. That is, however, difficult to state [formally], but it's better to verify the system against the goals instead of the specifications."* - P4.

Payment per Sprint. Among the participants that actively use Agile methods, the opinions on payment per sprint varied widely. While P4 has *"never seen contracts done in that way"*, P3 currently uses this method of payment with success, and is confident that this is a very effective way of handling the payments in Scrum projects. In contrary to the adVANTAGE pricing model [3], this organization uses a fixed price per sprint.

> *"Billing is easy; I want to do that per Sprint. I want to do the billing as soon as possible... however it doesn't always work, because before you know it you're behind 2-3 weeks. But in general...there will be a bill per Sprint, and we try to do this as close to the end of the Sprint as possible. It is a fixed amount per Sprint by the way, so whether or not we make the exact agreed hours or not, that doesn't matter. This also has to do with the current economic conditions. Every company is trying to invoice quickly these days, because payments are taking so long."* - P3.

Benefits. P3 believes that one of the benefits for the supplier is the fast payment, and thereby some reduction of risk. The benefit of payment per sprint for the customer is the flexibility that can be offered in multiple aspects of the relationship. In the contract an agreement is stated for a certain amount of sprints. Every time the agreed amount of sprints has been completed, the customer can buy more sprints. At the start of the project customers can agree to buy a small amount of sprints to get introduced to the Agile approach. In these sprints the customer can witness the Product Backlog being created and managed, and he could receive the first part of the system.

> *"I always say 'You only get on board for one sprint, and when you're dissatisfied with a sprint, we can always just say goodbye'. Those are always enlightening insights for the customer."* - P3.

P3 elaborates that it is of course the idea to build the desired system together, but it is best to offer options like these to the customer just in case the collaboration works out differently. Another benefit that was mentioned by P3, is that the payment directly follows the acceptance of the sprint, which makes the payment more 'natural' for the customer.

> *"It's very logical right? You have a sprint, you get a report with the sprint, you get a demo with the sprint, well then it's pretty logical to add a bill to the sprint as well."* - P3.

Disadvantages. P4 stated that a disadvantage could lie in the fact that such a contract could create a short term vision. When there is a contract for a couple of sprints, the supplier will likely focus on these sprints and try to please the customer by presenting as much functionality as possible. This would tempt developers to put the quality of the system at risk by creating a technical debt, leading to poor software architecture.

P3 mentioned that this payment structure requires discipline in the financial aspect. He stated that his environment — one of the largest professional services

organizations in the world, that mostly does financial advisory — plays a role in the successful realization of this payment method.

> *" However this requires discipline in the financial aspect, something that IT people are not very good at. I am being helped with that by my environment; in this house they always know everything financially."* - P3.

Agile Collaboration Agreement. When all the features have been formally specified upfront, then there is very little room for any scope change. P5 stated that it is essential to solve this problem, and that ideally in Agile contracts *"scope change should be freed"*. To free scope change P5 suggested forming a joint-venture, or setting up 'Collaboration Agreements' for Agile software projects.

> *"One way is to form a joint-venture. Then the software supplier gets a share of the profit...I believe this is a possible way to do it. Another option is aimed towards Collaboration Agreements..that would contain things like the responsibilities and expectations of the parties"* - P5.

The importance of proper collaboration and trust in Agile projects was also mentioned by P3 and P4. They believe that the customer should monitor the progress of the project more. According to them the responsibilities and expectations of the customer should be clarified and stated in Agile contracts for more effective collaboration, and a higher chance of project success.

> *"I think the best would be a pure Agile way, in which you, as a customer, can watch over the progress...very frequently...And of course, when you see that the team performs well, then you give them a bit more freedom, but if you get the impression of 'this doesn't suffice', well then as a customer I would be on that a bit tighter...And it's very possible to state that in a contract I think."* - P4.

P3 emphasized that *"on the one hand customers should have more trust, and on the other hand they should have less trust"*. They should have more trust in the fact that they can get a good end result without a complete upfront specification. *"They should have less trust that, once they have started the relationship, the supplier will make it all alright. They have to critically watch the supplier"*.

Early Termination. Something that was often mentioned, was the 'Money for Nothing' part of Sutherland's 'Money for Nothing, Change for Free' [16], which is an agreement that in the case when the customer decides to terminate the project early (because he is satisfied enough with the delivered increments of the system), he will pay 20% of the remaining budget to the supplier. This can partly be seen as a bonus for delivering enough business value sooner than expected. The other part is a risk premium, since the supplier's resources have to be reallocated unexpectedly (P5). P4 and P5 both have not seen this being put to use very often, although they both believe that this element is required in the ideal Agile contract.

P3 reiterated the possibility to terminate a project early due to ongoing challenges. However, he was not familiar with the idea of customers terminating a successful project early. Even though a customer might be satisfied with the deliverables, budget and personnel have already been allocated. This makes it rather unlikely to stop a project before the planned end date.

> *"I have never seen that in a contract. It's very funny, because it's actually very logical to do it like that. That's very good. Although it can be seen that many customers have a lot of trouble with de-scoping, it's very hard to decide that you don't want something, even though it barely has additional business value. Customers are not used to that."* - P3.

Two-phase Contract. To handle the 'time and money' aspect of Agile contracting, P5 recommended a 'Two-phase Contract', which is inspired by the *Cone of Uncertainty* (McConnell, 1997). Especially in the beginning of a project, there is a lot of uncertainty (about the scope, price, duration etc.). This gradually declines along the duration of the project. Two-phase contracts take this into account and divides the project into two phases. The first phase is a relatively short phase aimed at getting through the initial uncertainty and creating a base of trust between the customer and supplier. P5 recommended to use fixed-price for this phase since it is relatively short, and the required effort can be estimated relatively easily. In this phase the productivity of the team is witnessed by the customer, there is a more accurate view on the requirements, the main impediments have been identified and an increment of the system has been delivered. When those elements are already in place, the parties could basically make any contract type work well for the second phase (e.g. fixed-price, T&M etc.).

4.2 Affecting Factors

Size. The size of the organization (supplier side or customer side) and the size of the project all play a role in determining whether this method of payment is feasible. P3 explained that smaller projects have an upfront total price estimate, and a planning of the sprints.

> *"But smaller projects, there you first have a total price estimate and a planning of the sprints. So formally they also buy sprints, but there is a bit more specification of the system....so a bit more towards a waterfall-like description of the end result. But when I look at other Agile contracts, this is pretty Agile."* - P3.

P2 (Web development) did not see any added value in doing the payments per cycle for his organization, because he does not have very complex cases.

Government Waterfall. Something that was mentioned by multiple participants was the regulations that are associated with software projects for the government. Governmental software projects should always have a complete upfront specification, price estimation and deadline.

"The contract model, that's where you really see the friction between waterfall and [Scrum]...And especially in the government, all the standards are aimed towards waterfall-like constructions, where everything is specified very clearly. " - P3.

Further, as dictated by EU regulations, no bilateral communication between principal and service provider is allowed before the contract has been awarded

Customer Involvement. An aspect that often poses a big challenge in Agile projects is the lack of customer involvement [17]. P3 explained his current experience with regard to customer involvement, and underlines its importance.

"We also invite the customer, and if he joins in more than once a week, then that can be seen as often. Something that also happens often is that the product owner goes and works at the customer's site. In that case he is present [with us] less than you'd ideally want to, but then that's also understandable." - P3.

Understanding Jurists. When asked what current challenges are in contracting from jurists' perspective, P1 stated that when jurists are making a contract for (two) parties, they often don't really know what the underlying intentions from these parties are. This somewhat limits them in their ability of setting up correct and accurate contracts.

"What I often encounter is that many times jurists don't know, what the intentions are of the parties." - P1.

This was confirmed by P5, who explained that he had conducted a workshop at a Dutch organization of jurists. This workshop aimed to get the jurists thinking about possible solutions. So far they believe that a solution based on Collaboration Agreements is the most feasible, and P5 said to believe that it is largely a matter of time before the effect of 'Agile Collaboration Agreements' is discovered.

5 Discussion

Following the description of main contracting types and the perceptions of our interviewees towards those, we will now proceed to discuss these types and their suitability in different context in Table 2.

According to McLeod and MacDonell [18] there are four main factor categories contributing to software development project outcomes, these are: Project Content, Institutional Context, People and Development Processes [18]. The factors influence the suitability of a contract type for a particular project (e.g. projects with ambiguous requirements can be well combined with exit arrangements, governments generally require fixed-price contracts). During contract negotiation some of these factors are easier to adjust than others (e.g. development

process and project content are generally more flexible and easier to adjust than institutional context). Here we acknowledge the strong influence of contracting practices on the development process, as a contract can enforce a specific software development process. Following this line of thought, we consider Project Content, Institutional Context and People as independent factors that determine a project context. The contract practices then have to be selected accordingly based on these first three factors.

Further, based on our discussions with participants and contract elements in literature (e.g. incentives, uncertainty mitigation [5,9]) we made a distinction between four categories of contracting practices as depicted in Table 2: 1) Contract Basis, 2) an Incentive element 3) an Uncertainty Mitigation element and 4) an Governance element. The contract basis contains agreements on fairly basic contracting matters: the pricing model, delivery date and the scope of the system. However, since these basic agreements have proved to be insufficient in many project contexts, different variations and additions have been adopted in practice to overcome these challenges. The incentive element contains practices that focus on the unfair risk distribution that has often been associated with software development [3][7][9][10][11]. Leveraging additional incentive with a specific party can be used to balance the risk distribution for a project. The uncertainty mitigation element focuses on mitigating the uncertainty concerning the price, scope or deadline estimates that have to be made in the project. The introduction of a governance element to an Agile contract is focused on establishing a collaboration between the parties that is required for Agile software projects. This primarily aims to increase the amount of customer involvement in the project.

Contract basis In our interviews we found that fixed-price contracts are mainly used in projects for the government, or small projects. Despite of the negative connotations that fixed-price contracts have received in literature [3][9][10][11], fixed-price contracts have proved to work well in small projects, due to the fact that the requirements can generally be specified more accurately if the scope is small. Further, fixed-price agreements have been mentioned as efficient for small projects, as suppliers can avoid efforts associated with payments per sprint such as additional administration and billing.

We found that payment per sprint can be a good solution to increase the quality of the end result when the requirements are ambiguous, since this allows the scope to be specified incrementally. However, this can be regarded as unnecessary to implement in small projects, in which fixed-price contracts can also be used, since payment per sprint does not provide the customer with certainty about the price, and the time of delivery.

T&M contracts are not popular in Agile projects [3][7], because they put all the risk on the customer's side. However, we found that when there is mutual trust (i.e. supplier tries to work efficiently, and customer trusts this), T&M contracts can provide a very easy and straightforward payment structure.

Incentive Target-price contracts provide incentive for the supplier to deliver the system quickly, while not assuming all of the project risk (as is the case with

Table 2. Preliminary framework for understanding contracting practices as affected in project contexts. Practice contributes benefits predominantly for: Project Owner (◐), Supplier (◑), both (●), none (○).

	Contracting Practices (affect Development Process)								
	Contract Basis			Incentive		Uncertainty Mitigation			Governance
	Fixed-Price / Scope	Time & Material	Pay per Sprint	Target-Price	Incentive Fee / Bonus Penatly	Exit Arrangement	Risk Buffer	Two Phase	Collaboration Agreement
Project Content									
Focus on Budget	◐	◑	◑	●	●	●	●	◑	○
Focus on Quality	○	●	●	●	●	○	●	●	○
Focus on Time	◐	◑	○	●	●	●	●	●	○
Ambiguous requirements	○	●	●	●	○	●	●	●	●
Large Size Project	○	◑	●	●	●	●	●	●	●
Small Size Project	●	◑	○	●	○	○	●	○	●
Institutional Context									
Government	●	○	○	○	○	●	●	●	●
People									
Uncertain Customer Involvement	○	○	○	○	●	○	○	○	●
Low Trust	◐	○	●	●	●	●	●	●	●

fixed-price contracts), since any cost overruns are shared evenly. Pricing models can be expanded with an incentive fee (i.e. provision for the adjustment of the total profit), that provides the supplier with an incentive to deliver the system more quickly or to reduce the costs. It is not recommended to use incentive fees when the requirement for the system are ambiguous [5].

Uncertainty Mitigation In our interviews we found that a risk buffer is a practice that is often used by the supplier to mitigate the uncertainty that revolves around the estimates that have to be made in fixed-price contracts. This buffer contains a part of the budget and a part of the project time span, and can be used by the supplier to handle any 'unforeseen' change requests in the last phases of the project. The customer does not need to be aware of this.

In our interviews and in literature [10] we found that the flexibility of the Two-Phase contract makes it a suitable contract type in different contexts of Agile projects. After the first phase (i.e. the most uncertain period of the project), the budget and time estimates for the rest of the project can be made more accurately. The breakdown into two phases can be particularly helpful in large projects with ambiguous requirements. Two different pricing models can be applied in this contract type, and these can be selected according to project circumstances (e.g. preferences of the customer).

Governance A collaboration agreement can be included in a contract when the amount of customer involvement is uncertain or expected to be low (e.g. if

the customer is new to the Agile principles). In the Agile approach, the customer has an important role during the project, which is often not understood. A collaboration agreement helps create a common understanding by explicitly focusing on the responsibilities and expectations of the collaboration, and having the customer formally agree on this.

Based on the main challenges identified in our data (lack of customer involvement, difficulties in scope change, lack of- and too much trust), collaboration agreements can help by: Including collaboration expectations in contracts and creating an understanding of Agile values across the parties. Further, as collaboration agreements define how the teams work together expectations towards documentation and testing can be added. Especially as there is quite some uncertainty in Agile methods regarding documentation requirements [19,17], clarifying these expectations here can be helpful.

Although the contracting practices each have their benefits, many of them require additional effort in order to be implemented correctly. It is up to project managers (or whoever is in charge of choosing a specific contract type) to assess this trade-off, and to evaluate whether the practice is applicable in the specific project context.

5.1 Bias and Limitations

Although we followed a rigorous research process there are obvious limitations to our study. The major restriction is the limited amount of participants. We based our data on a workshop conducted with eight participants and five semi-structured interviews. Our sample might be difficult to reproduce and is not representative. To address external validity we support our preliminary model with findings available in agile as well as in traditional project management literature. To address construct validity we conducted several reality checks with experts and revised the final paper and preliminary model with IT lawyers. We believe that the quotes and extracted code categories put into perspective with already available individual case studies, provide a good overview of relevant themes and a solid ground for a further quantitative research.

6 Conclusions

In this paper we discuss contracting practices for Agile software development projects. First, we review common contracting types as discussed in Agile and traditional project management literature, and present our empirical evaluation of these contracting types applied in practice. Second, we analyze and divide contracting types into four categories of contracting practices (basis, incentive, risk mitigation and governance) and affecting factors (project content, institutional context and people). Finally, based on our analysis and empirical evidence, we give concrete recommendations to practitioners.

Based on our study the categories of contract practices presented in Table 2 contribute to the understanding of different configurations of contracts. In that

sense we can only provide examples of such configurations and their advantages and disadvantages in practice. Their tailoring for a project needs to be carefully considered by the involved parties. To what extent such arrangements can differ is illustrated in the following examples:

1. *Small size web development project: fixed-price contract with a statement of hours and scope, and a collaboration agreement.*
2. *Medium size projects, with an intermediate risk level but opportunities for value creation with preliminary functionality delivered (e.g. e-Commerce or infrastructure): payment-per-sprint contract with a collaboration agreement*
3. *Large enterprise-wide implementation projects, with high risk of making inaccurate cost estimates and difficulty to elicit clear requirements from the customer: Multi-phase contract, with fixed-price in the first phase, where the planning is made, and a Collaborative Agile Contract in the next phases. Payments are made after each delivered increment of the system.*
4. *Large governmental R&D system development project, with high level of uncertainty (comparable to large Agile software projects): Two-phase contract with collaboration agreement. Target-price in first phase (concept definition), T&M in second phase (system development and demonstration)*

To scholars our findings contribute a better understanding of contracting practices in context. To practitioners we provide a preliminary framework to understand basic contracting types, their elements, and choose them according to a respective project context. We would like to help building a shared understanding and trust across project teams and legal consultants to help making better contracting arrangements in the future. As such this paper contributes to managerial and governance aspects of Agile organizations and how collaborations across such can be formed.

6.1 Recommendations for Research and Practice

To get a better understanding of the interdisciplinary possibilities and challenges, especially to understand procurement in the public sector, further interviews with jurists, public administrators and economists should prove beneficial. Further, the preliminary framework presented in Table 2 provides good opportunities to be elaborated and strengthened in a more quantitative research setting.

Acknowledgments. This research project has been supported by the Living Lab The Hague project co-funded with support from the European Regional Development Fund of the European Union. We thank all participants for generously contributing to this study.

References

1. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. Information Software Technology 50(9-10), 833–859 (2008)
2. Boehm, B., Turner, R.: Management challenges to implementing agile processes in traditional development organizations. IEEE Software 22(5), 30–39 (2005)
3. Book, M., Gruhn, V., Striemer, R.: adVANTAGE: A fair pricing model for agile software development contracting. In: Wohlin, C. (ed.) XP 2012. LNBIP, vol. 111, pp. 193–200. Springer, Heidelberg (2012)
4. Hoda, R., Noble, J., Marshall, S.: Negotiating contracts for agile projects: A practical perspective. In: Abrahamsson, P., Marchesi, M., Maurer, F. (eds.) XP 2009. LNBIP, vol. 31, pp. 186–191. Springer, Heidelberg (2009)
5. Kerzner, H.R.: Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Wiley (2009)
6. Molokken-Ostvold, K., Furulund, K.M.: The relationship between customer collaboration and software project overruns. In: Agile Conference (AGILE), pp. 72–83. IEEE (2007)
7. Steven, P.: 10 contracts for your next agile software project (2009), `http://agilesoftwaredevelopment.com/blog/peterstev/10-agile-contracts` (accessed: April 14, 2013)
8. Hofbauer, J., Sanders, G.: Defense industrial initiatives current issues: Cost-plus contracts (2008), `http://csis.org/files/media/csis/pubs/081016_diig_cost_plus.pdf` (accessed: November 14, 2013)
9. Opelt, A., Gloger, B., Pfarl, W., Mittermayr, R.: Agile Contracts: Creating and Managing Successful Projects with Scrum. Wiley (2013)
10. Arbogast, T., Larman, C., Vodde, B.: Agile contracts primer (2012), `http://www.agilecontracts.org/agile_contracts_primer.pdf` (accessed: November 20, 2013)
11. Thorup, L., Jensen, B.: Collaborative agile contracts. In: Agile Conference, AGILE 2009, pp. 195–200. IEEE (2009)
12. Concha, M., Visconti, M., Astudillo, H.: Agile commitments: Enhancing business risk management in agile development projects, pp. 149–152 (2007)
13. Yin, R.K.: Case Study Research: Design and Methods (Applied Social Research Methods), 4th edn. Sage Publications (2009)
14. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14(2), 131–164 (2009)
15. Glaser, B.G., Strauss, A.L.: The discovery of grounded theory: Strategies for qualitative research. Transaction Books (2009)
16. Krebs, J.: Agile Portfolio Management. Microsoft Press (2008)
17. Hoda, R., Kruchten, P., Noble, J., Marshall, S.: Agility in context. In: Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications, OOPSLA 2010, pp. 74–88. ACM, NY (2010)
18. McLeod, L., MacDonell, S.G.: Factors that affect software systems development project outcomes: A survey of research. ACM Computing Surveys (CSUR) 43(4), 24 (2011)
19. Stettina, C.J., Kroon, E.: Is there an agile handover? an empirical study of documentation and project handover practices across agile software teams. In: 19th ICE & IEEE-ITMC International Conference, The Hague, Netherlands (2013)