Universiteit Leiden

Computer Science

Contracting practices in traditional and agile software development

Name: Emmanouil Pilios

Date: 09/08/2015

1st supervisor: Dr. C. J. Stettina
2nd supervisor: Prof. Dr. T. Baeck

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University

Niels Bohrweg 1

2333 CA Leiden

The Netherlands

To my family

Στην οικογένεια μου

Contents

Figures

1. INTRODUCTION

1.1 Summary

This thesis is about contracting practices that are used by software development companies and their clients. More specifically, it addresses what the current state of knowledge is on these and how Agile methodologies can be adjusted to the current methods that are being used.

Contract negotiation is a very important aspect for software development projects and project management in general. This is because it can affect the entire structure of the project, its methods and the actions that need to be followed in order to be completed. The collaboration for the project requires mutual trust between the supplier, who is the software developer, and the customer, who is the contractor, because they have to find a successful way to share the risks between them. Thus, a good contract arrangement can be important for future agreements for both.

While software product development increasingly moves towards collaboration with and across small entrepreneurial companies, our understanding in contract arrangements to enable more flexible collaborations is limited. For example, as Kettunen and Laanti (2008) describe, Agile project management is becoming increasingly popular in software and R&D projects in general. However, while Agile methods enable more flexible work in arrangements and contracting types, in large new product development (NPD) organizations these methods cannot be necessarily sufficient and thus, a more complete view of agility is needed.

1.2 Problem Statement

Many, especially smaller, software companies have little expertise in law and contract arrangements. It is difficult for them to adjust contracting practices to the current methods they use. Especially, it is difficult for them to understand the advantages and disadvantages a specific practice can bring for different stakeholders (e.g. supplier and project owner). The existing literature in contracting practices using Agile methods is insufficient. This is mainly because it is based in empirical experiences from practitioners and that is the reason for a context to exist and help the companies to understand how to choose contracting practices. In order to close this gap, a better understanding of the perceptions of suppliers and project owners towards the contract types, their elements as well as advantages and disadvantages for both sides, is necessary.

Thus, the aim of this thesis is to survey the current state of knowledge on contracting practices by the companies in order to understand which practice is favored, by which stakeholder group and in which context. This could build ground for a concrete guide that lists new practices that implement Agile methods. Having a guide with contracting practices would be beneficial for practitioners and especially helpful for them to try to find the best contracting type in order to share the risks and the benefits between the supplier and the customer of the contract.

1.3 Research Question

This thesis will examine in more detail the state of knowledge of contracting law in IT companies. Additionally, it will try to get more insight in their current challenges and their perceptions towards specific contracting practices (e.g. fixed-price, T&M) and agile methods. Subsequently, the following research questions can be concluded from the above questions that mentioned.

- *What is the current state of knowledge on contracting practices in software development companies and their clients?*

- *What are the preferred contracting practices across buyers and suppliers?*

1.4 Outline of the thesis

The thesis is organized as follows.

In the second chapter there is a literature review of what is project management and software project management. Also, there is an elaboration on traditional methods of software project management such as waterfall and on Agile methods such as Scrum. Their main differences are also presented. Additionally, procurement is described. Furthermore, a presentation in detail of some relevant contracting practices is given.

The third chapter contains the methodology that is followed for the thesis. Moreover, how the questionnaire for the survey was constructed and designed and where it was distributed are explained in detail.

In the fourth chapter the results of the questionnaire are presented and analysed.

The last chapter contains the conclusion of the thesis and a proposition for future work.

2. LITERATURE REVIEW

In this chapter the fields that constitute the background of the research question will be presented. More specifically, a literature review of what is project management and software project management, as well as methods for the latter and contracting practices that have been developed so far will be briefly analyzed. Additionally, agility in software project management and how it can be compatible or adjusted to current contracting practices will be examined alongside with the advantages and disadvantages that offer.

For the literature study, papers that analyze contracting methods and contracting practices or present empirical experiences of the way some software companies are making contracts, were used.

2.1 Project Management

Project management traditionally defined as the process of planning, organizing, motivating and controlling resources, actions and rules in order to complete successfully specific goals and objectives within a specified period of time (Turner & Müller, 2003).

Particularly, this process is used in a project as a set of actions in order to produce an exclusive deliverable and also, it uses deadlines, starting and ending points to reach the target.

In order to meet the requirements of the deadlines and the budget for a project, effective management is needed. For more complex projects, management can become more challenging. However, there are a lot of benefits of effective project management. For example, delivering things on time and on budget gives predictability to similar projects.

Additionally, optimizing the development method is reducing the risk on the project and adds security.

Given these benefits, it is understandable why project management is applied to projects that vary from simple ones, such as small software projects, to more complex ones, such as major construction jobs (Luecke, 2004).

More specifically, project management methods started to be applied in the 1980s in manufacturing and software development fields and by the 1990s more tools and techniques were developed and adopted by different industries and organizations (Kwak & Anbari, 2005).

2.2 Software Project Management

In this part we will focus on software project management, its methods, their history and how they evolved so far.

Software project management is defined as "the art and science of planning and leading software projects. It is a sub-discipline of project management, in which software projects are planned, implemented, monitored and controlled" (Stellman & Greene, 2005).

In order to manage new development projects, companies applied the existent project management methods, but very often the deliverables were not on time and budget. This was mostly a result of confusion between the specifications and the target of the projects. Thus, to be able to avoid these problems, a model that is focusing on matching the requirements to the deliverables was presented and is known as the *Waterfall model* (Royce, 1970).

Additionally, on 2001 agile methods were introduced with the publish of the "Manifesto for Agile Software Development" which are making use of iterations in order to meet the project's goal more efficiently (Beck, 2001).

2.2.1 Differences between methods

In this part, before presented a more detailed elaboration about the software project management methods, the main differences between the traditional methods like Waterfall and the emerging agile methods such as Scrum will be presented in a more compact way.
The traditional software development methods are trying to reach the goal by optimizing the process. They are making use of the best presumed ways following a strict flow of steps in order to come to a pre specified result. The environment of these methods is solid and predictable and their main characteristics are control and order (Nerur & Balijepally, 2007).
On the other hand the new software development methodologies are trying to reach to the final goal through adaptation and flexibility. Making experiments and following iterations through teamwork and self-organization is helping the problem and its solution to be reformed. The environment of the methods is unpredictable and the main characteristics are communication and collaboration (Nerur & Balijepally, 2007).
In Figure 1 all the differences between the methods are summarized in a table.

	Traditional View of Design	Emergent Metaphor of Design
Design process	Deliberate and formal, linear sequence of steps, separate formulation and implementation, rule-driven	Emergent, iterative and exploratory, knowing and action inseparable, beyond formal rules
Goal	Optimization	Adaptation, flexibility, responsiveness
Problem-solving approach	Selection of best means to accomplish a given end through well-planned, formalized activities	Learning through experimentation and introspection, constantly reframing the problem and its solution
View of the environment	Stable, predictable	Turbulent, difficult to predict
Type of learning	Single-loop/adaptive	Double-loop/generative
Key characteristics	Control and direction Avoids conflict Formalizes innovation Manager is controller Design precedes implementation	Collaboration and communication – integrates weltanschauungs, or worldviews Embraces conflict and dialectics Encourages exploration and creativity and is opportunistic Manager is facilitator Design and implementation are inseparable and evolve iteratively
Rationality	Technical/functional	Substantial
Theoretical and/or philosophical roots	Logical positivism, scientific method	Action learning theory, Dewey's pragmatism, phenomenology

Figure 1: Traditional and emerging agile methods (Nerur & Balijepally, 2007).

2.2.2 Waterfall

In software development there are two essential steps, regardless the size or the complexity of the software project. Firstly, there is an *analysis step* and secondly, a *coding step* is following. This simple implementation concept is sufficient enough for small projects where the final product is going to be used by those who built it. In contrast of internal use, such an implementation plan for larger software systems will fail. Thus, many additional development steps are required.

Dr. Winston Royce (1970) described an approach to software development which is showed in Figure 2.

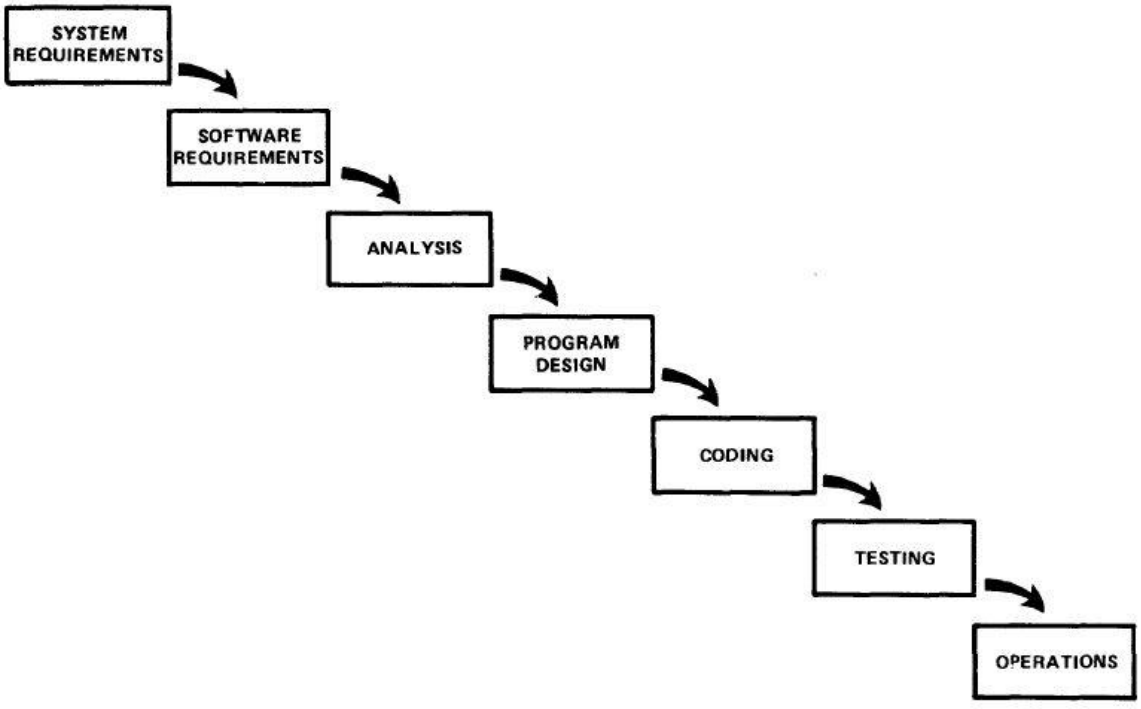

Figure 2: Implementation steps to develop a large computer program for delivery to a customer (Royce, 1970).

The analysis and coding steps are still present, but they are preceded by two steps of requirement analysis, they are separated by a program design step, and followed by a testing step. These steps are used separately from analysis and coding because of the different way they are executed. They must be planned and staffed differently for a more effective use of resources.

In the waterfall model if the testing phase, which occurs at the end of the development cycle, fail to satisfy the various external constraints, a major redesign is required. The design changes may be so disruptive that the software requirements will be violated. So, either the requirements must be modified, or a significant change in the design is required. As a result, the development process will return to the beginning and an overrun in schedule and costs can be expected.

Although this model is simple and customers can easily understand it, in reality, some additional features must be added to eliminate most of the development risk and thus the model is becoming more complex.

Step 1: Program design comes first

In this step, a basic program design phase has been introduced between the software requirements phase and the analysis phase. By this technique it is guaranteed that the software will not fail because of storage, timing, and data flux reasons, as can be shown in the testing phase. As the analysis proceeds in the succeeding phase, the program designer must specify clearly on the analyst the storage, timing, and operational constraints in a way that he can understand the consequences. In this way all the analysts and all the program designers will take be part of a substantial design process which will lead to a proper allocation of the resources. In order this procedure to take form, the following steps are required: (Royce, 1970)

1. Begin the design process with program designers
2. Design, define and allocate the data processing modes
3. Write an overview document that is understandable, informative and current.

Step 2: Document the design

During the testing phase, taking advantage of the documentation, the manager of the project can find more easily some mistakes in the program and as a result he can move the focus of the personnel to these and how to correct them.

During the operational phase, using effective documentation, the manager can use more specialized to the scope personnel in order to operate the program and to do a better and cheaper job.

Lastly, following initial operations, an efficient documentation can allow an effective redesign, updating and modification in the field (Royce, 1970).

Step 3: Do it twice

In this case a simulation which gives first results is included. With this simulation, questions of timing, storage, etc. can now be studied with more precision and experimental tests on some key assumptions can be performed.

Step 4: Plan, control and monitor testing

The goal of the previous three steps of designing the program before beginning, documenting it completely, and building a pilot model is to solve problems before entering the testing phase. However, even after following these steps there are still important things to be done like the following considerations: (Royce, 1970)

- Many parts of the test phase can be best assigned to test specialists who did not necessarily engaged in the original design.

- The analysis and code should be reviewed by a second party who did not participate in the original analysis or code and who could find simple errors.
- Every logic path in the computer program should be subjected to a numerical check at least once.

Step 5: Involve the customer

It is very important for the customer to be involved in the project in a formal way in order to have participation in earlier points before delivery (Royce, 1970).

In Figure 3 there is a summary of the five steps that are implemented in the original waterfall model. The simpler model without these steps is difficult to work on large software development projects and the costs to recover may exceed those required to finance the five steps.

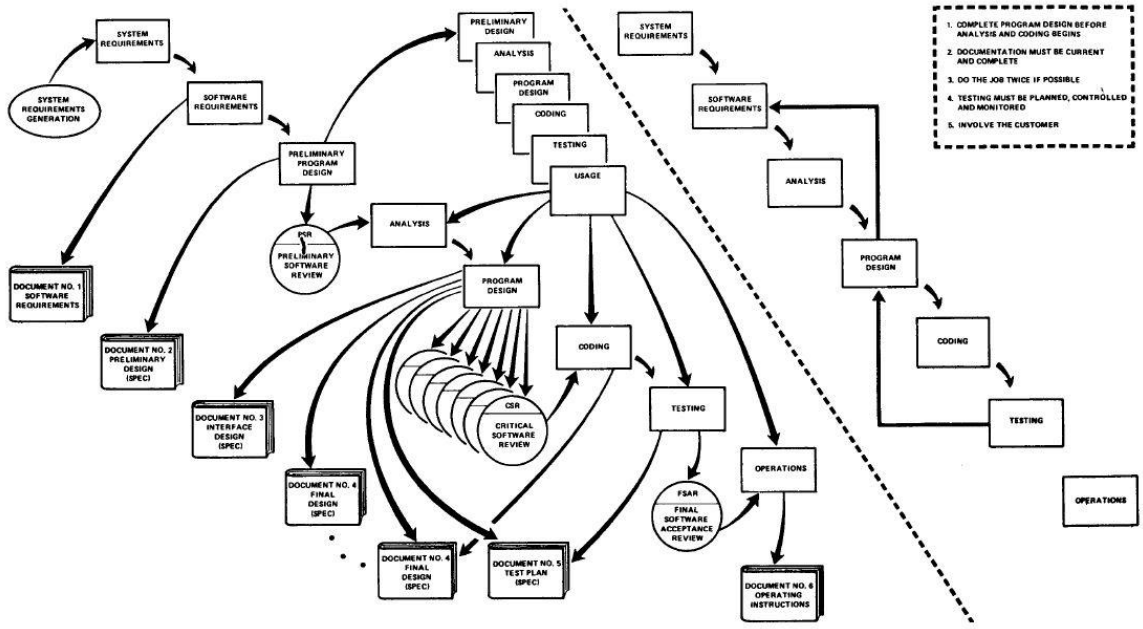

Figure 3: Summary (Royce, 1970)

2.2.3 Agile Methods

The traditional software development methods in 70s were following a strict schedule which was relied on requirements documentation and plan based testing. There was the feeling that sticking with the plan will keep the trouble away of the projects, but this was seldom being succeeded.

Simultaneously, some methodologies that were making use of iterations, started to being used from independent consultants, who managed to recover projects from trouble. These set of methodologies led to Agile software development (Williams, 2012).

Agile software development is not a simple methodology, but a group of software development methods which is based in iterative and incremental development. It differentiates itself from methods that are making an extensive use of documentation, such as waterfall and relies on self-organizing and cross-functional teams. In 2001 the "Manifesto for Agile Software Development" was published and since then it has a significant acceptance in project development (Beck, 2001). Results of a survey conducted in March 2006 by Scott Ambler between 4200 respondents (Ambler, 2006), (Szalvay, 2004) showed that 41% of development projects have adopted agile methodology, and agile techniques were being used on 65% of such projects. In Figure 4 these early adoptions are presented. A more recent survey and its results in a more detailed way are presented in section 2.2.6 after the elaboration on agile methods.

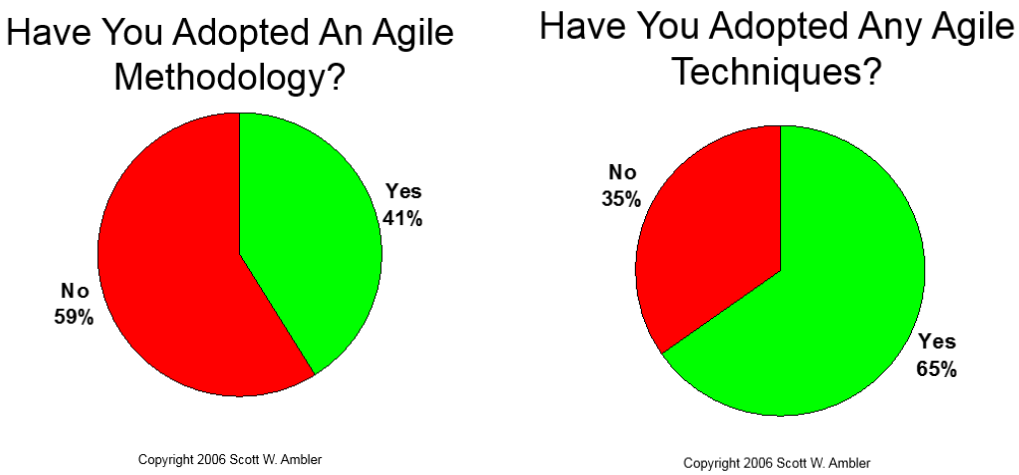

Figure 4: Adoption in Agile Methodology and Techniques (Ambler, 2006).

In Figure 5 the "Manifesto for Agile Software Development" is shown

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Figure 5: The Agile software development manifesto (Beck, 2001).

The "Agile Manifesto" is based on twelve principles according to Beck, Kent et al. (2001)

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstances

The supporters of Agile software development argue that the waterfall model is not efficient in practice, and it is difficult for complex projects to finish a phase of a software product's lifecycle perfectly before moving to the next phases and have feedback from them (Dischave, 2012). For example, clients may not know exactly the requirements needed before reviewing a working prototype. They may change their requirements constantly. If this happens after the design is finalized, the design must be modified to meet the new requirements. This means that more working hours are spent, also the costs are increasing, especially if a large amount of the project's resources has already been invested.

One of the most important differences between the agile and waterfall approaches is that waterfall has distinct phases with checkpoints and deliverables at each phase, while agile methods have iterations rather than phases. The output of each iteration can be used to

evaluate and change or evolve the requirements. Waterfall assumes that it is possible to have perfect understanding of the requirements from the start. But in software development, clients often do not know what they want and cannot express clearly their requirements (Stellman & Greene, 2005).

Two of the main Agile methodologies are XP and Scrum. They differ in particulars but share the iterative approach that is shown in Figure 6. In the following parts a brief presentation of these methods will follow.

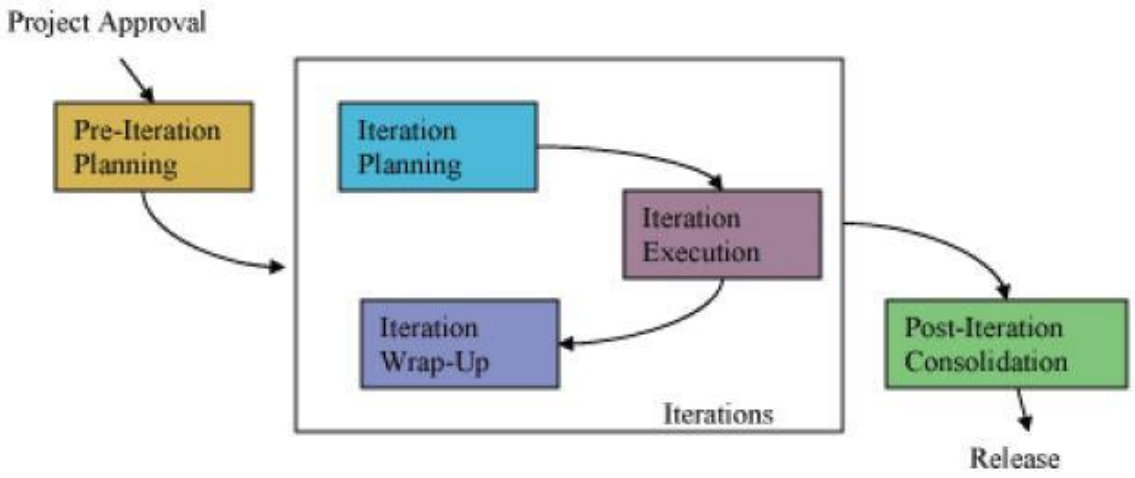

Figure 6: A general agile development process features an initial planning stage, rapid repeats of the iteration stage, and a form of consolidation before release (Szalvay, 2004).

2.2.4 Extreme Programming (XP)

In the 1990s there were two important factors that had a huge impact in software development. Procedural programming was replaced by object-oriented programming and also, the internet boost and the dot-com boom formed company growth and speed-to-market as competitive business factors. As a result, shorter product-life-cycles were

required and that was inconsistent with the traditional methods of software development (Larman & Basili, 2003).

XP programming met those requirements by making use of frequent releases in short development cycles, which improved the productivity.
XP projects start with a release planning phase, followed by several iterations, each of which ends with user acceptance testing. When the product meets the user's requirements, the team ends the iteration and releases the software. Using these "checkpoints" it is easier to adopt new requirements by the user ("Extreme programming," n.d.).

Two important features of XP programming are "user stories" and "pair programming".
User stories describe the problems that need to be solved by the system that is being built. They are written by the user and generally are only three sentences long.
In pair programming two people are writing code at the same computer reviewing each other. Although working alone could meet the requirements of the user stories, this way gives higher quality in the results (Szalvay, 2004).

In Figure 7 there is a summary of the steps of XP.

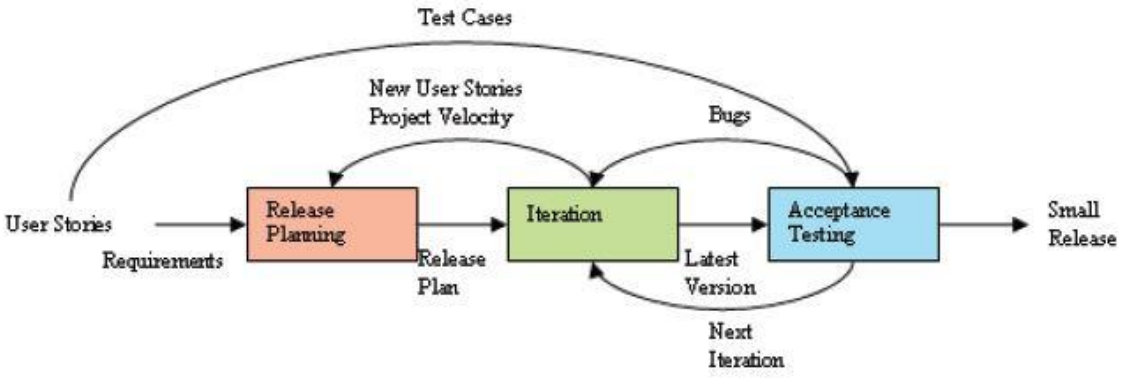

Figure 7: A simplified XP process (Szalvay, 2004).

2.2.5 Scrum

Scrum methodology is one of the most used Agile methods. It takes its name from rugby because of the use of Scrumdowns. The main characteristic of Scrum methodology is the use of Sprints. The Sprint is a "time-boxed" phase. It has a specific duration which is normally between one week and one month, although the most common that is used is two weeks.

Scrum is an empirical approach, which considers that the analysis, design, and development processes in the Sprint are unpredictable. This is the main difference against defined approaches such as waterfall. Particularly, the key principle of Scrum is the acceptance that during a project the clients can change the requirements and the target, and that unpredicted challenges cannot be treated easily using defined approaches (Schwaber, 1997).

There are three core roles and a range of subordinate roles. Core roles are the most committed to the project, while the other roles are only involved. Thus, core roles are often referred as "pigs" and subordinate roles as "chickens", after the story of "The chicken and the pig".

Core roles are:

- **The product owner**

  The Product Owner is the client. He is the one responsible for setting the requirements and the targets like in XP. Additionally, he writes in the user stories, he prioritizes them, and adds them to the product backlog.

- **The development team**

  The development team is responsible for the deliverables in each sprint. Usually, the team is made of 3-9 people that have cross-functional skills and performing tasks like analyzing, designing, developing and testing. Also the team is working in a self – organizing way.

The Scrum master

The Scrum Master, is responsible for removing any obstacles during the process in order to help the team reach the goals. He is different from the traditional project manager. He is more a coordinator for the team because he ensures that the Scrum process is used as intended, he arranges frequent meetings, and challenges the team to improve (Schwaber, 1997).

Two important characteristics of Scrum methodology are the ***backlogs*** and ***sprints***.

Backlog is a list of requirements that the final product have to meet. It is similar as the user stories in XP and the one responsible for it is the product owner. It is divided in the product backlog and the sprint backlog. In the first, the requirements for the deliverable product usually include risk predictability, business value and dates for the deliverables. The latter, includes the top tasks from the product backlog that are extracted and performed in each sprint. The one responsible for the sprint backlog is the Scrum team.

Sprint is the basic unit of Scrum methodology. It is a set of tasks that are performed during a specific period, who's duration is usually one to four weeks. The tasks are extracted from the sprint backlog. Each Sprint consists of one or more teams performing the following: (Schwaber, 1997)

Develop. The team defines the changes that are needed for the implementation of backlog requirements into packets. Using these packets, the team analyses, develops, tests and documents the changes.

Wrap. The team is closing the packets and wrapping up the way they implemented the backlog requirements.

Review. The teams are meeting in order to present their work and review their progress.

Adjust. The team is making new packets including possible changes on the requirements, as a result of the review.

Finally, the advantages of Scrum methodologies are followed and the Scrum process is depicted in Figure 8.

Traditional development methodologies are predefined and their only way to predict possible changes is on the start of the development cycle. Thus, their ability to respond to changing requirements is limited, once the project has started.
The Scrum methodology, on the other hand, is more flexible. With the use of sprints the requirements can be redefined after the development of the project has started. By this way, organizations can change the project and deliverables at any point in time without creating a bad working environment for the development teams and also, they can deliver products with higher quality (Schwaber, 1997), ("Scrum (software development)," n.d.).

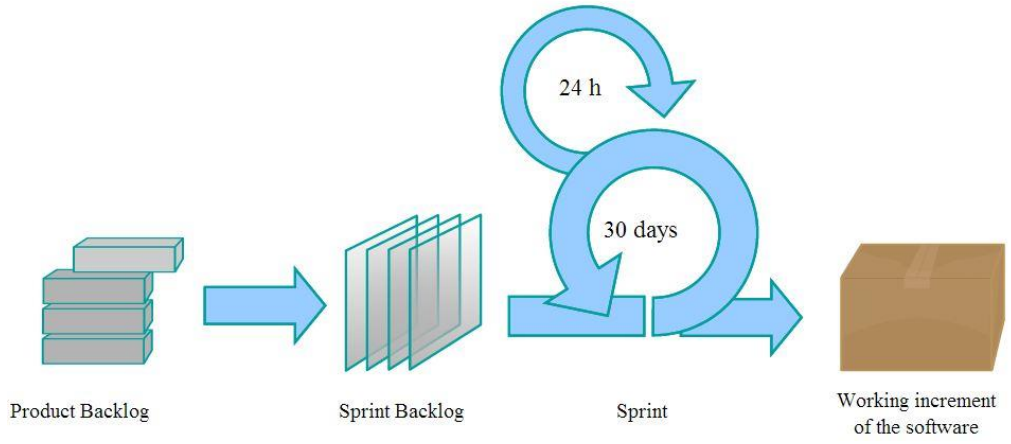

Figure 8: The SCRUM process (Stellman & Greene, 2005).

2.2.6 Agile adoption

In section 2.2.3 there were presented the results of an early adoption of agile methods based on a survey that conducted in 2006. Since then more and more people are realizing how beneficial to business can agile development be. Results of a survey that was conducted in

2013 (VersionOne, 2014) are showing that there is an increase of 11% on 2012 and 2013 on the people who suggest that agility is helpful for the faster completion of projects.

More specifically, this survey (VersionOne, 2014) conducted between August and October of 2013 among 3,501 respondents from the software development community. The majority of respondents was from North America (66%) and Europe (20%).

About 88% of respondents said they have knowledge on agile practices and the increase from 2012 was 7%. The same percentage of respondents replied that their organization was practicing agile development.

In Figure 9: Agile Methods in use (VersionOne, 2014)Figure 9 among the agile methods that are being used from the respondents is can be clearly seen that the most popular method is Scrum with 55%.

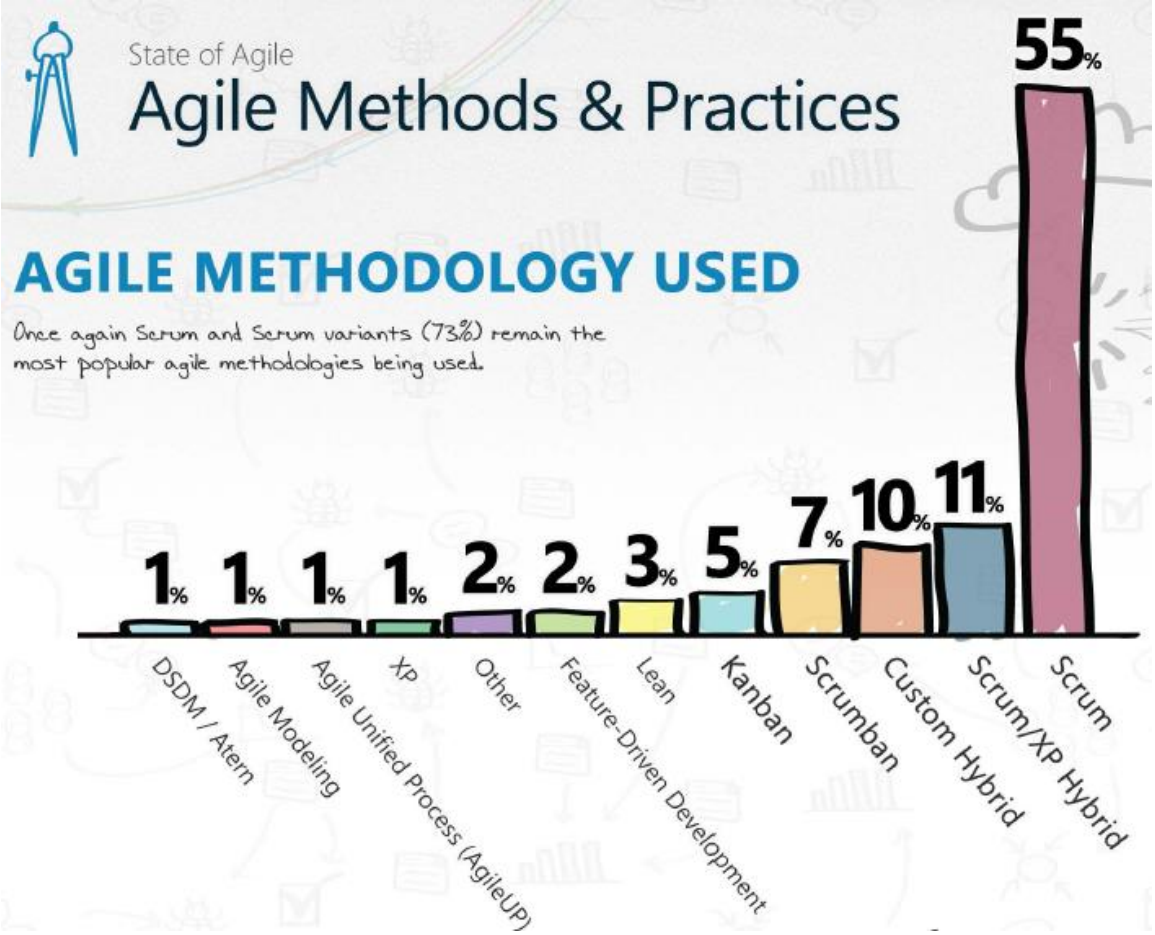

Figure 9: Agile Methods in use (VersionOne, 2014)

Another important result of the survey is about the failure of agile methods. As it is shown in Figure 10 15% of the respondents did not have any failure on their agile projects and from the respondents that had some failure it can be seen that the leading causes were the opposing company's philosophy or other form of cultural resistance.

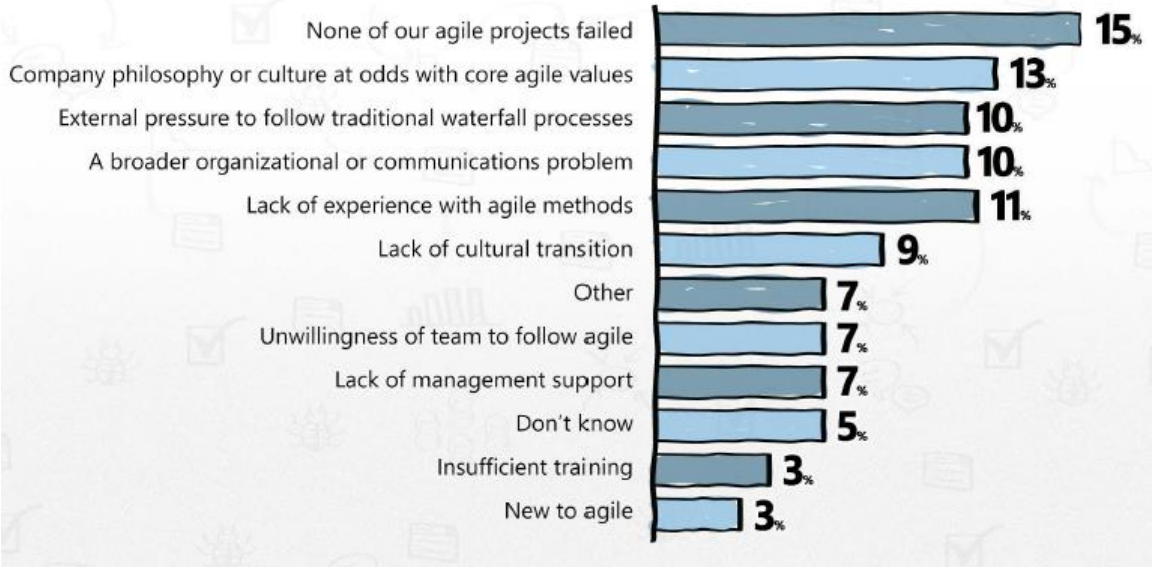

Figure 10: Leading causes of failed agile projects (VersionOne, 2014)

On the other hand, 73% of the respondents replied that agile projects required less time to complete than previous non agile ones and lastly, regarding the benefits from adopting agile methodologies, the most important were the ability to manage changing priorities, the increased productivity and project visibility and the improved team morale. In Figure 11 these benefits are presented in a more detailed diagram.

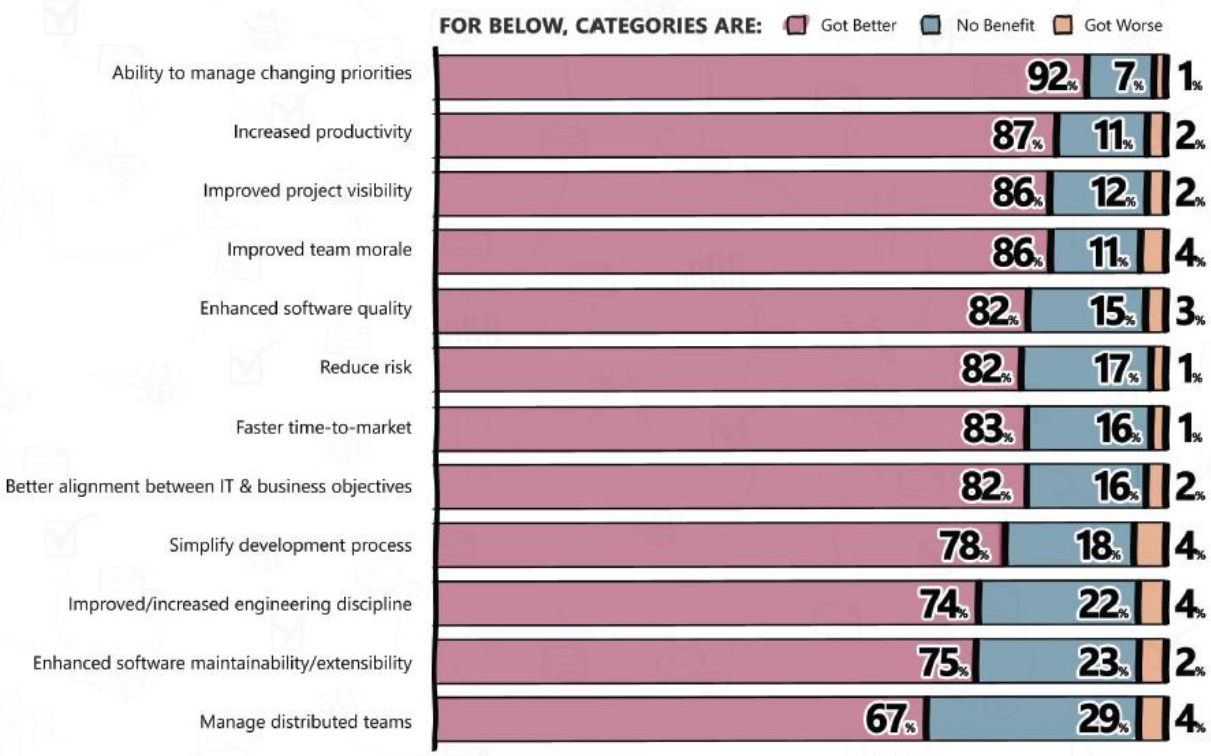

Figure 11: Improvement from adopting agile methodologies.

2.3 Procurement

Kerzner (2009) defines procurement as the action of acquiring goods or services. Procurement like contracting, which will be analyzed in the next part is a process where two sides with different goals are collaborating in order to make profit. Good procurement practices can provide an increase in the profit of an organization.

Because procurement is related to the profitability, there are standardized practices which contain frameworks in order for an organization to meet its objectives. Two basic procurement strategies are:

- Corporate Procurement
- Project Procurement

Corporate procurement is the procurement that is based on the organization's strategy. Centralized procurement is an example.

Project procurement is the procurement which is related to a specific project's environment. For example a project manager can purchase sources without involving the centralized procurement group of the organization.

Procurement is a process which leads to contracting. More specifically, the target of performing a procurement process is to negotiate a contract which will give the contractor or supplier some risk on his shoulder but on the other hand it will provide him with a strong incentive for an effective performance (Kerzner, 2009)

2.4 Project Contracting Methods

Contracts are forming in a legal way the steps that will be followed in a project. Specifically, contracts are setting the rules for the project and are binding for their sides (supplier and customer) (Stevens, 2009b).

In principle, the rules can be entered freely by both sides trying to make an ideal environment, in order for the project to be completed successfully. Practically, though, the sides of the contract are competitive to each other. They are trying to make the rules beneficial for themselves and to shift the risk on the shoulders of the other party, especially for the occasion that the project reaches an undesired situation. For that reason, the negotiations for important and expensive contracts should be done by professionals.

In Agile Manifesto is stated that customer collaboration is more important than contract negotiation (Beck, 2001).

This is correct because contracts allocate the risk and establish trust between the contractual sides. Miscalculated rules can be harmful for the success of the project. They can lead to budget extensions or to a lower quality than the expected, regarding the final outcome of the project (Stevens, 2009b).

In the next part some generic contract types are analyzed, followed by agile contract frameworks. Particularly, it is shown from empirical examples how agility can enter in the generic contract types in order to result in more effective contracting practices.

2.4.1 Related Framework

In (Zijdemans & Stettina, 2014) there was presented a preliminary framework for understanding contracting practices and how certain practices are affected in project contexts.

More specifically, some concrete practices in use are grouped into four categories. These practices and categories are based on the writers' discussions with participants in a survey and also on contract elements in literature. The four categories that are also shown in Figure 12 are:

- Contract Basis
- Incentive element
- Uncertainty Mitigation element
- Governance element

The contract basis comprises agreements that are related to basic elements such as the pricing model, the delivery date and the scope of the project. But, these basic agreements are insufficient when applying to many projects and thus, variations and additions have been applied in order to make them more sufficient (Zijdemans & Stettina, 2014).

The incentive element contains practices that are related to the unfairly shared risk in many cases of software development. Hence, in these practices there is an additional incentive for one side of the agreement in order the risk sharing will be more balanced.

The uncertainty mitigation comprises arrangements that are focusing on reducing the uncertainty which relates to the price, the scope or the deadline of the project.

The governance is focusing on the collaboration between the two sides of the project. This is relating to cases that the client's involvement needs to be increased and the contract to be more efficient.

Moreover, there is a distinction in affecting factors that the authors found as more important from their discussions and research. Hence, factors such as project's budget or quality, government and low trust between the two sides of the project, are distinguished in the following categories:

- Project Content
- Institutional Content
- People

Lastly, their framework is showing for each contracting practice based on each affecting factor, who benefits more, the project owner or the supplier.

	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	◐	○	●	●	●	●	●	●	●

Figure 12: Preliminary framework for understanding contracting practices as affected in project contexts. Practice contributes benefits predominantly for: Project Owner (◐), Supplier (◑), both (●), none (○) (Zijdemans & Stettina, 2014)

2.4.2 Fixed-Price-Fixed-Scope Contracts

In Fixed-price-fixed-scope contracts the budget of the project is prearranged. Moreover, there is a fixed deadline and the specifications of the project are predefined on the contract. Many companies like the idea of making contracts in this way because they think that the risk is lowered. Thus, there are a lot of customers that demand these kind of contracts. This is because they feel more secure by having more control over the project's schedule, cost and outcomes (Hoda, Noble, & Marshall, 2009).

On the other hand, suppliers need variations since, these types of contracts comprise tight margins and their profit cannot be increased. Moreover, the compensation for the supplier is limited and there is no safety for the risk. Because of these there is a lack of motivation for the supplier to meet the client's targets and they are willing to make the minimum effort for the minimum cost (Turner, 2003).

There are a lot of reasons that make these practices ineffective for both the supplier and the customer as presented in more detail by Venkatesh, Cherurveettil, & Post (n.d.).

- Firstly, the customer has the challenging task of specifying the scope of the software project before the project starts.
- Moreover, if the scope is not specified clearly from the customer there is a risk that the delivery product will not meet the requirements of what the customer originally wanted.
- Additionally, if the supplier will not choose a team correctly, then the customer could end up with an insufficient team that will deliver a product with worse quality than it was expected.

- Having an ineffective team could also lead to schedule extensions and overruns and as a result the supplier may end up paying more than he expected at the beginning of the project.
- When the scope is ambiguous from the start, the price could also exceed the one that was firstly expected and this risk is on the shoulders of the supplier.

In Figure 13 a plot depicts the profit and the revenue according to the effort of the supplier and the money spent. As it can be seen, the revenue is constant regardless the effort applied for the project. On the other hand, more effort decreases the profit for the supplier.

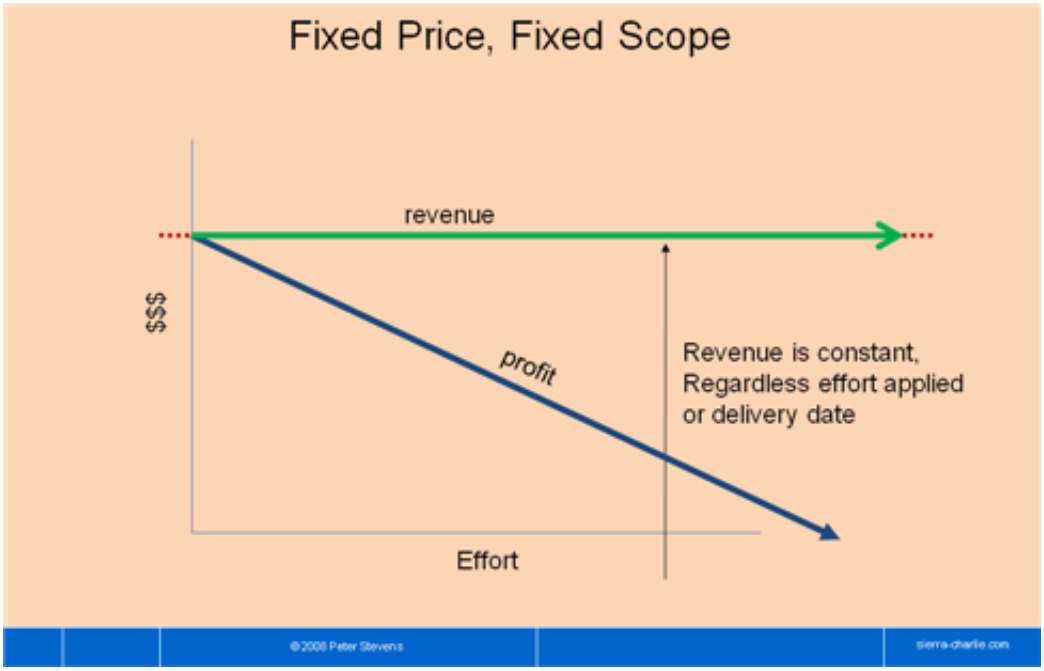

Figure 13: Fixed-Price-Fixed-Scope Contract Plot (Stevens, 2009a).

2.4.3 Time & Material Contracts

This type of contracts is different from the fixed-price contracts. The supplier's fee is based on its hourly rate and the compensation for the materials required for the development of

the project. Generally, after working for a certain period, for example one month, he sends the customer the bill (Stevens, 2009a).

The scope of the project is not fixed from the beginning of the project and the customer can end it whenever he wants. Although, the structure of the contract appears to be fair because the payment correlates exactly with the work that has be done, there is a lot of risk that lays on the shoulders of the customer (Book, Gruhn, & Striemer, 2012).

This is due to the fact that sometimes the supplier has no incentive to be efficient and also, can neglect quality control of the project. All these can lead to a waste of time and money for the customer because he cannot control if the supplier is using his time as productively as he could. The only possible incentive for the supplier is the high competition and the possibility to gain more contracts from the customer. Additionally, it can be hard for the customer to predict the budget of the project.

In Figure 14 the plot presents the structure of the contract. As it can be clearly seen, the revenue and the profit are correlated to the effort being done. When the effort increases, the profit and the revenue are also being increased.

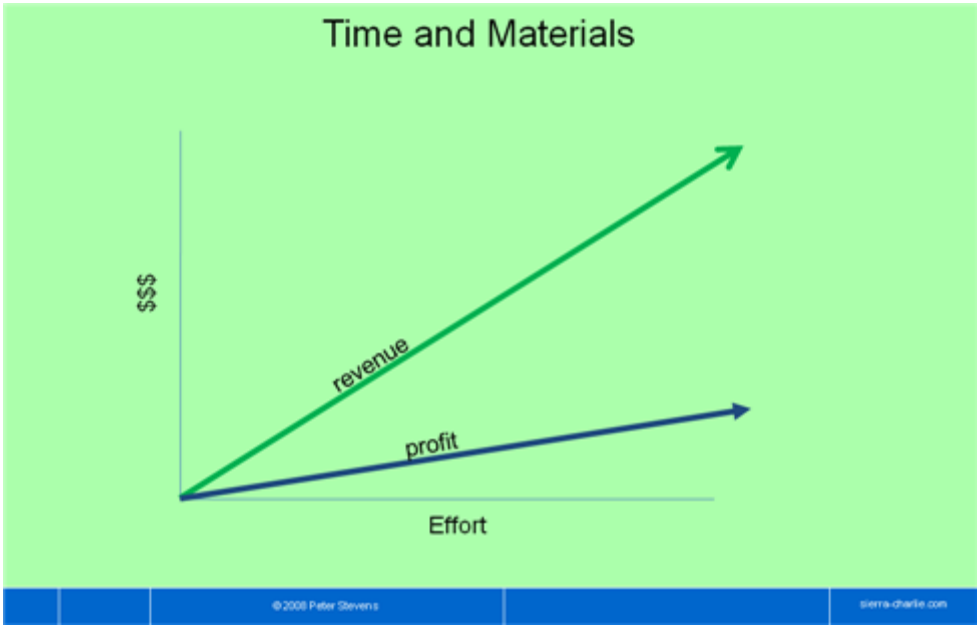

Figure 14: Time & Material Contract Plot (Stevens, 2009a).

According to Stevens (2009a) there are also two differentiations of the Time and Material contracts. These are:

- Time and Material with fixed scope and a cost ceiling
- Time and Material with variable scope and cost ceiling

In the first type of the differentiations the scope is predefined and there is also a cost ceiling. There is no incentive in the side of the supplier to finish the project earlier because this means that he will gain less profit. So, there is a motivation to extend the project's duration to his interest. However, the risk is shared between the two sides because of the existence of the cost ceiling which acts as a safety mechanism for the customer. On the other hand, the cost ceiling creates a risk for both sides if it is reached and the requirements have not yet been met because the project has to be continued until the completion of them.

The way the revenue and the profit are behaving according to the occurred effort is depicted in Figure 15.

In the second differentiation the contract is structured in the same way that Time and Material contracts are. That means that the scope can be altered. However, there is a difference in the budget. Like in the previous type, there is a cost ceiling that prevents the risk of a potential overrun on the budget but on the contrary, there is the risk for the customer of not getting what he asked for when reaching the cost ceiling because the project stops when the maximum budget is reached.

In Figure 16 the association of the revenue and profit according to effort is presented.

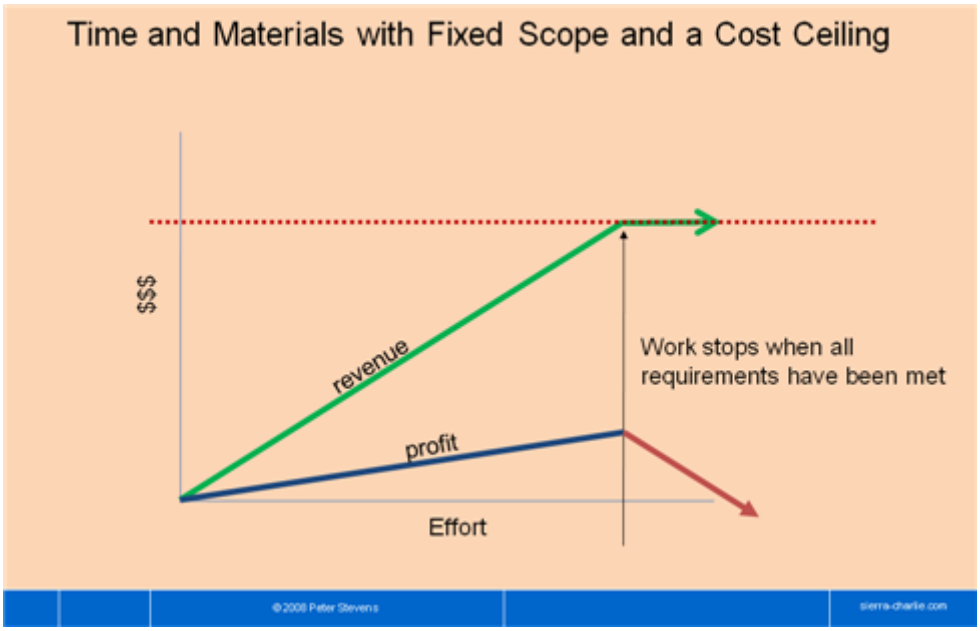

Figure 15: Time & Material with fixed scope and a cost ceiling Plot (Stevens, 2009a).

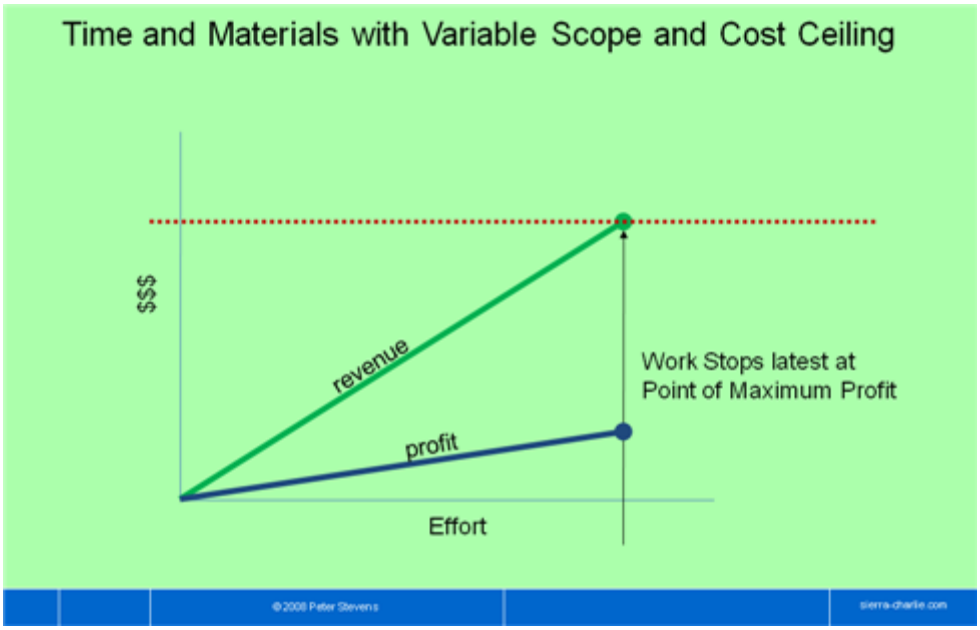

Figure 16: Time & Material with variable scope and a cost ceiling Plot (Stevens, 2009a).

2.4.4 Cost Plus Contracts

In this type of contracts the supplier is reimbursed for all his expenses plus an additional payment which contains his profit (Hofbauer & Sanders, 2008). This is the reason why cost plus are also called cost reimbursement contracts.

There are three main variations of cost plus contracts that are giving different incentives to the suppliers (Kerzner, 2009).

- **Cost plus fixed fee contracts.** A prearranged profit fee is provided to the supplier.
- **Cost plus incentive fee contracts.** A fee related to performance that exceeds the projects targets is provided to the supplier.
- **Cost plus award fee contracts.** A fee related to the quality of the end product is provided to the supplier.

More specifically, the characteristics of the variations of cost plus contracts are being described (Kerzner, 2009).

The cost plus fixed fee contracts are used traditionally when there is the belief that this is the only way to manage an accurate pricing for the project. In these contracts the cost may vary but the supplier's fee is pre negotiated and stable. By this way there are not any sufficient incentives for the supplier to reduce the project's costs since he is getting the same fee in any case. Thus, the project could end with cost overruns at the expense of the client.

The cost plus incentive fee contracts are similar to the fixed fee contracts with a main difference that is spotted on the way the fee that comprises the profit for the supplier is calculated. Particularly, instead of a stable fee the profit is based on a specific formula that compares the final cost with the initially expected cost of the project. This formula is pre agreed by both the supplier and the client. Hence, the supplier has a strong incentive of keeping the costs low. These contracts are mostly used for long duration and R&D projects.

The cost plus award fee contracts are similar to the incentive fee contracts. Their main difference is that the fee is more subjectively determined compared to the objectively

calculation that takes place in the incentive fee contracts. Award fees are determined by the quality of the end product, the levels of performance and the duration of the project. Generally, they are determined by the whole effort that the supplier puts in the contract and that consists an incentive for the supplier to make his best effort.

2.4.5 Target/Cost Price Contracts

In this type of contracts the risk is shared between the supplier and the customer. The contract determines a target of effort in working hours, a negotiable profit and sometimes, a deadline (Moløkken-østvold & Furulund, 2007).

The payments to the supplier are made on the basis of the supplier's accounts and records, which are provided to the customer for inspection on an open book basis. At the end of the project, the final target cost which is the original target cost plus any agreed changes, is compared to the actual cost expended by the supplier. If the actual cost is lower than the target cost, a saving has been made. Then it is shared between the parties on a pre-agreed percentage basis, which is usually 50%. And vice versa, if it is higher, the extra costs are also shared in the same way.

In Eckfeldt, Madden, & Horowitz, (2005) there is a description about their experiences working with a target cost model. More specifically, it is described how this approach was successfully used with a startup company and how their client relationship and collaboration with the development team was improved.

Prior to this they had two target-cost experiences. In the first one, in order to give incentive to both parties, extra or less days of the predefined time schedule were discounted 50%. For example, if the supplier finished 2 days earlier, the customer could only add one day of work. Respectively, if the supplier needed 3 more days to finish the project, the customer only had to remove 1.5 days of work. Their second experience worked exceptionally well as the project was completed within 5% of the initial target-cost budget and the client was very satisfied.

Their main experience that is described in the paper is showing why clients are choosing target-cost contracts. The client of the specific startup company liked the model for three reasons.

- The developers shown their determination to the project by putting their profit at risk.
- He could reduce the cost by finding ways to simplify the complexity of the project without reducing the quality.
- He was feeling that the developers did not have any incentive to extend the project for a bigger profit.

Most importantly, the client was feeling that he was in the same goal path with the developers and this creates a trusted relationship and collaboration.

By using their own time tracking system, the development team could monitor the actual time that was spent and compare it to the estimated one. The review for the hours spent was taking place after each iteration, usually weekly, with the client. This mechanism can be perceived as a very important tool that makes both client and development team to consider it before discussing potential changes on the project. Hence, both sides were considering the time and the budget effect on scope decisions, and also the balance between the need and the cost.

The changes in the scope were categorized into *Fixes, Clarifications* and *Enhancements.* For the first two there was an extra billing for the additional hours against the predefined. On the other hand, enhancements were treated as an increase of the project's scope and as a result, an increase on the profit.

Based on these experiences, the writers of the paper came to some interesting conclusions for using target-cost contracts.

The size of the project is affecting its sensitivity to changes and risk. Smaller projects are more sensitive while larger ones can better absorb them. The success of the project is dependent on the client's risk tolerance. In cases that the client was more comfortable in taking risks, he was more willing to make changes in the project. Moreover, the length of the developers' releases can affect the project. In the writers' experience with the startup company, the developers structured the project to be in several smaller releases instead of larger ones. This gave them the ability to make clean breaks between the releases and thus

avoid extra scope changes. Lastly, by tracking time, the client can observe the real status of the project and as a result a better relationship between the two sides can be built.

Overall, target-cost contracts can be a promising alternative to fixed price and time and material contracts because they allow risk sharing between clients and developers and make the projects successful.

According to Stevens (2009a) the main characteristics of target-cost contracts are that the scope of the contract can be changed. There is flexibility, and as a result, planned features can be replaced by others, but as mentioned before, additional features will cost extra and the costs are split by both parties.

All these lead to a shared risk between the supplier and the customer, because both sides have a common interest in completing the project early. The customer will have less costs and the supplier will gain more profit.

Figure 17 depicts a plot that shows the revenue and the profit according to the effort is done for the project and to the time it finishes.

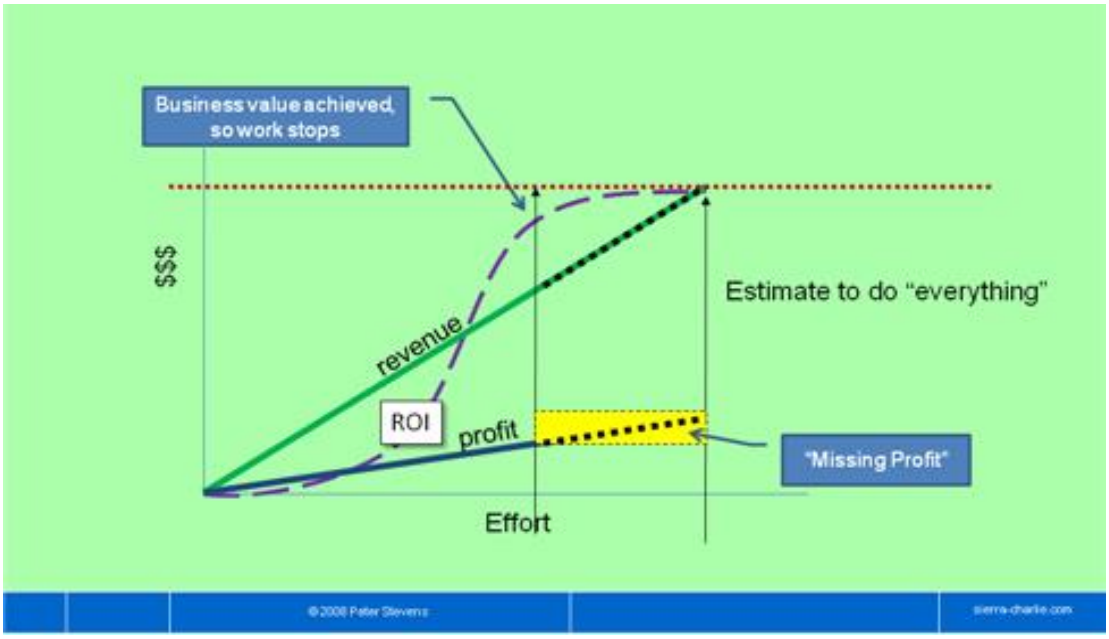

Figure 17: Target Price Contract Plot (Stevens, 2009a).

2.4.6 Pay per Sprint Contracts

According to Stevens (2009a), sprint contracts are not actual contracts. They are an arrangement between a customer and a supplier for one sprint.

Some suppliers offer this kind of agreements because they act like a teaser for the customers in using agile methods for their projects. As a result, by making these agreements, the suppliers offer agility to customers in a trial basis which leads to building trust between the two sides and giving confidence and risk coverage to hesitant customers (Hoda et al., 2009).

There is also a strong incentive for the suppliers because once the customers have tried one or more sprints, they have the option to buy more.

The scope for each sprint is usually consisting of an agreed set of features. However, there is flexibility for the customer to change the general scope when buying more sprints.

The risk is low for both sides because the most that can be lost in the meaning of work effort and costs, is just one sprint. This is the case where the customer will be unsatisfied and there will be no agreement for more sprints.

In Figure 18 the factors of this type of contract are displayed. The quality that will be delivered is affected by the time of the sprint and the scope is connected to the cost of the sprint.

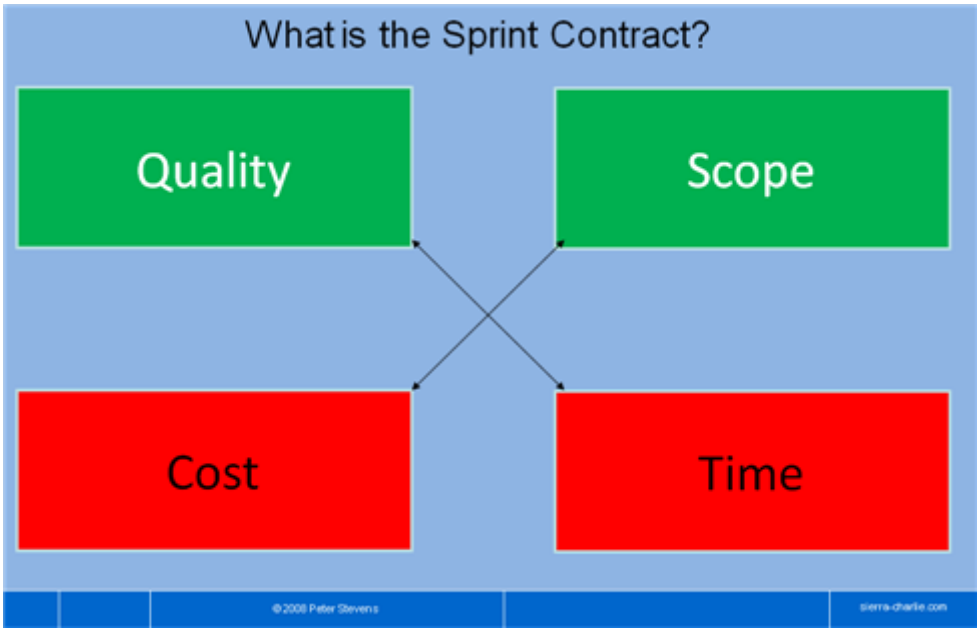

Figure 18: Pay per Sprint (Stevens, 2009a).

2.4.7 Agile-Fixed-Price-Contracts

Agile-Fixed-Price-Contracts are fixed price contracts where agile characteristics are implementing in order more room to be given for scope changes.

All the drawbacks of the fixed-price fixed-scope contracting practice are showing why agility can make this practice more efficient in order the delivery product to be of high quality and also, the risk to be shared between the contractor and the supplier. However, fixed scope and agility are two contradicting terms (Venkatesh et al., n.d.).

A solution in implementing agile characteristics is presented by Venkatesh et al. (n.d.). They are making use of the key agile elements such as user stories and backlog.

The important part for agility in fixed price contracts as it is recommended by the authors of the paper is the adoption of an 80-20 rule. 80% of the scope would be the must have features and 20% would be the nice to have features. The 20% comprises a stretched goal which can be achieved if the supplier has completed earlier the 80%.

By this way, more room is given for agility in the fixed-price-fixed-scope contracts and at the same time its security is maintained. Moreover, the supplier has incentive to complete the must have features earlier in order to increase his profits. Another important part of the contract is the existence of flexibility to add new features if some from the must have features is given up. However in that case the presence of change control is important. The success of the project is affected of the trust between the supplier and the client and on their ability to manage change control process. This process, also, needs to be clearly defined in the contract.

Additionally, there are some other features that could be used and perceived as success factors for the model apart from only the 80-20 rule and the change control process.

- The scope should be clearly defined in sprint backlog
- The backlog could be broken into releases
- The first release could be used as a baseline

Implementing the model of Venkatesh et al.(n.d.) there could be an agile fixed-price-fixed-scope contract where the scope and the price could be left as a variable. Although this sounds contradicting, it could result in better collaboration between the customer and the supplier, in the sharing of risk between the two sides, and most important in products of better quality.

2.4.8 Collaborative Agile Contracts

Collaborative agile contracts comprise a contracting model developed by Thorup & Jensen (2009) and its main feature is that the payment is delayed until a specific criterion is met.
In the paper of Thorup & Jensen (2009) the authors are describing their practical experience from a consulting company from Denmark that used to provide assistance for development of projects using time & material contracts. After they realized the disadvantages of this type of contracts, they decided to work with a collaborative agile setup. In order to achieve this, they had to make a contract with the following structure.

The supplier can be able to develop the solution of the project iteratively. The development quality can be kept in a high level and also, the customer and the supplier can have an incentive to work with collaboration and to finish the project on time and according the specified functionality.

By this way, the risk will be shared fairly and more room for adjustments will be available during development, which will be based on feedback. Due to their experience, they created a model that comes to this result. They called their model Collaborative agile contract and its main mechanism is the postponement of the payment until a certain criterion is reached. Additionally, the calendar date is not being used as this criteria and also, the criteria can be described as a situation where the customer is gaining value from the software project. Finally, there is a common interest on reaching the above situation for both sides of the contract. The customer will reach his goal and the supplier will be rewarded for his efficiency.

In more details, the method contains the following characteristics. The scope is described loosely in a form that is similar to user stories which mentioned in the agile methods. Also, the price for the hourly rate is 10% to 50% lower than the one in time and material contracts. There is a set of milestones similar to sprints where each one of these are leading to payment. The milestones can be reached when the customer is able to use and deploy the software for which the specific milestone is dedicated. Moreover, there is a suggested time frame for the overall project and for each milestone and generally, the development process is following the agile practices. An important aspect of the method is that there is no detailed requirement specification, no fines and the deadline is not fixed. As a result there is more space for agility and collaborative behavior between the customer and the supplier.

2.4.9 Two Phased Model Contracts

This contract model is actually formed by the using of two different contracting methods in certain phases. For example for the first months of the contract a fixed-price-fixed-scope

method could be used and for the remaining period, Time and Material method could be applied (Arbogast, Larman, & Vodde, 2012).

According to Arbogast et al., (2012) the incentives in order this method to be used are the lack of trust, a regulatory constraint, a need to define high level requirements and also the optimizing of a secondary goal.

Specifically, in the beginning of a project there is a lot of uncertainty regarding the scope, the budget and the duration of the project. Additionally, sometimes the trust between the customer and the supplier is low if they have not cooperated before. For instance, a fixed-price-fixed-date contract for a period of one year could seem risky and could treated with hesitance. Applying the two phase model, customers can limit their risk (Arbogast et al., 2012).

The first phase is a short phase that is targeting to build trust between the two sides of the project and also to come through the initial uncertainty. Moreover, the scope can be defined more accurately, the way of working and the productivity of the supplier can be observed, and the budget and time can be estimated more precisely. After these uncertainties have been surpassed, then the two sides of the project can negotiate for another contract method to be followed in the next phase (Zijdemans & Stettina, 2014).

2.4.10 Exit Arrangement

Exit arrangement is not a contract type. It is actually a way of terminating a contract in a controlled manner. Specifically, exit arrangements are exit points in time where the involved sides of the contract can abort the project in a disciplined way (Zijdemans & Stettina, 2014).

Moreover, these arrangements can be applied in different contract types and especially in those containing ambiguous requirements. By this way more safety is added for both sides of the project.

According to Wedutenko & Watson (2012) the best time to make these exit arrangements is when the initial contract agreement is taken place. Hence, for the customer side, when

the project will be terminated, there will not be any difficulties to switch the outsourced services to another supplier.

2.4.11 Risk Buffer

Risk buffer is not a contracting type. It is a practice often used by the suppliers in order to reduce the uncertainty associated with the assessments that have to be made in fixed-price contracts. The use of this practice was mentioned during the interviews that conducted by Zijdemans & Stettina, (2014). According to those interviews, the buffer is related with the budget and the project duration and can be used by the supplier in order to cope with unexpected change request in the last stages of the project. The customer does not need to know about the using of risk buffer.

3. METHODOLOGY

This study will be based on the existing paper of Zijdemans & Stettina (2014). In this paper, a preliminary framework based on a survey that took place in the form of interviews with participants in different organizations was made. Hence, the study will try to assess quantitative data based on the qualitative model that described on the paper and as a result, the research strategy that will be followed will be a quantitative survey using questionnaire.

In order to cover the research question and establish a concrete guide for the better understanding of contracting practices, a broad questionnaire can be conducted among different IT, software, consultant and lawyer companies. In the paper "Contracting in Agile Software Projects: State of Art and How to Understand it" (Zijdemans & Stettina, 2014) there was a division of contracting types into categories and factors. This was the result of the interviews that took place in different organizations. The analysis of the empirical evidence from the survey that conducted, gave concrete recommendations to practitioners. The framework that was developed, can provide good opportunities to be elaborated and strengthened in a more quantitative research setting. This is the reason why in this thesis, the methodology that will be followed will be a survey.

More specifically, a survey using questionnaire will be operated among organizations that cover the range of the people concerned in contracting. Surveys allow the researcher to collect a wide range of information, illustrating characteristics of different groups and measuring their attitudes and opinions toward certain issues. For example, the questionnaire will be sent to software developers, project owners and lawyers who have experience in contracting practices as legal consultants. By this way, an adequate level of validity will be reached. The content of the questions would be broad trying to gain deeper knowledge on their perceptions and challenges towards different contracting practices. Thus, it will be investigated the *why* and *how* of decision making regarding the contracting types they use and also, how agile characteristics and methods are compatible with other software development methods.

In more details, an overview of the phases of the research process is this:

1. Literature study
2. Survey design
3. Conduct of survey
4. Data processing
5. Analysis
6. Presentation of results

3.1 Survey Methodology

A survey is a scientific process of collection of data. Specifically, a survey methodology studies the data that are gathered from a population that is chosen as a sample, which is representative of a more generic population (Scheuren, 2004). Additionally, survey methodology studies data collection techniques, such as questionnaire construction and methods that their main goal is to improve the accuracy of the survey's results.

Quantitative research also, uses current survey methodologies in order to gather information for a specific population. A simple survey consists of the sample, which is a fraction of the population for which the information is going to be extracted, and of a method of data collection, such as questionnaires. The accuracy of the results is strongly dependent on the representativeness of the sample. For instance, for the thesis' scope, the sample must include suppliers, customers and lawyers that have relevance with contracting practices.

The surveys that are using questionnaires are conducted in a way that every individual of the sample is asked the same questions in a similar way. Moreover, the goal is not to describe the particular individual but to extract a compound profile of the population. That is the reason why most of such surveys are completely anonymous.

Surveys can be classified by the method of data collection that is used. For example, the most widely used are mail or telephone questionnaires and in-person interviews. Mail surveys are the easiest to conduct and the lowest in cost. Usually, there is a danger that people from the sample will not give enough attention and these could lead to insufficient

results. In order to be more efficient, mail surveys should focus on particular groups. In-person interviews are more time consuming and expensive but are necessary when more complex information is needed. Some surveys use a combination of different methods. For instance, mail survey could be used for more generic results and in person interviews for more detailed information.

In this particular thesis the method that is being followed is mail survey. Since the research is quantitative and a big number as well as a wide range of respondents is needed, this method suits best for the author's purpose.

3.2 Questionnaire Construction

Questionnaires are mostly used for quantitative research. They are an efficient way of collecting information from a large number of individuals. A sufficient questionnaire is important for the success of the survey and this is the reason why constructing the questionnaire is one of the most critical phases in the development of the survey. A bad questionnaire can lead to inadequate results that may not represent the opinions of the participants and be misleading (Scheuren, 2004).

Besides well-structured and clear questions, one of the main factors that has to be taken into consideration is the questionnaire's length. Long questionnaires are easier to cause fatigue, negligence and as a result, incomplete answers. Other factors that can be taken into account are the order of the questions and the format of the questionnaire. Moreover, the topics of the questions should fit the respondent's background. For accurate results, the respondents should have enough expertise to answer the questions truthfully. The type of the questions should be aligned with the scope of the survey. For instance there are open, yes-no or multiple choice questions. Lastly, the questions should be neutral and not biased in order to avoid leading the respondents to a predefined outcome.

3.3 Questionnaire Design

The questionnaire that is designed for this particular survey is based on the preliminary framework of Zijdemans & Stettina (2014) from which the main factors that are affecting contracting practices and also, some concrete practices that found in their survey, are being used. Additionally, the model of McLeod & MacDonell, (2011) was used to group the main factors, that are perceived to influence these contracting practices, into categories except for the "Institutional Context" because it is out of the scope of this study.

Following, a diagram of the questionnaire is presented in Figure 19 where its structure can be seen. Also in Figure 20 the grouping of the affecting factors (McLeod & MacDonell, 2011) is depicted.

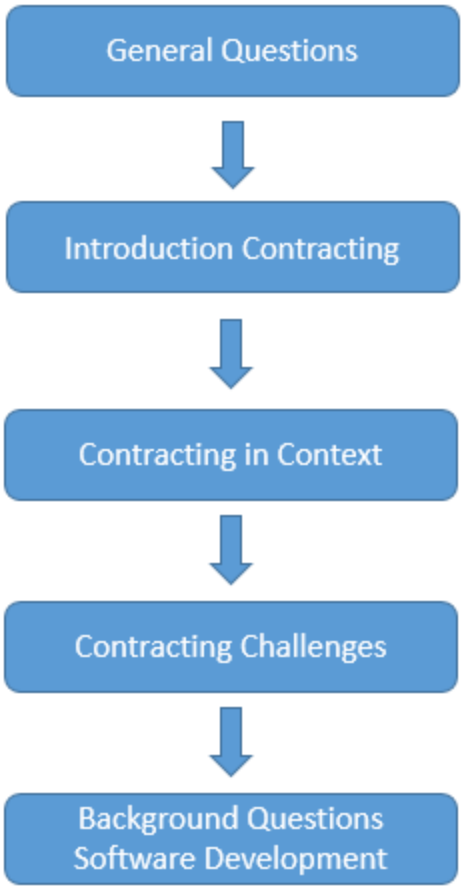

Figure 19 : Questionnaire structure

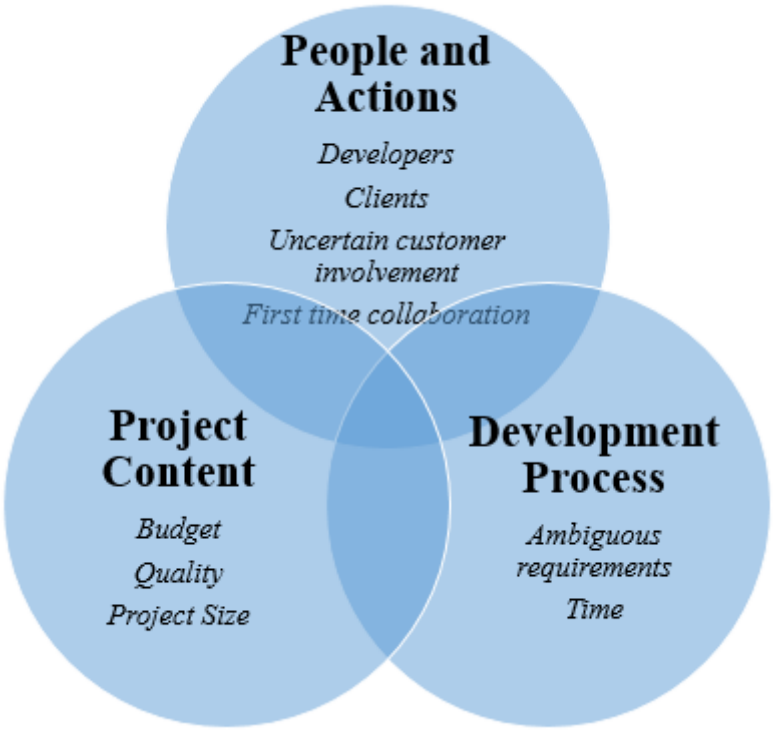

Figure 20: Main factors affecting IT project success (McLeod & MacDonell, 2011)

In the beginning of the questionnaire there is an introduction text which explains the goal of the survey, the estimated time that is needed to be filled up and under which institute the survey is conducted. In order for the questionnaire to be more impartial, the respondents are assured that their data will be treated anonymously. Also, they are encouraged to give their honest opinions. The introduction text is provided in the appendix.

As it is can be seen in Figure 19 the questionnaire is structured in different sections.
In *General Questions* there are questions that are targeting to the professional background of the respondent. The first question is about whether the respondent is a project owner or a supplier. This is very important for the research because a distinctive analysis of the data is being done which is based in these two different groups. In this section there is also a question that is identifying the industrial sector that the organization of the respondent belongs. Additionally, the respondent has to identify his role in the organization between

several choices such as project manager, product owner, scrum master, legal advisor, developer and consultant. Lastly, there is a question about the years of experience of the respondent in the field of software contracting.

The next section comprises an *introduction to contracting.* It has questions that are measuring the knowledge of the respondent on software development and on agile methods. The measuring in all this types of questions is based on a Likert scale which consists of 5 items:

1: not at all knowledgeable
2: slightly knowledgeable
3: moderately knowledgeable
4: very knowledgeable
5: extremely knowledgeable

In the paper "Contracting in Agile Software Projects: State of Art and How to Understand it" (Zijdemans & Stettina, 2014) some categories were distinguished from the interviews of practitioners. From that categories, the main contracting practices that are used from the practitioners and the biggest concerns for software projects were identified. Hence, in this section the question that is following is measuring the knowledge of the respondent in these contracting practices:

- fixed-price fixed-scope
- time & material
- pay per sprint
- target-price
- cost - reimbursement
- exit arrangement
- risk buffer
- two phase
- collaboration agreement
- hiring the developer

Lastly, there is a question categorizing the budget volume of the projects that the respondent is involved with and also, a question asking the respondent's biggest concern when contracting software projects. The concerns that are stated are based on the preliminary framework of Zijdemans & Stettina, (2014) :

- Budget
- Quality
- Time
- Ambiguous Requirements
- Project Size
- Uncertain Customer Involvement
- First Time Collaboration

In *Contracting in context* the contracting practices and the biggest concerns from the previous section are combined. According to the respondent's answer the questionnaire differentiates and contains more specific questions depending on his options about his biggest concerns in contracting software projects. Thus, if the respondent's opinion is that quality and time are the most important concerns regarding contracts in software projects then questions specifically relevant with these options will appear. Particularly, these questions will ask the respondent to rate the suitability of each of the previous stated contracting practices based on their chosen concern. By this way the questionnaire becomes faster and there are better possibilities that the respondent will finish it instead of leaving it incomplete because of fatigue. The rating in these questions is based on a Likert Scale which consists of 5 items:

1: not at all suitable
2: slightly suitable
3: moderately suitable
4: very suitable
5: extremely suitable

The next section is about *Contracting Challenges*. Specifically, the respondent can choose the biggest challenge that he would consider in agile projects from choices such as scoping, pricing, collaboration, dispute resolution and acceptance. There are also two open questions where the respondent can elaborate more on the challenges and make his own proposition for improvement in contracting for software development and procurement projects.

In the last section of the questionnaire there are some *Background Questions for Software Development.* Particularly, there are three questions for rating the importance of discipline, high level planning and documentation in Agile methods. Moreover, the effectiveness of Agile methods in developing software is rated as well as the software development method that is used in the respondent's organization. The section and the whole questionnaire closes with some open questions. The respondent can answer whether he chose to apply an Agile project management method, why did he choose it and if it catered his needs and also if he could think a way of fitting the existing contracting practice of his organization to the iterative nature if Agile methods.
For the ease of the reader the whole questionnaire with its different sections is provided in the appendix.

3.4 Data Collection

In this part the way the survey was conducted is described. More specifically, the actions that required to be done in order to test the questionnaire and to distribute it are analyzed.

3.4.1 Pre Study

Before distributing the questionnaire to different contacts, it was tested by the author and his supervisor in context of understanding and duration. These concepts are highly connected because we wanted to make the questionnaire understandable, interesting and less time consuming in order the respondent to complete it and not to leave it unanswered.

For instance, in the section *Contracting in context*, two versions of questions were tested. Firstly, the rating on this questions decided to be done with a list of the contracting practices where the respondent could rearrange their order by dragging and dropping them. By this way, the respondent could indicate which contracting practice would presume more suitable that the others by putting it higher in the list. Although, this was interesting, it was confusing for the respondent and time consuming. It would require a lot of effort to rearrange the whole list and the respondent could lose his interest to complete the questionnaire. Thus, in order to avoid that we chose to apply a Likert scale in these questions. Additionally, in the beginning all the questions of this section would appear to the respondent but after tests, just the questions that are based to the respondent's choices in the previous question regarding the biggest concern in contracting software projects will were chosen to appear. By this way the questionnaire became more efficient and less time consuming.

Moreover, in order to find weaknesses in the questionnaire and also possible new questions, a meeting was held between the author, his supervisor and a legal advisor who specializes in intellectual property and technology. Thus, indications were made about removing some not relevant questions and adding some more relevant and interesting ones for professionals. Based on the legal advisor's feedback, the questionnaire's content completed and the survey needed to be distributed.

3.4.2 Questionnaire Distribution

In the beginning, before distributing the questionnaire, the type of professionals that needed to respond had to be identified. Since the survey is about contracting practices in traditional and agile software development, it concerns people in companies or organizations that are developing software for internal or external use. It also, concerns people in companies that are outsourcing the development of software projects as well as legal departments of such companies that are managing these kind of contracts. Specifically, the range of interest for possible respondents comprises of

- project managers
- product owners
- scrum masters
- agile coaches
- development staff
- consultants
- legal advisors

3.4.2.1 Own Network

After identifying the possible respondents of the survey, the distribution was made to the author's and his supervisor's personal contacts. Hence, the questionnaire was sent to their personal emails. All the emails contained a standardized text which was asking the potential respondents to participate in the survey by presenting them briefly the benefits of the results on their account in order to be intrigued and complete the survey. Additionally, the respondents are kindly asked whether they would be able to distribute the survey in their organization and to their network. Because the contacts belong to different groups a s described in the identification process, the standardized text was differentiated according

to the respondent's professional background. Examples of the texts that were sent through direct emails are shown in the appendix.

3.4.2.2 Network Expansion

In order for a bigger amount of relevant contacts to be found and the author's network to be expanded, there was made a research in the internet for finding consulting, software providers and other relevant to the survey companies. Within their management teams, when there was the ability for contact details to be found, relevant people were identified in order the questionnaire to be distributed directly to them with the way described before. When there were occasions of contacts that did not reply, a reminder email was sent to them after one month.

3.4.2.3 Social Networks

Social networks comprise a very important and helpful tool for researchers that try to conduct a quantitative survey. More specifically, Linkedin is a network for professionals where you can very easily identify specific people that fit in the requirements of the survey. Linkedin gives the opportunity to make a search based on the person's profession and to contact someone directly. For instance, a lot of IT project managers were found and contacted through the network's platform.

In addition there are a lot of agile or project management oriented groups where there is capability of posting the survey in order more people that would be interested in to see it and participate. In appendix the message posted in Linkedin groups can be found.

Moreover, other social networks were used but not with the same success of LinkedIn. For instance, particular groups were identified in Google Plus, Facebook and Flickr where the survey was posted.

Lastly, in SlideShare platform there are slides from professionals where their contact details can be found and the survey could be sent to them directly.

4. RESULTS

In this section the author is presenting the results that are based on the collected data. The results are divided into the same sections that constitute the questionnaire. Since the survey is quantitative, the data are presented with descriptive statistics, charts and data tabulation. The data were gathered through the online survey tool Qualtrics, where also, some of the tables and graphs were developed. Additionally, Microsoft Excel was used in order for some charts to be created.

Moreover, some correlations between the results are presented while the differences between groups that are statistically important are shown. Correlations were found using linear regression analysis and differences were found using analysis of variance (ANOVA) in SPSS.

The questionnaire was answered by 79 respondents in 14 countries. As it can be seen in Figure 21 the vast majority of the respondents is from The Netherlands. This is reasonable because most of the personal contacts of the author and his supervisor work and distributed the questionnaire there. Nevertheless, there are respondents from more countries due to the broad distribution of the questionnaire through social networks.

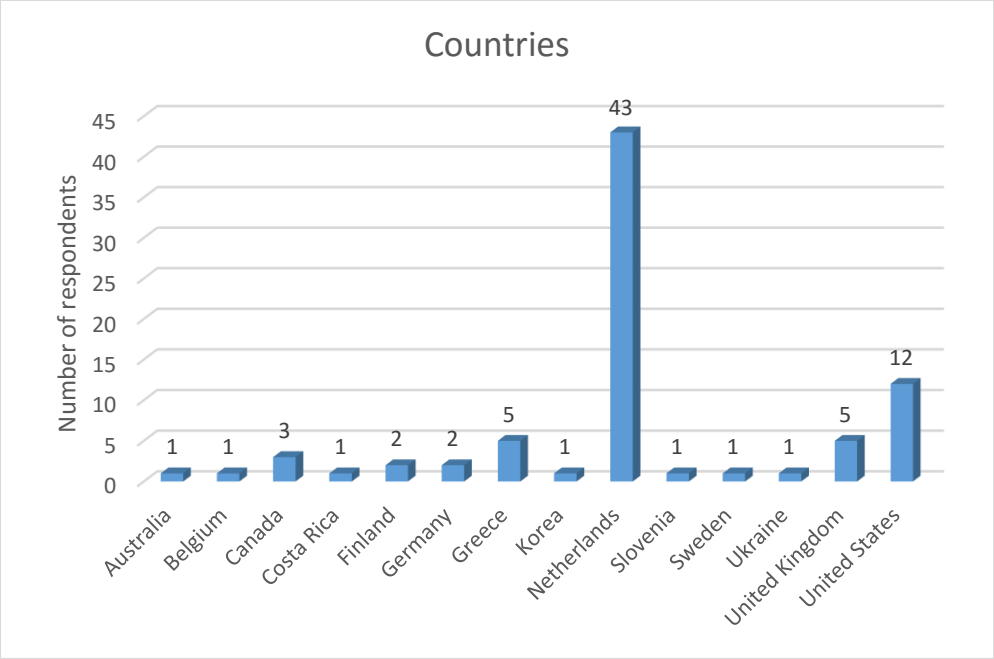

Figure 21: Countries of distribution

4.1 General Questions

In the first group of results there is the identification of the professional background of the respondents. Also, there is a distinction between the two sides of the respondents which is useful for the categorization of the results and the comparison between their different choices and concerns.

Between 79 respondents there were 51 who work in companies that supply software projects and 28 respondents in companies that are buying such projects. The percentage of the distinction of the two sides is shown in Figure 22.

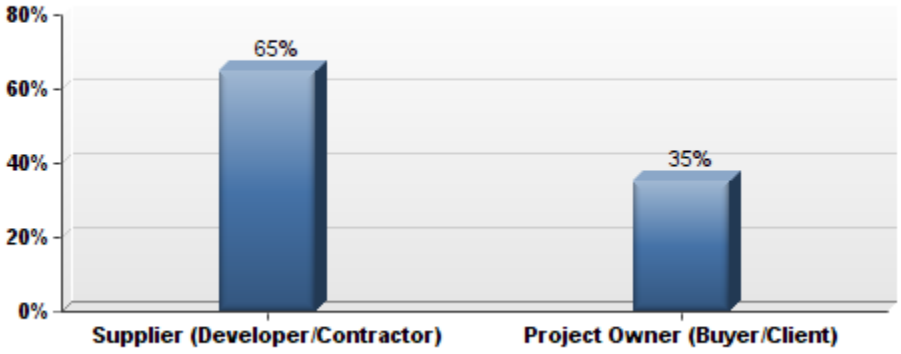

Figure 22: Respondent's side

Most of the respondents' organizations belong to services sector such as IT, banking and retail followed by organizations that belong to public sector as can be shown in Figure 23. This can be easily explained because the questionnaire was mostly distributed to contacts belong in services sector and also to researchers in universities.

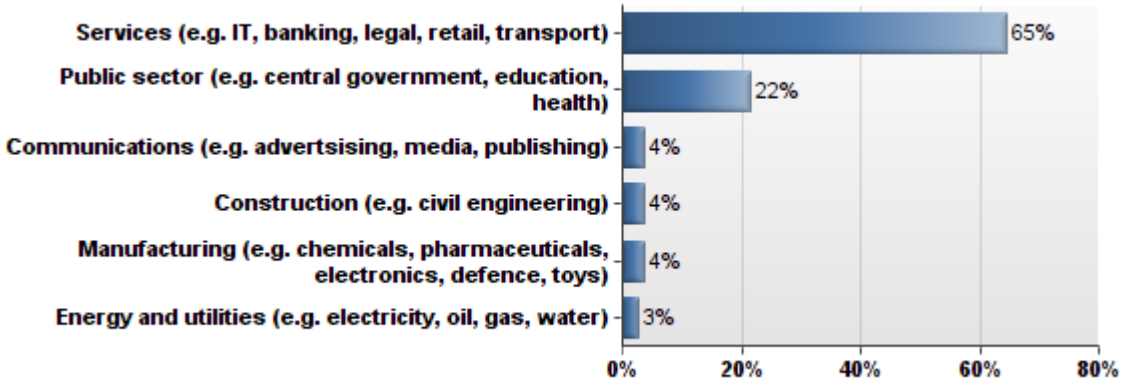

Figure 23: Industrial sector of respondent's organization

Additionally, in Figure 24 the roles of the respondents are depicted. Between suppliers it is expectable that most of them are project managers, consultants and developers. There are also suppliers that identify themselves as delivery and service managers.

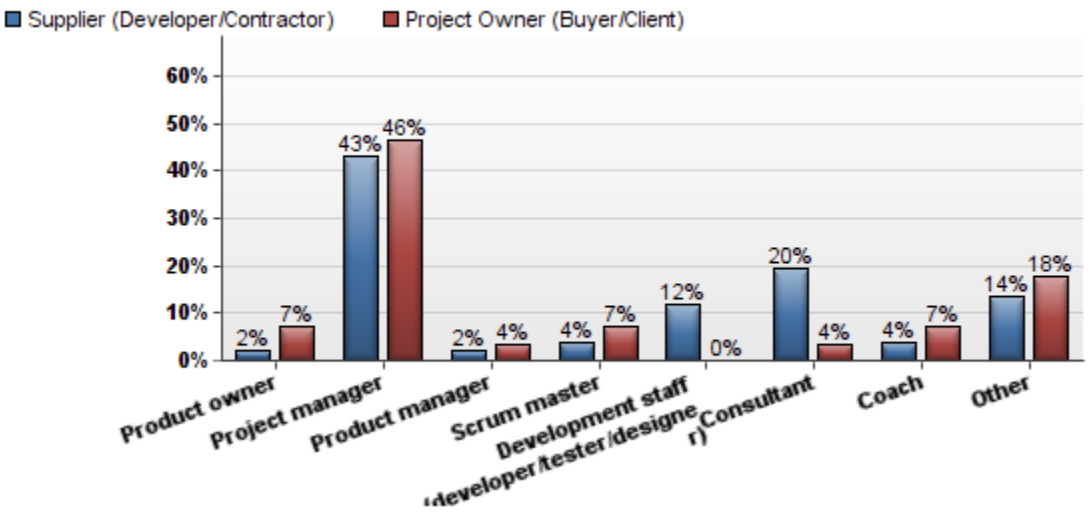

Figure 24: Respondents' roles

On the other hand, in the buyer's side, most of the respondents are project managers and scrum masters, while a big percentage did not choose a role from the list but is identified as Chief Information Officers, Portfolio managers and Quality Assurance Supplier.

The majority of the respondents that completed the questionnaire is quite experienced in the field of software contracting as it can be shown in Figure 25.

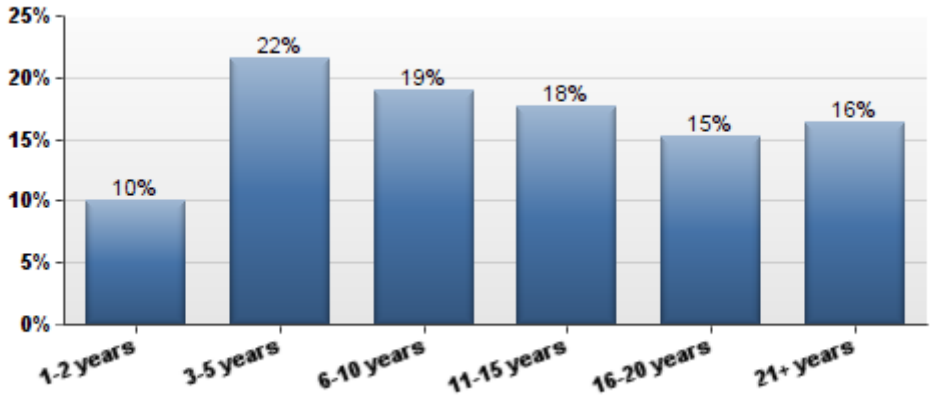

Figure 25: Years of experience in the field of software contracting

Only 10% has little experience of 1 to 2 years, while most of the respondents (22%) have 3 to 5 years of experience. Additionally, the remaining 68% has an experience in software contracting practices of more than 5 years. The average years of experience are between 6 to 15.

4.2 Introduction Contracting

In this group of results the knowledge of the respondents regarding software development methods and specific contracting practices is presented. Additionally, the average size of projects, in euros, that the respondents are managing is identified. Lastly, the biggest concerns about contracting software projects are shown.

In Figure 26 and Figure 27 the knowledge of suppliers and customers on Agile methods and software development is presented. The majority of both sides has more than adequate knowledge since they identify themselves as very knowledgeable in these fields. This is important because their sufficient knowledge make the data in the next sections more accurate for the scope of this study.

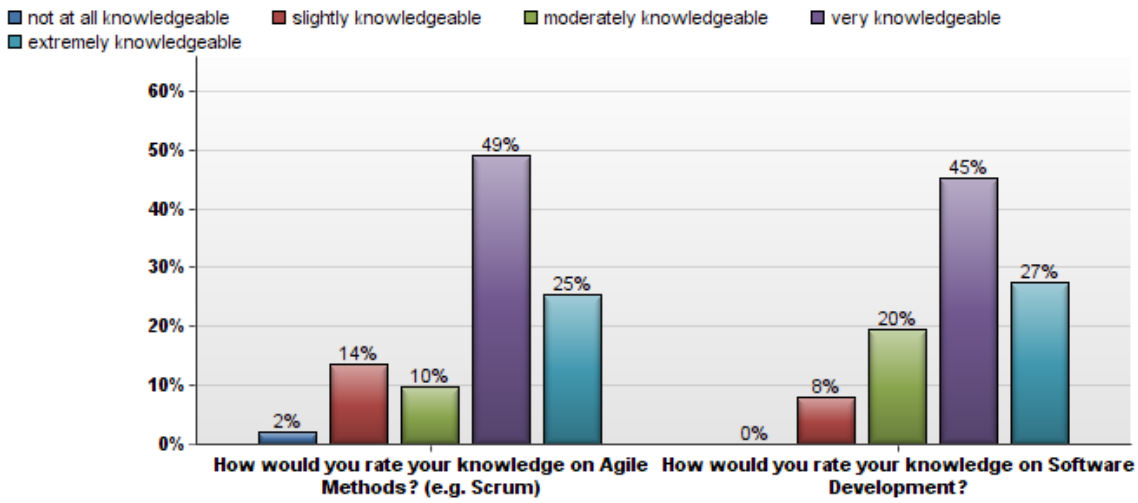

Figure 26: Suppliers' knowledge in software development projects

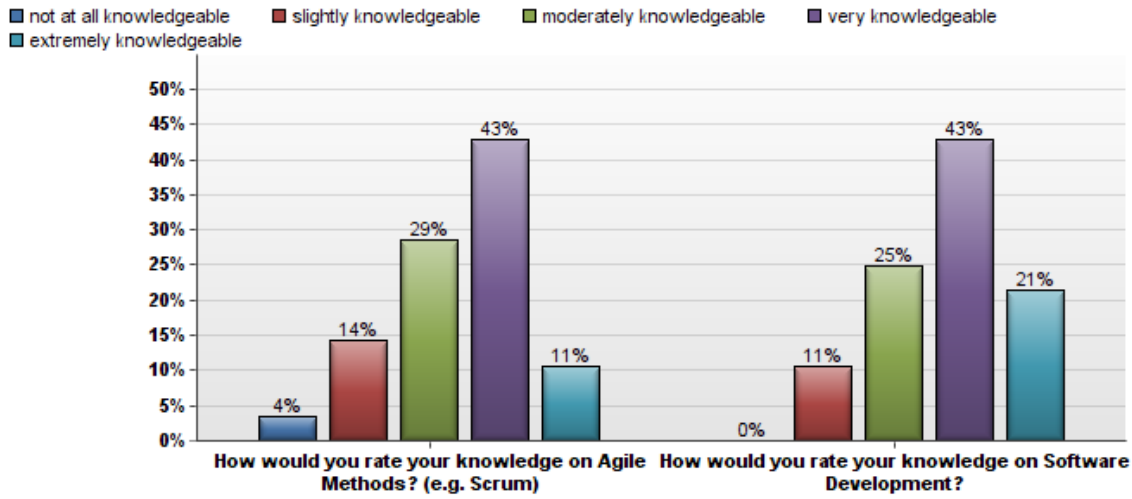

Figure 27: Customers' knowledge in software development projects

Figure 28 shows the average knowledge of suppliers and buyers regarding specific contracting practices for software development projects that were described in chapter 2 of the thesis, while Figure 29 shows the general picture for all respondents.

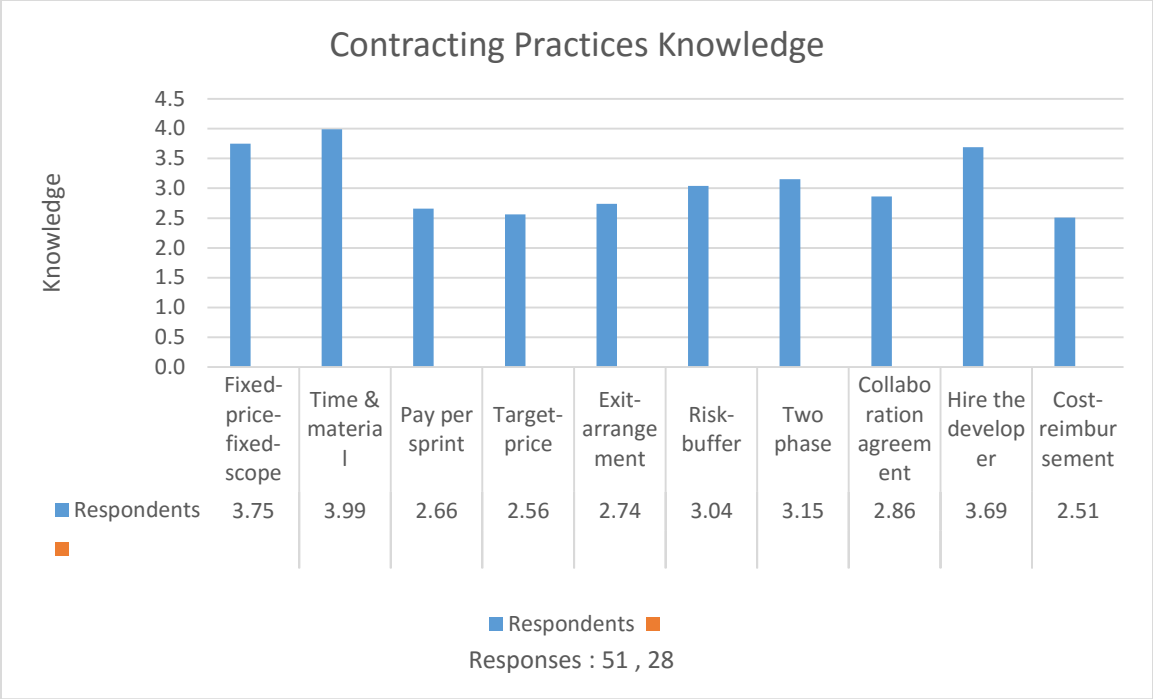

Figure 28: Average knowledge in contracting software development projects and respective contracting practices for suppliers and buyers. Knowledge is measured as: 1. not at all, 2. slightly, 3. moderately, 4. very, 5. extremely knowledgeable

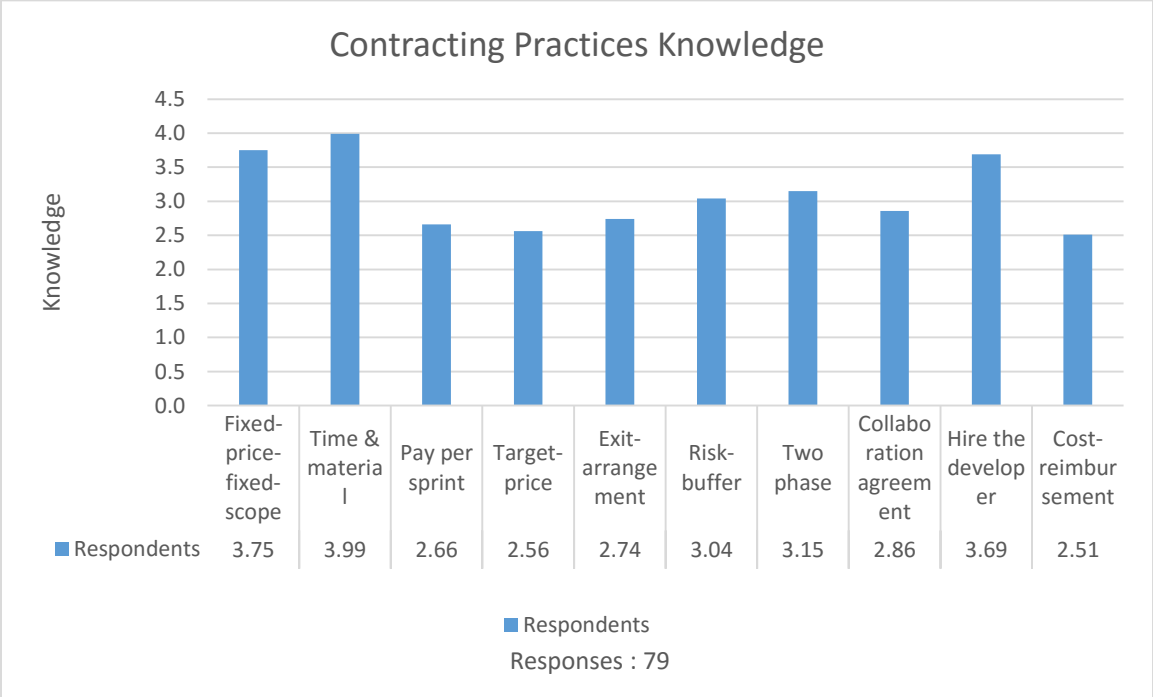

Figure 29: Average knowledge in contracting software development projects and respective contracting practices for all respondents. Knowledge is measured as: 1. not at all, 2. slightly, 3. moderately, 4. very, 5. extremely knowledgeable

As it can be seen, both suppliers and buyers have better knowledge in traditional contracting practices such as Fixed-price-fixed-scope and Time & Material. Additionally, they have a strong familiarization in hiring the developer. However, the familiarization of the respondents with all the contracting practices that were asked is high.

The average size in euros of the projects that the respondents are managing or participating can be found in Figure 30. Mostly, projects in the range of 100.000 to 499.999 euros and more expensive than 1.000.000 euros are being managed by the respondents.

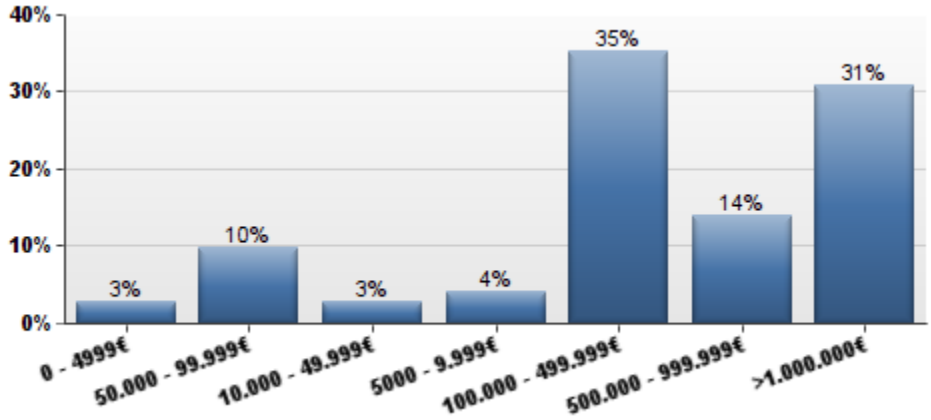

Figure 30: Average size of projects in euros (€)

Lastly, the biggest concerns regarding contracting in software projects are presented in Figure 31.

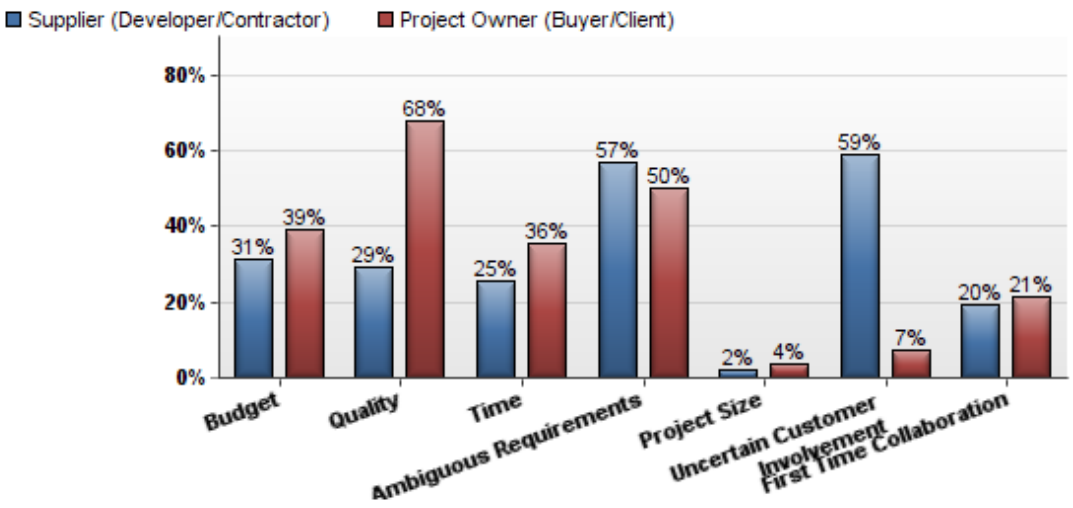

Figure 31: Concerns when contracting in software projects

4.3 Contracting in Context

In this section the suitability of the specific contracting practices that were given to the respondents based on the concerns that affecting them is presented.

- Budget

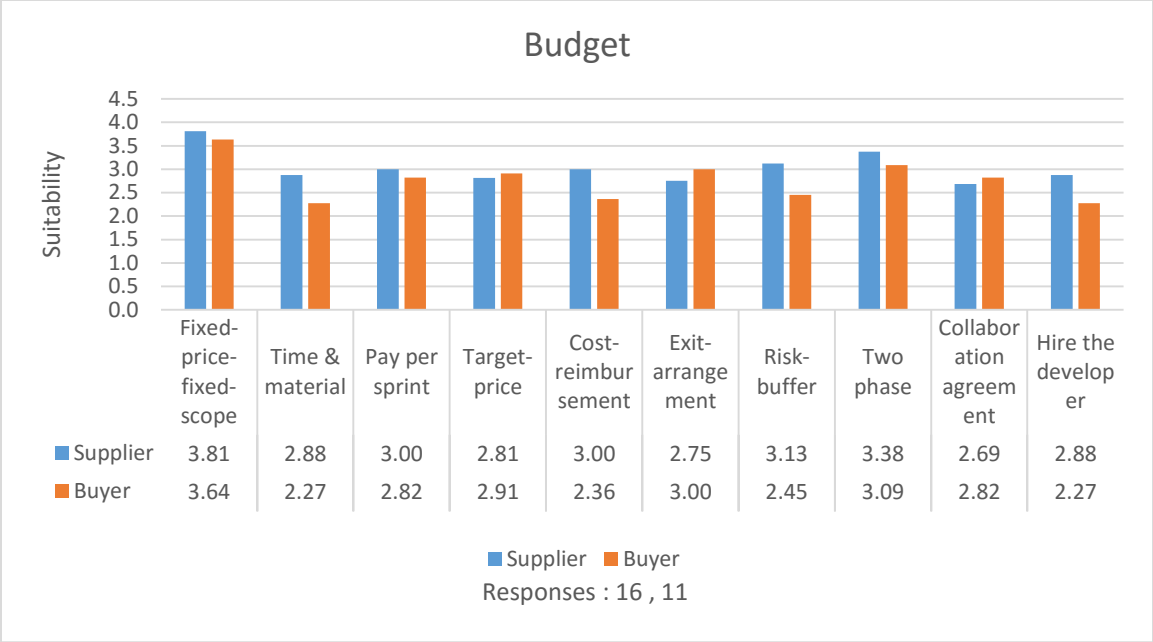

Figure 32: Average suitability of contracting practices considering budget as the most critical aspect of the project. Suitability is measured as: 1. not at all, 2. slightly, 3. moderately, 4. very, 5. extremely suitable

Assuming that budget plays the most important role in the project both suppliers and buyers consider Fixed-price-fixed-scope as the most suitable contracting practice as it can be seen in Figure 32.

- Quality

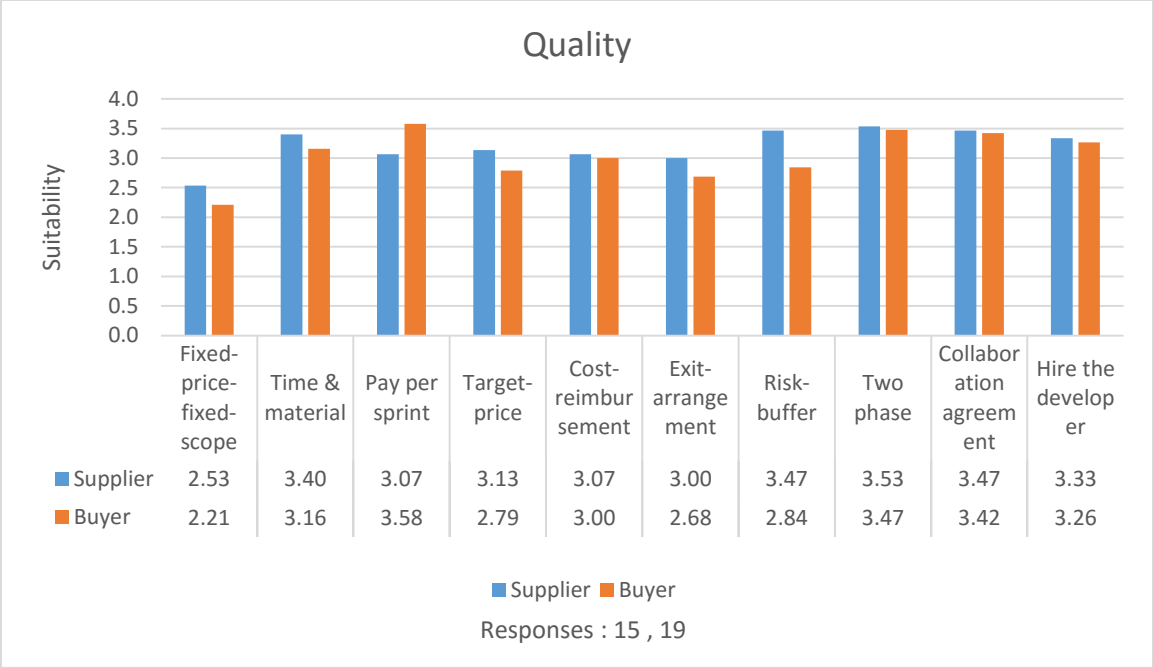

Figure 33: Average suitability of contracting practices considering quality as the most critical aspect of the project. Suitability is measured as: 1. not at all, 2. slightly, 3. moderately, 4. very, 5. extremely suitable

Figure 33 shows that considering quality as the most critical aspect of the project, suppliers assume that Two phase is most suitable while buyers assume that Pay per sprint has the best suitability.

- Time

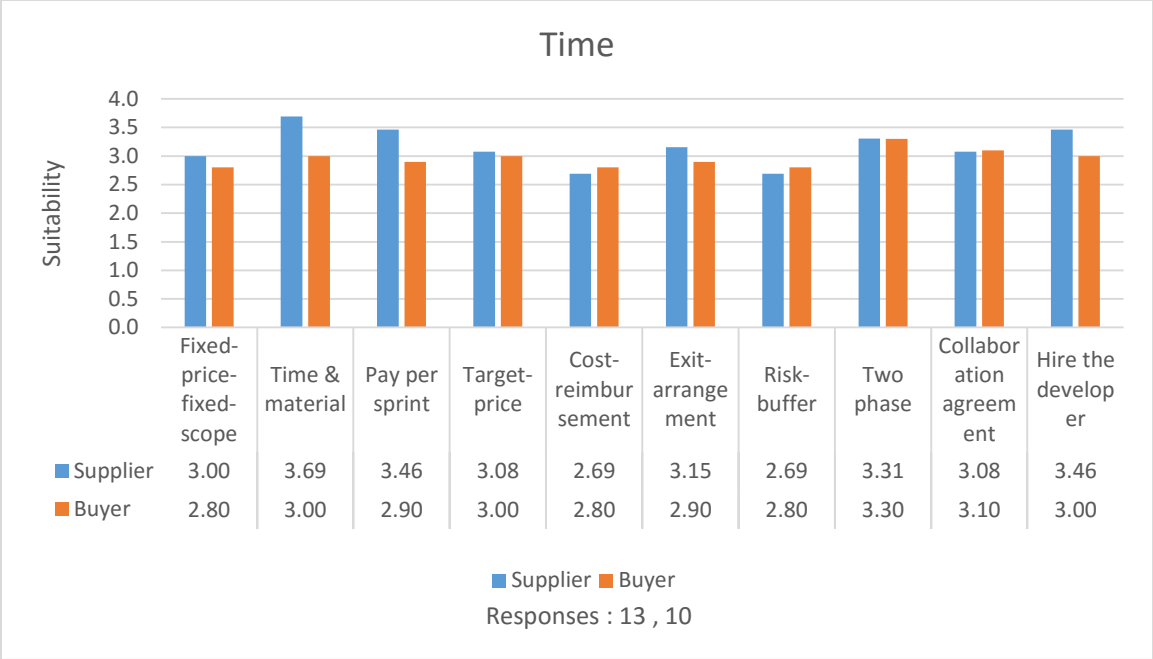

Figure 34: Average suitability of contracting practices considering time as the most critical aspect of the project. Suitability is measured as: 1. not at all, 2. slightly, 3. moderately, 4. very, 5. extremely suitable

Figure 34 shows that when time plays the most important role for a project, suppliers consider Time & material as most suitable while buyers consider Two phase as most suitable.

- Ambiguous Requirements

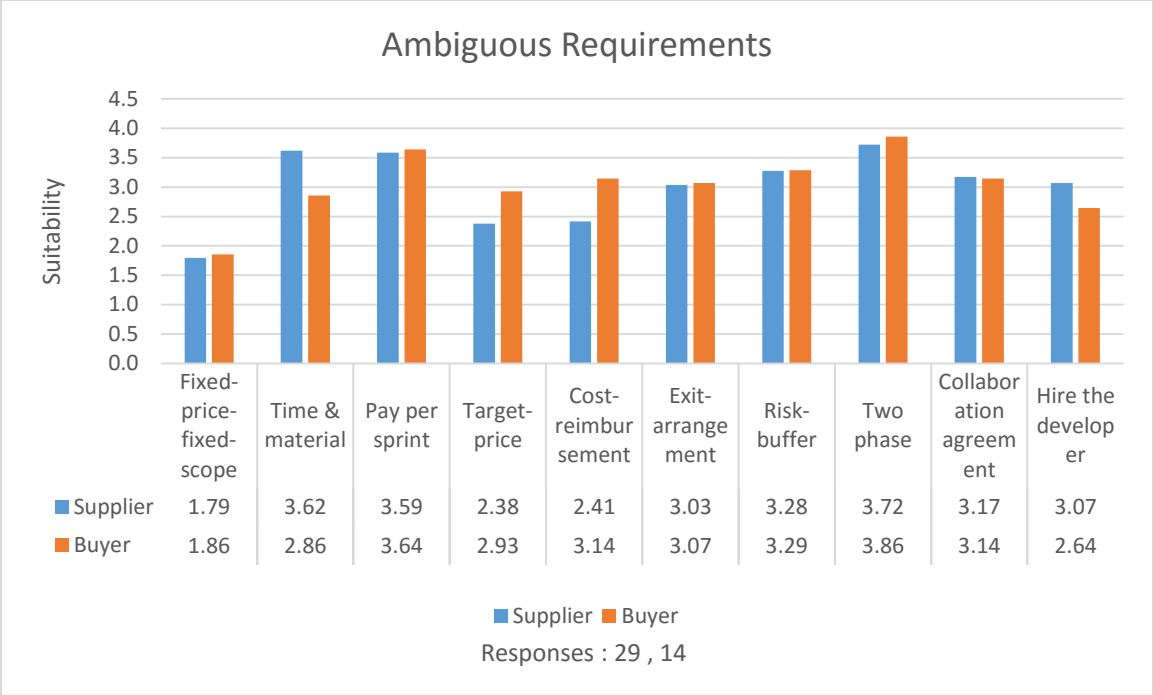

Figure 35: Average suitability of contracting practices considering ambiguous requirements as the most critical aspect of the project. Suitability is measured as: 1. not at all, 2. slightly, 3. moderately, 4. Very, 5. extremely suitable

Figure 35 shows that when ambiguous requirements is the biggest concern for a project, both suppliers and buyers consider Two phase as most suitable contracting practice.

- Uncertain Customer Involvement

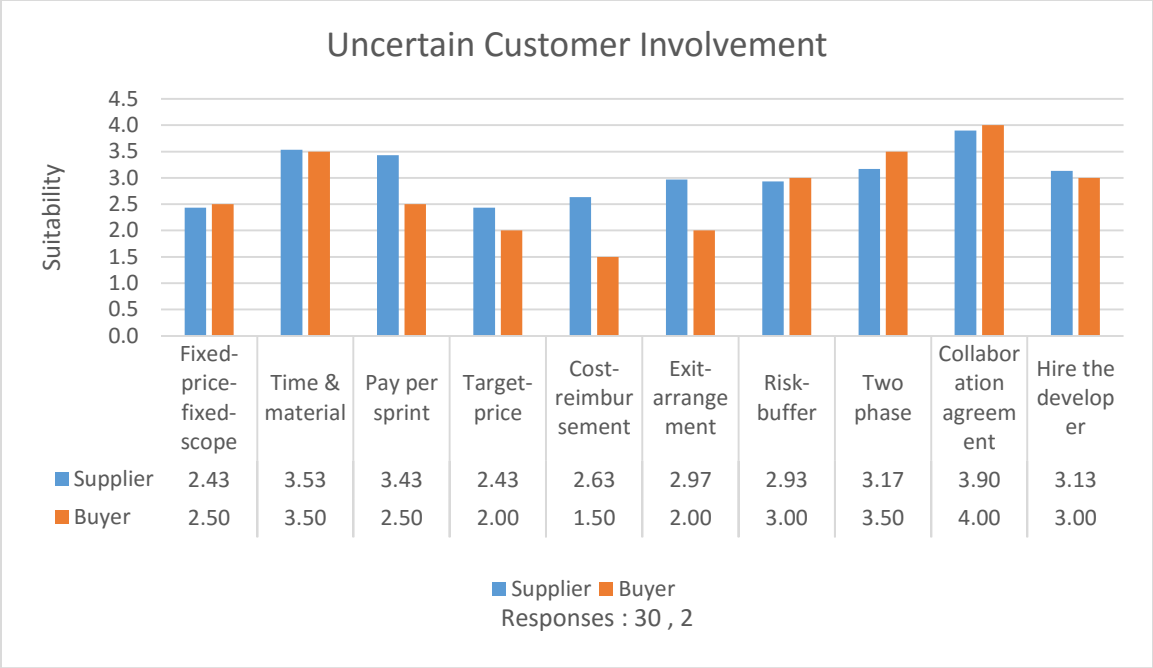

Figure 36: Average suitability of contracting practices considering uncertain customer involvement as the most critical aspect of the project. Suitability is measured as: 1. not at all, 2. slightly, 3. moderately, 4. very, 5. extremely suitable

Figure 36 it is shown that when uncertain customer involvement plays the most important role in a project, suppliers assume Collaboration agreement as the most suitable contracting practice.

- First Time Collaboration

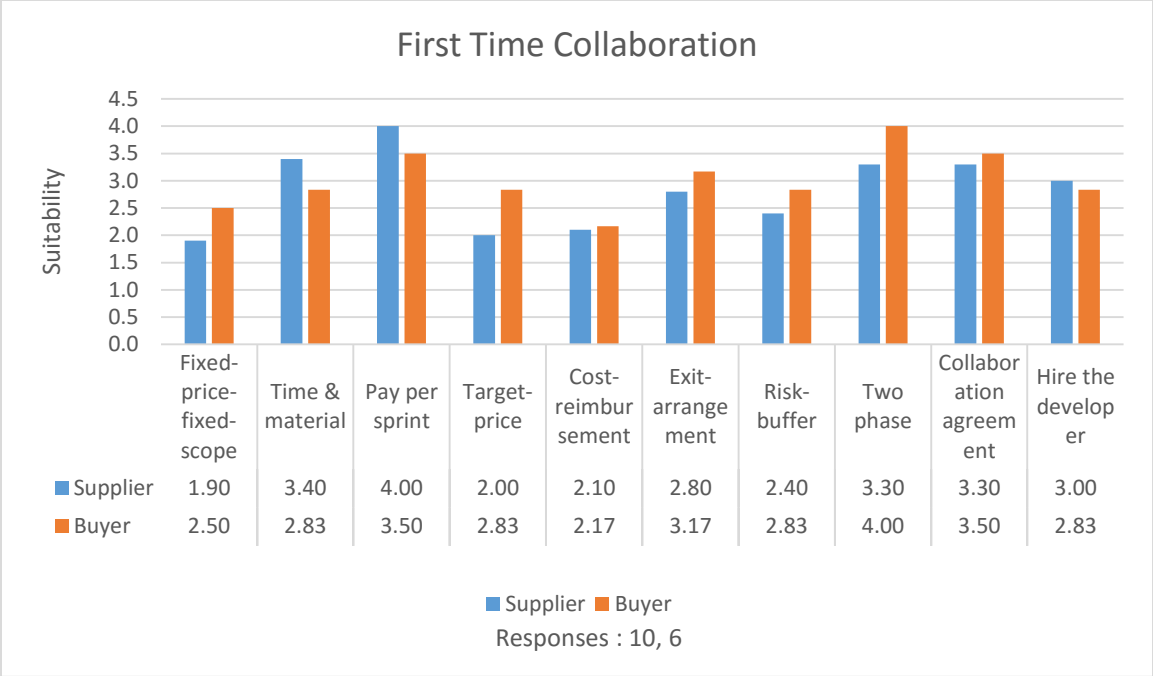

Figure 37: Average suitability of contracting practices considering first time collaboration as the most critical aspect of the project. Suitability is measured as: 1. not at all, 2. slightly, 3. moderately, 4. very, 5. extremely suitable

In Figure 37 it is shown that suppliers find Pay per sprint most suitable when the biggest concern is first time collaboration. On the other hand, buyers find Two phase most suitable.

- Agile Software Development Methods

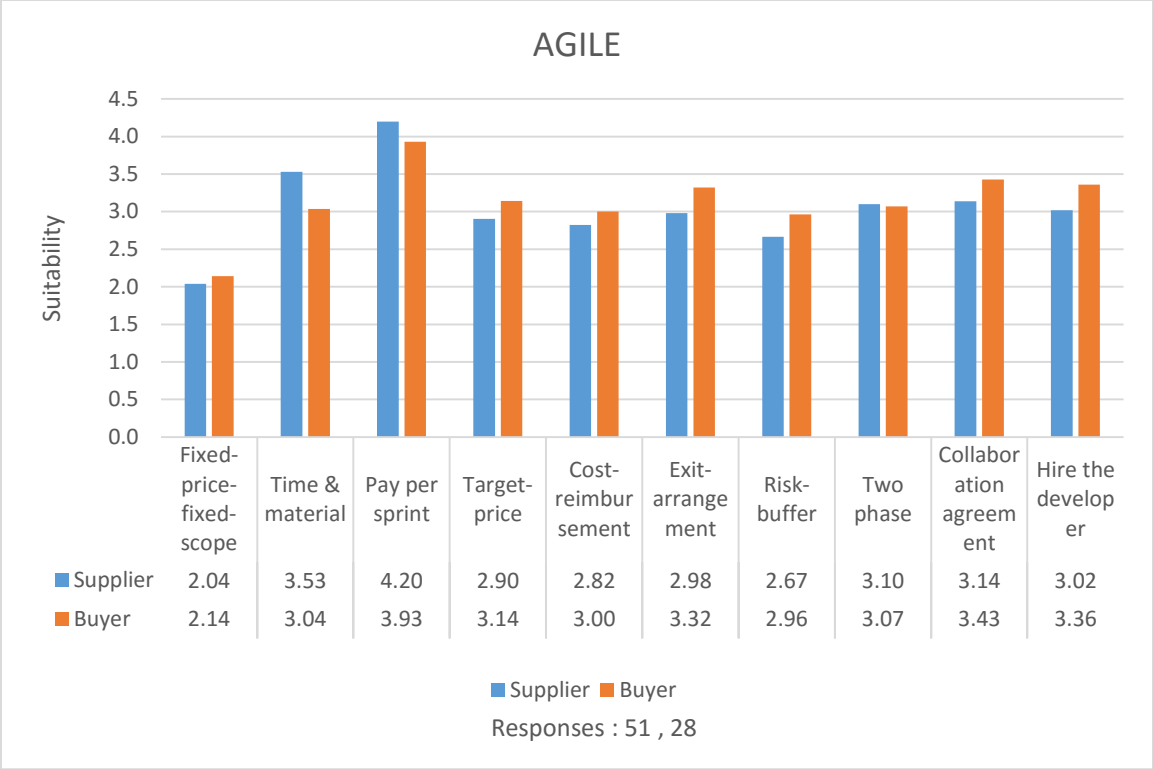

Figure 38: Average suitability of contracting practices considering the project is executed using agile software development methods. Suitability is measured as: 1. not at all, 2. slightly, 3. moderately, 4. Very, 5. extremely suitable

Figure 38 shows which contracting practices are perceived as most suitable when the project is executed using agile software development methods. Both suppliers and buyers find Pay per sprint as most suitable.

The tables in Figure 39 and Figure 40 show the contracting practices that have the best average suitability for each factor for suppliers and buyer. Figure 41 shows the respective average suitability for all respondents alongside with standard deviation. Also, the total responses for each practice are shown in the last column.

Factor	Practice	Mean of Suitability	Responses
BUDGET	Fixed-price fixed-scope	3.81	16
QUALITY	Two phase	3.53	15
TIME	Time & material	3.69	13
AMBIGUOUS REQUIREMENTS	Two phase	3.72	29
UNCERTAIN CUSTOMER INVOLVEMENT	Collaboration agreement	3.9	30
FIRST TIME COLLABORATION	Pay per sprint	4	10
AGILE SOFTWARE DEVELOPMENT METHODS	Pay per sprint	4.2	51

Figure 39: Best average suitability on each factor for suppliers

Factor	Practice	Mean of Suitability	Responses
BUDGET	Fixed-price fixed-scope	3.64	11
QUALITY	Pay per sprint	3.58	19
TIME	Two phase	3.3	10
AMBIGUOUS REQUIREMENTS	Two phase	3.86	14
UNCERTAIN CUSTOMER INVOLVEMENT	Collaboration agreement	4	2
FIRST TIME COLLABORATION	Two Phase	4	6
AGILE SOFTWARE DEVELOPMENT METHODS	Pay per sprint	3.93	28

Figure 40: Best average suitability on each factor for buyers

Factor	Practice	Mean of Suitability	Std. Deviation	Responses
BUDGET	Fixed-price fixed-scope	3.74	1.02	27
QUALITY	Two phase	3.5	0.86	34
TIME	Time & material	3.39	1.27	23
AMBIGUOUS REQUIREMENTS	Two phase	3.77	0.92	43
UNCERTAIN CUSTOMER INVOLVEMENT	Collaboration agreement	3.91	1.00	32
FIRST TIME COLLABORATION	Pay per sprint	3.81	0.83	16
AGILE SOFTWARE DEVELOPMENT METHODS	Pay per sprint	4.1	0.89	79

Figure 41: Best average suitability on each factor for all respondents

Additionally, there was created a variable in SPSS which measures the average suitability of each contracting practice for all the aforementioned factors. Although it could not be tested for consistency because in most factors there are missing values, a tangible

impression of which contracting practice is considered as most suitable for all these factors is shown in Figure 42 and Figure 43.

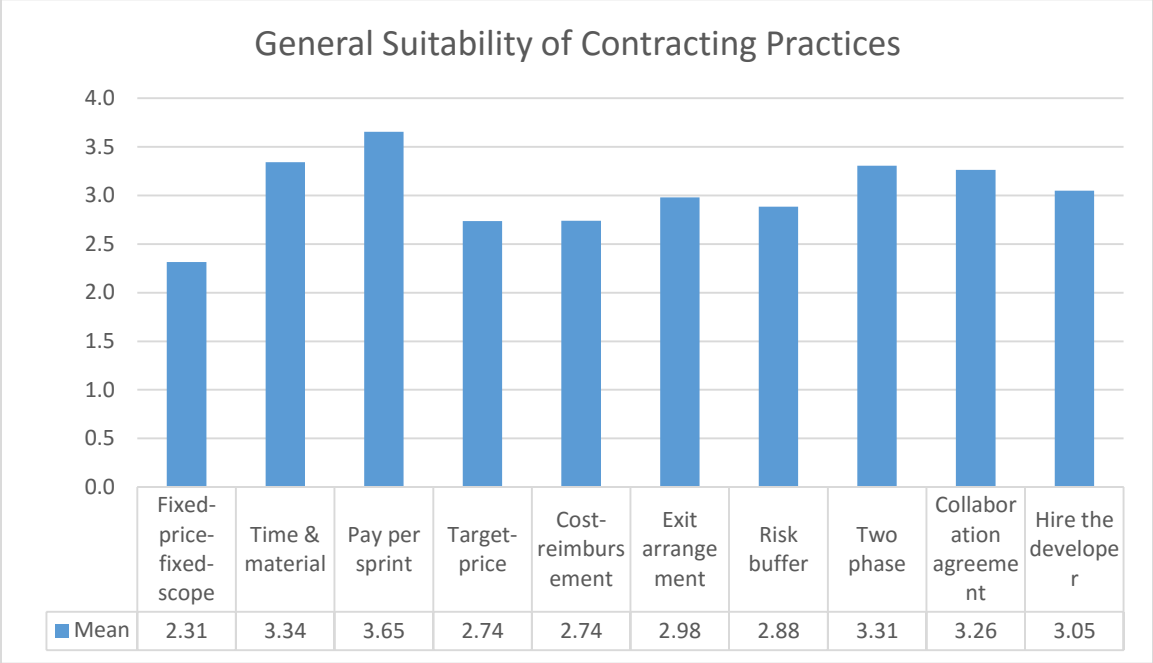

Figure 42: Best general suitability

Contracting Practice	Std. Deviation	Mean
Fixed-price-fixed-scope	0.99	2.31
Time & material	0.83	3.34
Pay per sprint	0.69	3.65
Target-price	0.74	2.74
Cost-reimbursement	0.84	2.74
Exit arrangement	0.76	2.98
Risk buffer	0.76	2.88
Two phase	0.86	3.31
Collaboration agreement	0.93	3.26
Hire the developer	1.01	3.05

Figure 43: Best general suitability with standard deviation

4.4 Contracting Challenges

In this section the contracting challenges that the respondents consider regarding Agile projects are shown. Additionally, there is an elaboration on these challenges and improvement suggestions for contracting practices in software development and procurement projects which are based on the survey's responses.

The most important challenges for suppliers and buyers are depicted in Figure 44.

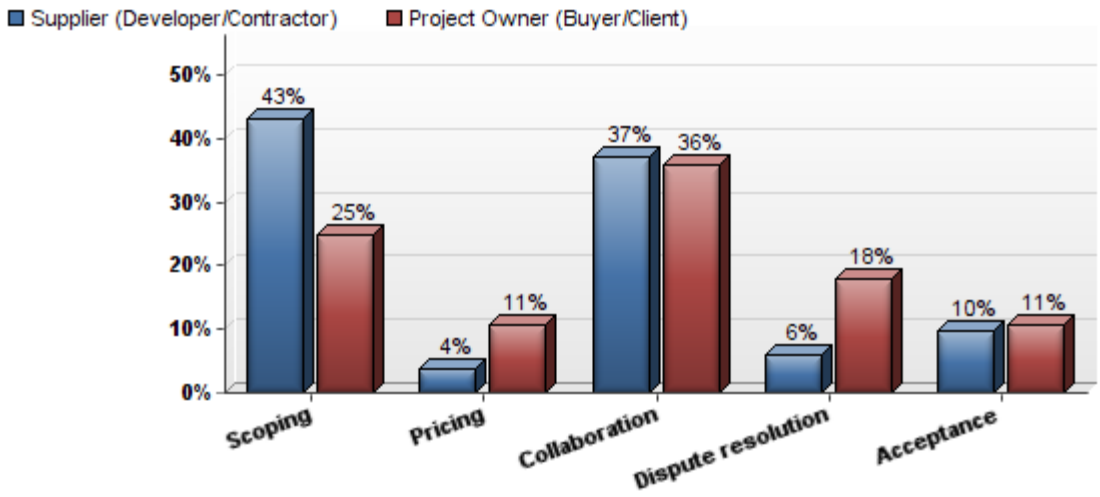

Figure 44: Challenges when contracting in Agile projects

The participants of the questionnaire were asked to elaborate on these challenges. Regarding scope they answered that it is sometimes difficult to be defined. It is a challenge to prevent the scope to be extravagant when lots of new requirements are being added to the backlog impacting the initial scope of the project. Also, sometimes the customers are not fully aware of the entire scope and that is the reason that it needs to be defined as smart as possible and the customers need to accept that it is flexible. Moreover, suppliers could explain what agile means to the customers. Regarding collaboration, the respondents consider that it is important for the suppliers to ensure that customers understand what kind of involvement is needed by them. Additionally, both of the two sides need to really

understand the context in which the project takes place in order to have better collaboration. Furthermore, regarding general contracting in Agile projects, it is challenging when there are customers who want estimates and pricing upfront. Getting used to an Agile way of working means to change old habits and trust each other. This requires a lot of time and effort. Lastly, if there is no collaboration, Agile is not going to work at all because scope is a moving target.

In question on how contracting and contracting practices could be improved in software development and procurement projects, many of them replied that common understanding of Agile and tight collaboration is needed. An interesting point of view is that procurement needs to be done by the project team and not by a different department which is driven by cost, and also, contracts must focus on quality instead of time and scope. Some respondents, also think that contracting practices could be improved with the use of paying per sprints while some others suggest the using of function point and metrics for scope.

4.5 Background Questions for Software Development

In the last part of results the understanding of Agile methods between the respondents is presented. Furthermore, the effectiveness of the software development method followed in the respondent's organization is measured. Also, there is an elaboration on the reasons for a possible preference in an Agile project management method by the respondents and explanation of possible ways that the contracting practice followed in the respondent's organization could fit the iterative nature of Agile methods.

In Figure 45 the average importance of certain factors –discipline, high level planning, documentation- in Agile methods is depicted. As it can be clearly seen that both suppliers and buyers agree on the importance of these factors. Discipline is considered as most important for an appropriate use of Agile methods.

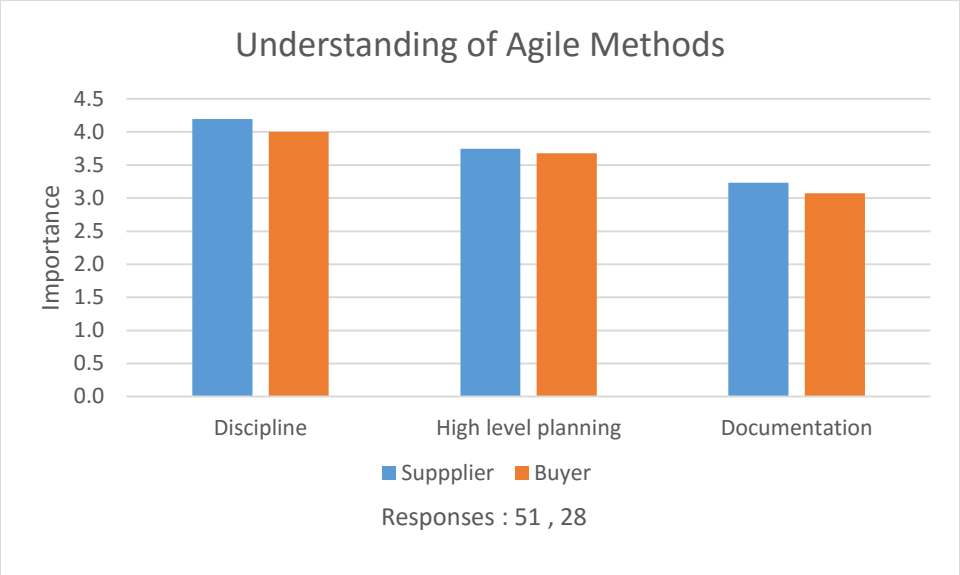

Figure 45: Understanding of Agile methods. Importance is measured as: 1. not at all, 2. slightly, 3. moderately, 4. Very, 5. extremely important

Figure 46 shows the perceived effectiveness of Agile methods for developing software and of the specific software development method used by the respondent's organization.

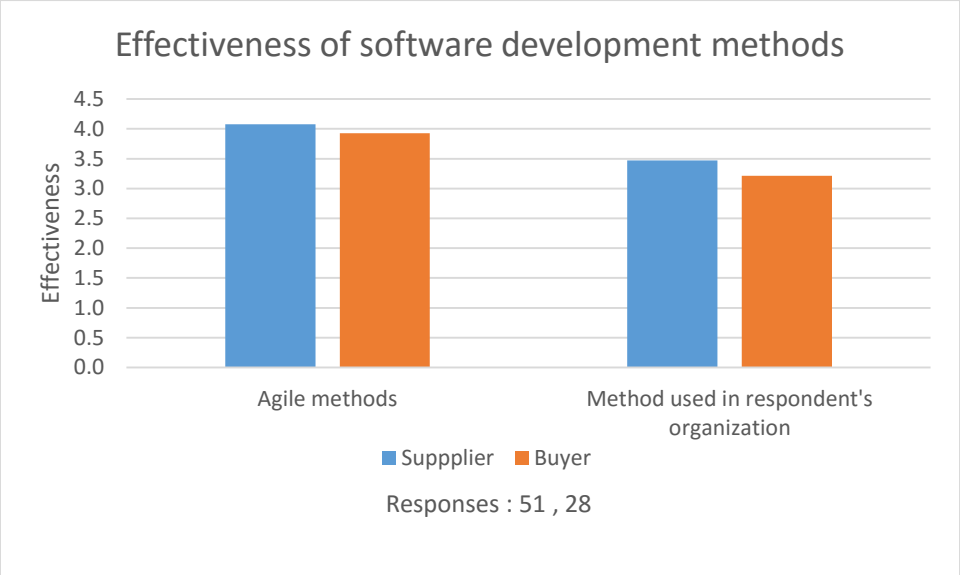

Figure 46: Effectiveness of software development methods. Effectiveness is measured as: 1. not at all, 2. slightly, 3. moderately, 4. Very, 5. extremely effective

In this section there was given a chance to respondents to elaborate whether they chose an Agile method in the past, why and if it catered their needs. The majority of them have used or using Agile methods in the past. Most popular method between the respondents is Scrum, while there were some of them that have used Kanban, XP, DSDM (Dynamic Systems Development Method), SAFe (Scaled Agile Framework) and Discipline Agile Delivery. Almost all of them answered that Agile methods catered their needs but there are some challenges. In some cases there is misunderstanding with the product owner because he assumes that everything in Agile is flexible and as a result there are incomplete requirements before the sprint starts. In other cases, some responded that their organization is not ready for Agile, or that it is not fully understood by upper level management and this makes the adoption difficult. Additionally, there are statements that some clients are not willing to truly commit in the project as Agile methods require. On the other hand, many respondents elaborated on the benefits of using Agile methods. Some of them responded that these methods give a better insight in real wishes of customers while other stated that Agile improves development time, it has lowest risk and it requires less procedure and focuses more on the results of the project.

Lastly, there is a question where the respondents could give their opinion on how the existing contracting practice that is used by their organization could fit the iterative nature of Agile methods. Some of them proposed that it is better to start with in-house development or hire the contractors directly while others proposed fixed price contracts with the using of variable scope. However most of them tried to incorporate iterations in their existing contracting practices like sprints and paying per sprints, using iterative billing and also, adding pay per delivery features. Furthermore, there were some respondents who proposed to make their contracting practices more focused to collaboration.

5. DISCUSSION

In this chapter a more detailed analysis of the results shown in the data presentation chapter is attempted.

5.1 Data Analysis

5.1.1 General Information for the Respondents

As it was shown in the data presentation chapter, there is a big diversity in the results. Although half of the respondents are from The Netherlands, the remaining half is spread between 13 different countries worldwide as it can be shown in Figure 21. Also, despite the fact that the respondents are categorized between suppliers and buyers, their roles vary between more sectors, such as services, public sector, communications, constructions, manufacturing and energy and utilities. All these can be show in Figure 23. Moreover, Figure 25 shows that most of the respondents have many years of experience and that makes the questionnaire's results a safe ground for analysis.

5.1.2 Introduction in Contracting

In Figure 26 and Figure 27 of the presentation of data, the knowledge of suppliers and buyers on Agile methods and on software development methods is rated as very

knowledgeable. There was an attempt to connect the respondent's years of experience with their knowledge on these methods.

Thus, a standard multiple regression analysis was performed to predict participants' years of experience based on their software development and Agile methods knowledge. A significant regression equation was found (F (2, 76) = 4.329, p= .017) as can be shown in Figure 47.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	20.811	2	10.405	4.329	.017[b]
	Residual	182.683	76	2.404		
	Total	203.494	78			

Figure 47: Regression Analysis

Figure 48 shows the correlation between these variables.

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.416	.812		1.743	.085
	Software development knowledge / These questions refer to your knowledge with software development...-How would you rate your knowledge on Agile Meth	-.027	.194	-.018	-.139	.890
	Software development knowledge / These questions refer to your knowledge with software development...-How would you rate your knowledge on Software D	.577	.222	.328	2.601	.011

Figure 48: Coefficients

This analysis shows that, based on variable B in Figure 48, knowledge on Agile methods is inversely proportional to the years of experience for these respondents, while for the knowledge in general software development the opposite applies. Thus, this tells us that respondents with less years of experience have better knowledge in Agile methods. This is expectable since Agile methods were introduced in 2001 and they are being used in the recent years. Regarding the knowledge on general software development methods, it is increasing when years of experience are also increasing.

In order to identify if there is any difference among the groups of suppliers and buyers regarding their knowledge on software development methods and Agile methods, an analysis of variance (ANOVA) was done. This analysis did not reveal any significant group effects on their knowledge and means that these methods are equally knowledgeable between the groups.

Figure 28 and Figure 29 shows the knowledge of the respondents regarding specific contracting practices. An analysis of variance (ANOVA) was conducted again in order to identify any differences between the groups of suppliers and buyers. Like the previous analysis, it did not reveal any significant group effects on their knowledge and means that these methods are equally knowledgeable between the groups.

Based on the results in Figure 31 which depicts the concerns of suppliers and buyers when contracting in software projects, it is logical that they have different concerns when it comes to a contract. For instance, it can be clearly seen that the biggest concern on the supplier side is an uncertain customer involvement. This is reasonable since in occasions that there is lack of customer involvement in the project, the supplier has more difficulty to develop it (Zijdemans & Stettina, 2014). On the project owner side the biggest concern

is the quality of the end result since he has the biggest interest in the final product. Regarding the other concerns that were listed, both sides presume very important the existence of ambiguous requirements and the budget of the project.
Concerning these challenges, there was an analysis in order to identify differences in the results when it comes to expensive projects. However, there was not any significant difference and the results were similar.

5.1.3 Contracting in Context

The choices of the suppliers and buyers regarding which contracting practice is most suitable when a certain factor is the most critical aspect of the project are explained.

- **Budget**

Figure 32 shows that when budget considered as the most important factor for a project, both suppliers and buyers find Fixed-price-fixed-scope as the most suitable contracting practice. For the buyer's side this can be explained by the presumed idea that this practice has low risk since there is more control in the project's costs (Hoda et al., 2009). However, it was expected that suppliers would have presumed another practice as most suitable since this one is more risky for them as described in section 2.4.2 of the thesis (Zijdemans & Stettina, 2014), (Turner, 2003).

- **Quality**

Figure 33 shows that when quality considered as the most important factor for a project, suppliers find Two phase as most suitable while buyers find Pay per sprint most suitable. According to the preliminary framework (Zijdemans & Stettina, 2014) in these two practices both the supplier and the buyer benefit, so these results are expectable.

- **Time**

Figure 34 shows that when time is the most critical aspect of the project, suppliers find Time & Material as most suitable while buyers find Two phase as most suitable. Both results are expectable based on the literature. According to the preliminary framework (Zijdemans & Stettina, 2014) and on the description of Time & material contracts (Book et al., 2012), this contracting practice is more beneficial for the supplier. On the other hand, Two phase is beneficial for both of the sides (Zijdemans & Stettina, 2014).

- **Ambiguous Requirements**

Figure 35 shows that when ambiguous requirements are the most critical aspect of the project, both suppliers and buyers find Two phase as most suitable. According to Arbogast et al., (2012) Two phased model contracts can be used in order to define high level requirements.

- **Size of the project**

Only 2 respondents out of the 79 consider project size as an important factor of the project and as a result we cannot proceed to further analysis.

- **Uncertain Customer Involvement**

Figure 36 shows that when Uncertain customer involvement is the most critical aspect of the project, suppliers find Collaboration agreement as most suitable contracting practice. For buyers we cannot have safe conclusions due to little response from their side although, from the 2 responses it can be seen that there is an accordance with suppliers considering the same practice as most suitable. This result is expected since this contacting practice can be used in order to improve the collaborative behavior between the supplier and the buyer (Thorup & Jensen, 2009).

- **First Time Collaboration**

Figure 37 shows that when First time collaboration is the most critical aspect of the project, suppliers find Pay per sprint as most suitable while buyers find Two phase as most suitable. Both of these two practices are suitable according to the literature. Pay per sprint acts like a teaser and is used to build trust between the two sides (Hoda et al., 2009) while one of the incentives for Two phased model to be used is the luck of trust (Arbogast et al., 2012).

- **Agile**

Figure 38 shows that when Agile software methods are used for a project, then both suppliers and buyers consider Pay per sprint as most suitable contracting practice. According to Hoda et al., (2009) using Pay per sprint contracts, the suppliers offer agility to buyers in a trial basis which leads to building trust and giving confidence and risk coverage to hesitant customers.

In the tables in Figure 39 and Figure 40 there is a presentation of which contracting practice has the best average suitability for every factor for suppliers and buyers. Additionally, Figure 41 shows the average suitability for all respondents. Moreover, in Figure 42 the suitability of each contracting practice is presented which was calculated for all the affecting factors in a project. Additionally, Figure 41 and Figure 43 are showing the standard deviations whose values make the data valid. From both figures it can be concluded that Pay per sprint and Two phase have a high level of acceptance as the most suitable contacting practices. This is reasonable, since as described in previous sections, Pay per sprints is not an actual contract. It is an arrangement for a trial agreement between the supplier and the buyer which offers an introduction to agility and can lead to the building of trust between the two sides, to a reduction of the risk that lies on their shoulders and finally to an actual and long term agreement. On the other hand, Two phase model is consisted of different contracting methods in certain phases. By this way, building trust is achieved more easily between the two sides when in the beginning of the project there is uncertainty in the project and lack of previous cooperation.

5.1.4 Contracting Challenges

Figure 44 shows the most important challenges for suppliers and buyers when contracting in Agile projects. It can be clearly seen that scoping and collaboration are the most important challenges for both sides. Although, the number of suppliers that consider

scoping a most important challenge is significantly bigger. This is expectable, since suppliers have bigger risk on their shoulders when scope is unclear.

Regarding the challenges, there was an analysis in order to identify differences in the results when it comes to expensive projects. However, there was not any significant difference.

5.1.5 Software Development Background

Based on Figure 45 which depicts the respondents' understanding of discipline, level planning and documentation as appropriate use for agile methods, a standard multiple regression analysis was performed to predict their knowledge on Agile methods. A significant regression equation was found (F (3, 75) = 3.741, p= .015) as it can be seen in Figure 49.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	11.234	3	3.745	3.741	.015[b]
	Residual	75.069	75	1.001		
	Total	86.304	78			

Figure 49: Regression analysis

This significance let us to observe a correlation between the effectiveness of these factors and the knowledge on Agile methods of the respondents. As it can be seen in Figure 50, the coefficients tell us that the consideration of documentation as important for Agile methods is inversely proportional with Agile knowledge of the respondents, while the other two factors are proportional. Thus, respondents who consider documentation very important they have less knowledge in Agile methods than the other respondents.

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	3.003	.790		3.799	.000
	Understanding of Agile Methods.-Do you think discipline is important for an appropriate use of Agile Methods?	.159	.148	.119	1.076	.285
	Understanding of Agile Methods.-Do you think high level planning is important for an appropriate use of Agile Methods?	.280	.145	.213	1.934	.057
	Understanding of Agile Methods.-Do you think documentation is important for an appropriate use of Agile Methods?	-.312	.123	-.276	-2.532	.013

Figure 50: Coefficients

This proves the main characteristics of Agile methods which differentiate themselves from methods that are making an extensive use of documentation, such as waterfall and rely on self-organizing and cross-functional teams (Beck, 2001) as described in section 2.2.3 of the thesis. Additionally, in order for the teams to be functional and with high level of collaboration, discipline is very important.

Based on the results of the efficiency of Agile methods for developing software, as shown in Figure 46, a standard multiple regression analysis was performed to predict the average size of the participants' projects in euros. A significant regression equation was found (F (1, 69) = 0.275, p= .0602) as can be shown in Figure 51.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	.791	1	.791	.275	.602[b]
	Residual	198.645	69	2.879		
	Total	199.437	70			

Figure 51: Regression analysis

Value of *p* tells us the correlation of these variables is insignificant, thus the respondents that are dealing with more expensive projects that the others, do not have more preference in Agile methods and they do not perceive them as more effective.

5.1.6 Agency Theory

As it can be clearly seen in the previous sections in the discussion of results, there are some differences between suppliers and buyers.

Although they have similar knowledge in contracting practices and similar years of experience in the field of software contracting, they have different perceptions of the suitability of certain contracting practices and also of the concerns about contracting.

The reason why they came up with different choices regarding the contracting practices that were presented to them in the questionnaire and also with the concerns, can be explained by the basic idea of Agency Theory.

According to Agency Theory or principal-agent problem there are two entities: the principal and the agent, where the principal hires the agent to perform on the interest and take decisions on behalf of the principal (Laffont & Martimort, 2002).

In this thesis, the principal is the buyer and the agent is the supplier. The problem occurs because sometimes the supplier's (agent) actions are based on the incentives to act on his

own self-interest rather than on the buyer's (principal). This happens because there is information asymmetry between the two entities. For instance, information such as the scoping of a project, the pricing and the collaboration between the two sides is perceived different by each other. For that reason buyers try to create incentives for the suppliers in order their incentives to be aligned and to sign an effective contract.

Figure 52 depicts the way the two entities act in Agency Theory.

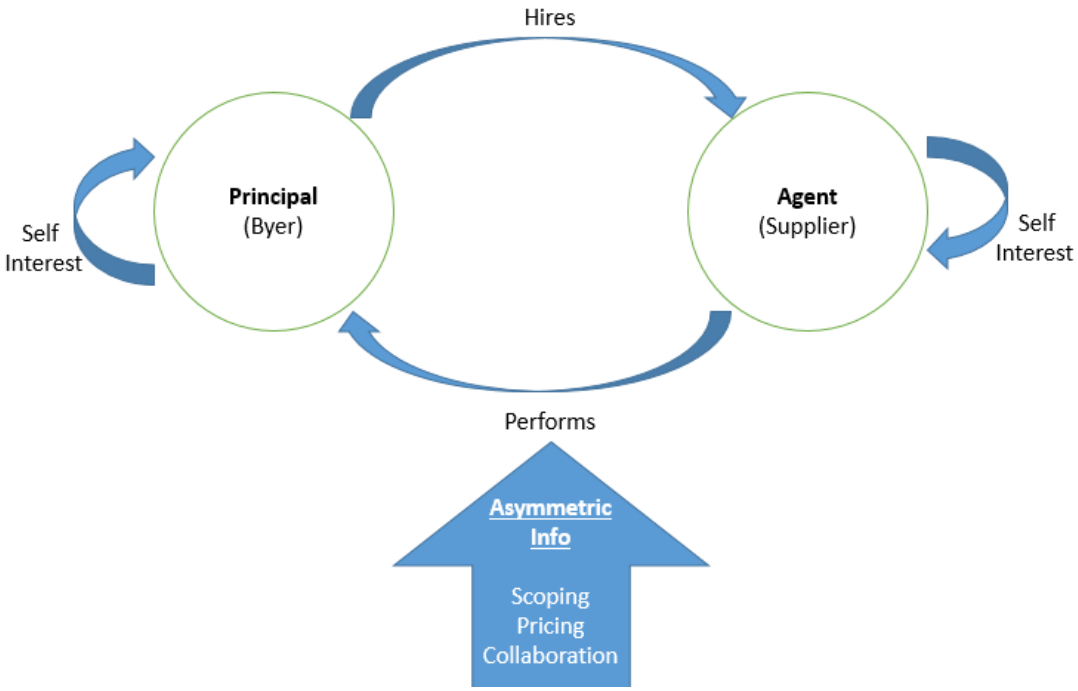

Figure 52: Presentation of supplier and buyer in Agency Theory

5.2 Research Questions

The main target of this thesis is to answer the two research questions that were formulated in the first chapter. Specifically, the two research questions that stated in the first chapter are:

- *What is the current state of knowledge on contracting practices in software development companies and their clients?*

- *What are the preferred contracting practices across buyers and suppliers?*

As shown in the data analysis section, the knowledge of 79 respondents across 14 countries on contracting practices is measured between moderately and very knowledgeable. More specifically, respondents were asked for their knowledge in 10 specific contracting practices that were presented in the thesis. As shown in Figure 29, in 3 of them they rated themselves almost moderate knowledgeable and in the rest they rated themselves almost very knowledgeable.

Moreover, in the tables in Figure 39 and Figure 40 the preferred contracting practices across suppliers and buyers are presented. More specifically, it is shown that despite the fact that there are differences in their choices, both sides have a strong preference in Pay per Sprint and Two phase generally as it is shown for all respondents in Figure 41.

5.3 Threats to Validity

The validity of the data gathered through the questionnaire is a very important factor for a research. According to (Ghauri & Gronhaug, 2010) comprises of two types: internal and external validity. Internal validity examines how well the survey was set up in context of

the research design, while external validity examines whether the results can be generalized.

- Internal Validity

In order to ensure that the design and the context of the questionnaire was adequate, feedback from a legal expert was used and also, the questionnaire was based on other researchers' works and frameworks.

- External Validity

In order to ensure that the results have external validity and can be generalized, the author found the results of the questionnaire valid by checking the standard deviation in the statistical analysis of the data and also, tests in SPSS were performed in order to find that the variables and the correlations from the data are valid too.

6. CONCLUSION

In this thesis the author researches the current state of knowledge and also the preference on contracting practices in software development companies and their clients. A quantitative research using a questionnaire was conducted in order as many respondents as possible to be reached. The statistical analysis that was followed found the results valid. There was a collection of 79 valid responses across 14 countries with an average of 6-15 years of experience in the field of software contracting.

The average size of projects that the respondents are involved is between 100.000 and 499.999 euros. Their knowledge in both software development and Agile methods is measured as very knowledgeable while their familiarization with the specific contracting practices that were asked is measured as moderately and very knowledgeable.

Regarding their preferences on the practices it was found that more popular contracting practices are Pay per sprint and Two phase which show that the majority has a preference on Agility and building trust. This can be also shown by the considered effectiveness of Agile methods which was measured as very effective.

Lastly, regarding the most important challenges in both software contracting and Agile projects, factors such as quality ambiguous requirements, uncertain customer involvement, scoping and collaboration are found higher than factors such as budget or duration and size of the project.

This final conclusion can be reformed to a general statement for this thesis. Most professionals, from both sides of suppliers and buyers are most interested in the collaboration of the two parties in the contract and more general in the process of software development under the contract.

6.1 Future Work

This survey could be used as a base for other more detailed and more concrete guides for contracting practices in software development and Agile methods.

Moreover, it could be a basis for the development of a software tool. By this way a more automated process could be developed in order to be able to measure the knowledge and the concerns of software development contractors and their customers and as a result to propose which contracting practices suit them better.

Lastly, there can be a more detailed research on how the different choices of buyers and suppliers regarding contracting practices and concerning factors can be correlated with Agency Theory.

6.2 Recommendations

As it is stated in the general conclusion part, software development practitioners and their clients can be more effective and have a better result under a contract when they perceive their collaboration and the process of the development more important than using fixed prices or fixed scoping in projects. For that reason, companies could educate their employers in these perspectives in order Agile methods could be more widely acceptable.

APPENDIX

A.1 Email Text Template

Dear Andrea,

My name is Emmanouil Pilios. We are currently working on a *contracting* study about agile software projects. The research is conducted at Leiden University in the Netherlands.

The results of the study might be interesting for you as it aims to help *understanding and improving the relation between contractors and contracting parties*. As such it should be beneficial for the relationship with your clients in software and IT industry.

Would you be able to promote the link to the online questionnaire within your organization? Specifically project development managers and others directly involved in writing and signing project contracts?

All the best,

Manos

Survey on Contracting Practices in traditional and agile software development

Survey link: https://uleidenipa.qualtrics.com/SE/?SID=SV_57w8vQjK81XbDed

The survey is expected to take about 5-10 minutes. The participation is anonymous and voluntary.

If you have any further questions, please send me an email at: e.pilios@umail.leidenuniv.nl

A.2 LinkedIn Post Template

Survey on Contracting Practices in traditional and agile software development

Are you struggling with contracting and finding the right contract type for your software development projects? Join our survey. It takes only 5-10 minutes.

https://uleidenipa.qualtrics.com/SE/?SID=SV_57w8vQjK81XbDed

As part of a research project at Leiden University we are conducting a survey aiming to improve contracting practices in software development and their understanding. Hereby we would like to ask you for 5 to 10 minutes of your time in order to participate to the survey. Please find further information under the survey link.

A.3 Questionnaire

Contracting Practices

Dear Prospective Participant,

This survey is about contracting practices used by software development companies and their clients. The goal is to collect current experiences and challenges in contracting and help improving them.

The participation is anonymous and voluntary. You may find some of the questions below personal, but your responses will be held in strict confidence, and you should notice that we are not even collecting your name. If anything is published as a result of this research, no one reading these publications will ever be able to tell that you participated in this survey. If you choose not to participate, no further action is required from you. It will not affect your current or future relations with the university.

The survey is part of a research project at Leiden University. Your careful attention is important to us. The benefits of participation are: Access to valuable information regarding the current state of knowledge on contracting practices. Building ground for a concrete guide that lists the contracting practices.

Please take your time answering the questions. Some of them might look a bit long but for consistency of the data it is **important** to answer all questions. It takes about **5-10 minutes** to fill out the questionnaire.

General Questions

Which side would you rather identify yourself with?

- ○ Project Owner (Buyer/Client)
- ○ Supplier (Developer/Contractor)

What industrial sector does your organization/department operate in?

- ○ **Agriculture** (e.g. forestry, fishing, mining)
- ○

- **Energy and utilities** (e.g. electricity, oil, gas, water)
- **Manufacturing** (e.g. chemicals, pharmaceuticals, electronics, defence, toys)
- **Services** (e.g. IT, banking, legal, retail, transport)
- **Construction** (e.g. civil engineering)
- **Public sector** (e.g. central government, education, health)
- **Communications** (e.g. advertsising, media, publishing)

What is your role in your organization?

- Project manager
- Product owner
- Product manager
- Scrum master
- Legal advisor
- Development staff (developer/tester/designer)
- Consultant
- Coach
- Other

How many years of experience do you have in the field of software contracting?

- 1-2 years
- 3-5 years
- 6-10 years
- 11-15 years
- 16-20 years
- 21+ years

Introduction Contracting

Software development knowledge
These questions refer to your knowledge with software development projects.

	not at all knowledgeable	slightly knowledgeable	moderately knowledgeable	very knowledgeable	extremely knowledgeable

How would you rate your knowledge on **Software Development**?	○	○	○	○	○
How would you rate your knowledge on **Agile Methods**? (e.g. Scrum)	○	○	○	○	○

Contracting practices knowledge

These questions refer to your knowledge in contracting software development projects and respective contracting practices.

	not at all knowledgeable	slightly knowledgeable	moderately knowledgeable	very knowledgeable	extremely knowledgeable
Fixed-price fixed-scope (*the budget and the scope of the project are fixed*)	○	○	○	○	○
Time & material (*the supplier is paid by the hour*)	○	○	○	○	○
Pay per sprint (*the supplier is paid per iteration, which usually lasts 2 weeks*)	○	○	○	○	○
Target-price (*if the final cost exceeds the pre defined budget, it is shared fairly between the supplier and the contractor. The same happens if there is profit*)	○	○	○	○	○
Cost-reimbursement (*also known as cost-plus, the supplier is paid for all of his allowed expenses up to a set limit plus an additional payment which contains the profit*)	○	○	○	○	○
Exit arrangement (*the contract has predefined points in time where the parties can terminate the project in a controlled manner*)	○	○	○	○	○
Risk buffer (*it is used by the supplier to mitigate the uncertainty*)	○	○	○	○	○

Two phase *(after the first uncertain period of the project, the budget and time can be estimated more accurately)*	○	○	○	○	○
Collaboration agreement *(can be included in a contract when the amount of customer involvement is uncertain or expected to be low)*	○	○	○	○	○
Hire the developer *(Hire development staff directly)*	○	○	○	○	○

What is the average size of your projects in €?

- ○ 0 - 4999€
- ○ 5000 - 9.999€
- ○ 10.000 - 49.999€
- ○ 50.000 - 99.999€
- ○ 100.000 - 499.999€
- ○ 500.000 - 999.999€
- ○ >1.000.000€

What is your biggest concern when contracting software projects?
(you can choose more than one)

- ☐ Budget
- ☐ Quality
- ☐ Time
- ☐ Ambiguous Requirements
- ☐ Project Size
- ☐ Uncertain Customer Involvement
- ☐ First Time Collaboration

Contracting in context

Considering -BUDGET- as the most critical aspect of the project, which contracting practices do you consider suitable?

	not at all suitable	slightly suitable	moderately suitable	very suitable	extremely suitable
Fixed-price fixed-scope *(the budget and the scope of the project are fixed)*	○	○	○	○	○
Time & material *(the supplier is paid by the hour)*	○	○	○	○	○
Pay per sprint *(the supplier is paid per iteration, which usually lasts 2 weeks)*	○	○	○	○	○
Target-price *(if the final cost exceeds the pre defined budget, it is shared fairly between the supplier and the contractor. The same happens if there is profit)*	○	○	○	○	○
Cost-reimbursement *(also known as cost-plus, the supplier is paid for all of his allowed expenses up to a set limit plus an additional payment which contains the profit)*	○	○	○	○	○
Exit arrangement *(the contract has predefined points in time where the parties can terminate the project in a controlled manner)*	○	○	○	○	○
Risk buffer *(it is used by the supplier to mitigate the uncertainty)*	○	○	○	○	○
Two phase *(after the first uncertain period of the project, the budget and time can be estimated more accurately)*	○	○	○	○	○
Collaboration agreement *(can be included in a contract when the amount of customer involvement is uncertain or expected to be low)*	○	○	○	○	○
Hire the developer *(hire developer(s) in-house)*	○	○	○	○	○

Do you have any comments regarding your choice?

Considering -QUALITY- as the most critical aspect of the project, which contracting practices do you consider suitable?

	not at all suitable	slightly suitable	moderately suitable	very suitable	extremely suitable
Fixed-price fixed-scope *(the budget and the scope of the project are fixed)*	○	○	○	○	○
Time & material *(the supplier is paid by the hour)*	○	○	○	○	○
Pay per sprint *(the supplier is paid per iteration, which usually lasts 2 weeks)*	○	○	○	○	○
Target-price *(if the final cost exceeds the pre defined budget, it is shared fairly between the supplier and the contractor. The same happens if there is profit)*	○	○	○	○	○
Cost-reimbursement *(also known as cost-plus, the supplier is paid for all of his allowed expenses up to a set limit plus an additional payment which contains the profit)*	○	○	○	○	○
Exit arrangement *(the contract has predefined points in time where the parties can terminate the project in a controlled manner)*	○	○	○	○	○
Risk buffer *(it is used by the supplier to mitigate the uncertainty)*	○	○	○	○	○
Two phase *(after the first uncertain period of the project, the budget and time can be estimated more accurately)*	○	○	○	○	○
Collaboration agreement *(can be included in a contract when the amount of customer involvement is uncertain or expected to be low)*	○	○	○	○	○
Hire the developer *(hire developer(s) in-house)*	○	○	○	○	○

Do you have any comments regarding your choice?

Considering -TIME- as the most critical aspect of the project, which contracting practices do you consider suitable?

	not at all suitable	slightly suitable	moderately suitable	very suitable	extremely suitable
Fixed-price fixed-scope *(the budget and the scope of the project are fixed)*	○	○	○	○	○
Time & material *(the supplier is paid by the hour)*	○	○	○	○	○
Pay per sprint *(the supplier is paid per iteration, which usually lasts 2 weeks)*	○	○	○	○	○
Target-price *(if the final cost exceeds the pre defined budget, it is shared fairly between the supplier and the contractor. The same happens if there is profit)*	○	○	○	○	○
Cost-reimbursement *(also known as cost-plus, the supplier is paid for all of his allowed expenses up to a set limit plus an additional payment which contains the profit)*	○	○	○	○	○
Exit arrangement *(the contract has predefined points in time where the parties can terminate the project in a controlled manner)*	○	○	○	○	○
Risk buffer *(it is used by the supplier to mitigate the uncertainty)*	○	○	○	○	○
Two phase *(after the first uncertain period of the project, the budget and time can be estimated more accurately)*	○	○	○	○	○
Collaboration agreement *(can be included in a contract when the amount of customer involvement is uncertain or expected to be low)*	○	○	○	○	○
Hire the developer *(hire developer(s) in-house)*	○	○	○	○	○

Do you have any comments regarding your choice?

Considering -AMBIGUOUS REQUIREMENTS- as the most critical aspect of the project, which contracting practices do you consider suitable?

	not at all suitable	slightly suitable	moderately suitable	very suitable	extremely suitable
Fixed-price fixed-scope (*the budget and the scope of the project are fixed*)	○	○	○	○	○
Time & material (*the supplier is paid by the hour*)	○	○	○	○	○
Pay per sprint (*the supplier is paid per iteration, which usually lasts 2 weeks*)	○	○	○	○	○
Target-price (*if the final cost exceeds the pre defined budget, it is shared fairly between the supplier and the contractor. The same happens if there is profit*)	○	○	○	○	○
Cost-reimbursement (*also known as cost-plus, the supplier is paid for all of his allowed expenses up to a set limit plus an additional payment which contains the profit*)	○	○	○	○	○
Exit arrangement (*the contract has predefined points in time where the parties can terminate the project in a controlled manner*)	○	○	○	○	○
Risk buffer (*it is used by the supplier to mitigate the uncertainty*)	○	○	○	○	○
Two phase (*after the first uncertain period of the project, the budget and time can be estimated more accurately*)	○	○	○	○	○
Collaboration agreement (*can be included in a contract when the amount of customer involvement is uncertain or expected to be low*)	○	○	○	○	○
Hire the developer (*hire developer(s) in-house*)	○	○	○	○	○

Do you have any comments regarding your choice?

Considering -LARGE SIZE PROJECTS- as the most critical aspect of the project, which contracting practices do you consider suitable?

	not at all suitable	slightly suitable	moderately suitable	very suitable	extremely suitable
Fixed-price fixed-scope *(the budget and the scope of the project are fixed)*	○	○	○	○	○
Time & material *(the supplier is paid by the hour)*	○	○	○	○	○
Pay per sprint *(the supplier is paid per iteration, which usually lasts 2 weeks)*	○	○	○	○	○
Target-price *(if the final cost exceeds the pre defined budget, it is shared fairly between the supplier and the contractor. The same happens if there is profit)*	○	○	○	○	○
Cost-reimbursement *(also known as cost-plus, the supplier is paid for all of his allowed expenses up to a set limit plus an additional payment which contains the profit)*	○	○	○	○	○
Exit arrangement *(the contract has predefined points in time where the parties can terminate the project in a controlled manner)*	○	○	○	○	○
Risk buffer *(it is used by the supplier to mitigate the uncertainty)*	○	○	○	○	○
Two phase *(after the first uncertain period of the project, the budget and time can be estimated more accurately)*	○	○	○	○	○
Collaboration agreement *(can be included in a contract when the amount of customer involvement is uncertain or expected to be low)*	○	○	○	○	○
Hire the developer *(hire developer(s) in-house)*	○	○	○	○	○

Do you have any comments regarding your choice?

Considering -SMALL SIZE PROJECTS- as the most critical aspect of the project, which contracting practices do you consider suitable?

	not at all suitable	slightly suitable	moderately suitable	very suitable	extremely suitable
Fixed-price fixed-scope (*the budget and the scope of the project are fixed*)	○	○	○	○	○
Time & material (*the supplier is paid by the hour*)	○	○	○	○	○
Pay per sprint (*the supplier is paid per iteration, which usually lasts 2 weeks*)	○	○	○	○	○
Target-price (*if the final cost exceeds the pre defined budget, it is shared fairly between the supplier and the contractor. The same happens if there is profit*)	○	○	○	○	○
Cost-reimbursement (*also known as cost-plus, the supplier is paid for all of his allowed expenses up to a set limit plus an additional payment which contains the profit*)	○	○	○	○	○
Exit arrangement (*the contract has predefined points in time where the parties can terminate the project in a controlled manner*)	○	○	○	○	○
Risk buffer (*it is used by the supplier to mitigate the uncertainty*)	○	○	○	○	○
Two phase (*after the first uncertain period of the project, the budget and time can be estimated more accurately*)	○	○	○	○	○
Collaboration agreement (*can be included in a contract when the amount of customer involvement is uncertain or expected to be low*)	○	○	○	○	○
Hire the developer (*hire developer(s) in-house*)	○	○	○	○	○

Do you have any comments regarding your choice?

Considering -UNCERTAIN CUSTOMER INVOLVEMENT- as the most critical aspect of the project, which contracting practices do you consider suitable?

	not at all suitable	slightly suitable	moderately suitable	very suitable	extremely suitable
Fixed-price fixed-scope *(the budget and the scope of the project are fixed)*	○	○	○	○	○
Time & material *(the supplier is paid by the hour)*	○	○	○	○	○
Pay per sprint *(the supplier is paid per iteration, which usually lasts 2 weeks)*	○	○	○	○	○
Target-price *(if the final cost exceeds the pre defined budget, it is shared fairly between the supplier and the contractor. The same happens if there is profit)*	○	○	○	○	○
Cost-reimbursement *(also known as cost-plus, the supplier is paid for all of his allowed expenses up to a set limit plus an additional payment which contains the profit)*	○	○	○	○	○
Exit arrangement *(the contract has predefined points in time where the parties can terminate the project in a controlled manner)*	○	○	○	○	○
Risk buffer *(it is used by the supplier to mitigate the uncertainty)*	○	○	○	○	○
Two phase *(after the first uncertain period of the project, the budget and time can be estimated more accurately)*	○	○	○	○	○
Collaboration agreement *(can be included in a contract when the amount of customer involvement is uncertain or expected to be low)*	○	○	○	○	○
Hire the developer *(hire developer(s) in-house)*	○	○	○	○	○

Do you have any comments regarding your choice?

Considering -FIRST TIME COLLABORATION- as the most critical aspect of the project, which contracting practices do you consider suitable?

	not at all suitable	slightly suitable	moderately suitable	very suitable	extremely suitable
Fixed-price fixed-scope (*the budget and the scope of the project are fixed*)	○	○	○	○	○
Time & material (*the supplier is paid by the hour*)	○	○	○	○	○
Pay per sprint (*the supplier is paid per iteration, which usually lasts 2 weeks*)	○	○	○	○	○
Target-price (*if the final cost exceeds the pre defined budget, it is shared fairly between the supplier and the contractor. The same happens if there is profit*)	○	○	○	○	○
Cost-reimbursement (*also known as cost-plus, the supplier is paid for all of his allowed expenses up to a set limit plus an additional payment which contains the profit*)	○	○	○	○	○
Exit arrangement (*the contract has predefined points in time where the parties can terminate the project in a controlled manner*)	○	○	○	○	○
Risk buffer (*it is used by the supplier to mitigate the uncertainty*)	○	○	○	○	○
Two phase (*after the first uncertain period of the project, the budget and time can be estimated more accurately*)	○	○	○	○	○
Collaboration agreement (*can be included in a contract when the amount of customer involvement is uncertain or expected to be low*)	○	○	○	○	○
Hire the developer (*hire developer(s) in-house*)	○	○	○	○	○

Do you have any comments regarding your choice?

Considering that your project is executed using AGILE SOFTWARE DEVELOPMENT METHODS, which contracting practices do you consider suitable?

	not at all suitable	slightly suitable	moderately suitable	very suitable	extremely suitable
Fixed-price fixed-scope (*the budget and the scope of the project are fixed*)	○	○	○	○	○
Time & material (*the supplier is paid by the hour*)	○	○	○	○	○
Pay per sprint (*the supplier is paid per iteration, which usually lasts 2 weeks*)	○	○	○	○	○
Target-price (*if the final cost exceeds the pre defined budget, it is shared fairly between the supplier and the contractor. The same happens if there is profit*)	○	○	○	○	○
Cost-reimbursement (*also known as cost-plus, the supplier is paid for all of his allowed expenses up to a set limit plus an additional payment which contains the profit*)	○	○	○	○	○
Exit arrangement (*the contract has predefined points in time where the parties can terminate the project in a controlled manner*)	○	○	○	○	○
Risk buffer (*it is used by the supplier to mitigate the uncertainty*)	○	○	○	○	○
Two phase (*after the first uncertain period of the project, the budget and time can be estimated more accurately*)	○	○	○	○	○
Collaboration agreement (*can be included in a contract when the amount of customer involvement is uncertain or expected to be low*)	○	○	○	○	○
Hire the developer (*hire developer(s) in-house*)	○	○	○	○	○

Do you have any comments regarding your choice?

Contracting Challenges

What would you consider the biggest challenge when contracting in Agile projects?

- ○ Scoping
- ○ Pricing
- ○ Collaboration
- ○ Dispute resolution
- ○ Acceptance

Can you elaborate on the challenges?

How do you think could contracting and contracting practices be improved in software development and procurement projects?

Background Questions Software Development

Understanding of Agile Methods.

	not at all important	slightly important	moderately important	very important	extremely important
Do you think **discipline** is important for an appropriate use of Agile Methods?	○	○	○	○	○
Do you think **high level planning** is important for an appropriate use of Agile Methods?	○	○	○	○	○
Do you think **documentation** is important for an appropriate use of Agile Methods?	○	○	○	○	○

Effectiveness of software development methods

	not at all effective	slightly effective	moderately effective	very effective	extremely effective
How effective do you consider **Agile Methods** for developing software?	○	○	○	○	○
How effective do you consider the **software development method** that is used by your organization?	○	○	○	○	○

Did you choose to apply an Agile project management method (e.g., Scrum)? If yes, why? Did it cater your needs?

Can you think of a brief way that the existing contracting practice could fit the iterative nature of Agile methods?

Do you have any further comments? Is there anything you think is interesting to add?

REFERENCES

Ambler, S. (2006). *Agile Adoption Rate Survey Results: March 2006*. Retrieved from http://www.ambysoft.com/surveys/agileMarch2006.html

Arbogast, T., Larman, C., & Vodde, B. (2012). AGILE CONTRACTS PRIMER. *History*, (c), 1–44.

Beck, K. et al. (2001). *Manifesto for Agile Software Development*. Retrieved October 20, 2014, from http://agilemanifesto.org/

Book, M., Gruhn, V., & Striemer, R. (2012). adVANTAGE: A Fair Pricing Model for Agile Software Development Contracting. *Agile Processes in Software Engineering …*, 1–8. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-30350-0_15

Dischave, D. (2012). *A Waterfall Systems Development Methodology … Seriously?* Retrieved from http://get.syr.edu/news_alt.aspx?recid=401

Eckfeldt, B., Madden, R., & Horowitz, J. (2005). Selling agile: target-cost contracts. *Agile Conference, 2005. …*. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1609816

Extreme programming. (n.d.). Retrieved from http://en.wikipedia.org/wiki/Extreme_programming#cite_note-UPenn49-2

Ghauri, P., & Gronhaug, K. (2010). *Research Methods in Business Studies*.

Hoda, R., Noble, J., & Marshall, S. (2009). Negotiating contracts for agile projects: A practical perspective. *Agile Processes in Software Engineering …*. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-01853-4_25

Hofbauer, J., & Sanders, G. (2008). *Defense Industrial Initiatives Current Issues: Cost-Plus Contracts*. Retrieved from http://csis.org/files/media/csis/pubs/081016_diig_cost_plus.pdf

Kerzner, H. (2009). *Project management: a systems approach to planning, scheduling, and controlling*. *New York*. Retrieved from http://books.google.com/books?hl=en&lr=&id=QgQQC5qRtzgC&oi=fnd&pg=PT18&dq=Project+Management+A+SYSTEMS+APPROACH+TO+PLANNING,+SCHEDULING,+AND+CONTROLLING&ots=C-xCmnG1MQ&sig=bK0-57PodokJlrHPIsNW3IOnJcY\nhttp://books.google.com/books?hl=en&lr=&id=QgQQC5qRtzg

Kettunen, P., & Laanti, M. (2008). Combining agile software projects and large-scale organizational agility. *Software Process: Improvement and …*, (July 2007), 183–193. doi:10.1002/spip

Kwak, Y., & Anbari, F. (2005). *The story of managing projects*. Retrieved from http://discovery.ucl.ac.uk/46752/

Laffont, J.-J., & Martimort, D. (2002). *The Theory of Incentives: The Principal-Agent Model*. *Princeton University Press*.

Larman, C., & Basili, V. (2003). Iterative and incremental development: A brief history. *Computer*. Retrieved from http://www.computer.org/csdl/mags/co/2003/06/r6047.pdf

Luecke, R. (2004). *Managing projects large and small: the fundamental skills for delivering on budget and on time*. Retrieved from http://courses.washington.edu/com585/2006/pub/book_reviews/Managing_Projects_Large_and_Small.doc

McLeod, L., & MacDonell, S. G. (2011). Factors that affect software systems development project outcomes. *ACM Computing Surveys*, *43*, 1–56. doi:10.1145/1978802.1978803

Moløkken-østvold, K., & Furulund, K. M. (2007). The relationship between customer collaboration and software project overruns. *Agile Conference (AGILE), …*. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4293577

Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, *50*(3), 79–83. Retrieved from http://dl.acm.org/citation.cfm?id=1226739

Royce, W. (1970). Managing the development of large software systems. *Proceedings of IEEE WESCON*, (August), 1–9. Retrieved from http://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf

Scheuren, F. (2004). What is a Survey. *American Statistical Association*.

Schwaber, K. (1997). Scrum development process. *Business Object Design and Implementation*, (April 1987), 23. Retrieved from http://link.springer.com/chapter/10.1007/978-1-4471-0947-1_11

Scrum (software development). (n.d.). Retrieved from http://en.wikipedia.org/wiki/Scrum_(software_development)

Stellman, A., & Greene, J. (2005). *Applied software project management*. Retrieved from http://books.google.com/books?hl=en&lr=&id=IYdJocLVa8wC&oi=fnd&pg=PR1&dq=Applied+software+project+management&ots=zY54fi2n7g&sig=LAXNMevRndrsm66ELdeoESd7SK0

Stevens, P. (2009a). *10 Contracts for your next Agile Software Project*. Retrieved October 25, 2014, from http://agilesoftwaredevelopment.com/blog/peterstev/10-agile-contracts

Stevens, P. (2009b). *Contracting for Agile Software Projects, Part 1*. Retrieved from http://agilesoftwaredevelopment.com/blog/peterstev/contracting-agile-software-projects

Szalvay, V. (2004). An introduction to agile software development. *Danube Technologies*, (June). Retrieved from http://www.danube.com/docs/Intro_to_Agile.pdf

Thorup, L., & Jensen, B. (2009). Collaborative Agile Contracts. *2009 Agile Conference*, 195–200. doi:10.1109/AGILE.2009.19

Turner, J. (2003). *Contracting for project management*. Retrieved from http://books.google.com/books?hl=en&lr=&id=iwHqWXRIYjgC&oi=fnd&pg=PP1&dq=Contracting+for+Project+Management&ots=vmIiRFjg1e&sig=wqHfHKbmEdX2foyo3wkeNUMfsEc

Turner, J., & Müller, R. (2003). On the nature of the project as a temporary organization. *International Journal of Project Management*, *21*, 1–8. Retrieved from http://www.sciencedirect.com/science/article/pii/S0263786302000200

Venkatesh, J., Cherurveettil, P., & Post, J. (n.d.). Fixed Bid Pricing Expectations in Agile Models, *2*(2).

VersionOne. (2014). *8th annual state of agile survey*. Retrieved from http://stateofagile.versionone.com/

Wedutenko, A., & Watson, L. (2012). *Have you planned your exit? Exit management of an outsourcing agreement*. *08 November*. Retrieved April 1, 2015, from http://www.claytonutz.com/publications/edition/08_november_2012/20121108/have_you_planned_your_exit_exit_management_of_an_outsourcing_agreement.page

Williams, L. (2012). What agile teams think of agile principles. *Communications of the ACM*, 0–5. Retrieved from http://dl.acm.org/citation.cfm?id=2133823

Zijdemans, S., & Stettina, C. (2014). Contracting in Agile Software Projects: State of Art and How to Understand It. *Agile Processes in Software Engineering and …*. Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-06862-6_6