UNIVERSITETET I BERGEN
Det samfunnsvitenskapelige fakultet

MASTER THESIS

Software estimation process: a comparison of the estimation practice between Norway and Spain

Author
Paul Salaberria

Supervised by
Solveig Bjørnestad

December 1, 2014

Abstract

This research presents a comparative study regarding the current state of the software estimation practice in Norway and Spain. A survey was conducted within the software industry in both countries, and the results are analyzed and discussed in this thesis. Significant differences are found in how companies in these two countries estimate software projects. Estimation methods, units, agile techniques, participants and their respective roles, as well as to what extent the customer can influence the estimation process, are some of the main topics presented. The selection of development methods might have been a key factor behind the differences we find in the estimation process between the company samples in Norway and Spain.

Acknowledgments

This paper is written as the final thesis of my Master's Degree in Information Science at the University of Bergen, and will conclude my studies in higher education.

I want to thank my supervisor Solveig Bjørnestad for her advice and guidance during this period.

I am also grateful to my friend Celine for the support and encouragement she offered me from the very beginning through the finalization of the paper.

Finally I would like to thank all my friends who have made my stay in Bergen memorable.

Contents

List of Figures

List of Tables

1 Introduction

Software estimation is one of the most substantial challenges we face when carrying out a software project. Forecasting the size or the effort needed in order to accomplish a project, is a challenging process which does not always end successfully.

Software projects are known for exceeding their original estimates. A review of surveys on software effort estimation shows that more than 60% of the projects experience effort and schedule overruns (Moløkken-Østvold and Jørgensen, 2003).

In order to solve the problem of estimation, many techniques and approaches have been put into practice (Jørgensen and Shepperd, 2007), but it does not seem to be a consensus on which approach is the most accurate.

In this thesis, I will analyze the software estimation process in both Norway and Spain, in order to get an overview of the current state of the estimation practice in these countries.

1.1 Motivation for Research

The idea for this research was first introduced by my supervisor. As I started to read various articles about software estimation, my interest in the topic grew.

One of the main reasons for this increased interest was the close relation the topic has with my studies in Business Informatics from the University of Basque Country. These included different courses in economics and business management.

In software projects, estimation is one of the main activities of the business process. Forecasting the size, and as a consequence the cost, of a project is necessary if we want to plan needed resources for a project.

My enthusiasm about economics also positively affected my decision to write this thesis.

Nonetheless, the topic was not new to me. I did have previous experience working in agile projects, in which estimating tasks was a usual practice. However, the existence of different estimation methods and techniques of which I had never heard before intrigued my curiosity. And above all, I wondered how these methods and techniques were being used in the software industry.

The reason why I will compare the estimation process between Norway and Spain has it's roots in my background. I am from the Basque Country, which is located in the north of Spain, and thus my interest in the IT business started there. I moved to Norway the last year of my Bachelor as an exchange student, and since then I have settled here. I am interested in finding out more about how the estimation process, methods, and techniques differ between Norway and Spain.

1.2 Research Questions

This section presents my research questions. Most of the questions are concerned with the current estimation methods, and the role of the participants when estimating.

Software estimation can be carried out by using many different estimation methods. These methods can be combined in order to generate more accurate estimates. Having knowledge about the current state of the estimation practices will be the main focus of my research project.

The people who participate in the estimation process varies from project to project. Project managers, customers, and developers are some of the examples. I will investigate how the estimating teams are composed, and which roles participants play.

Continuous customer collaboration during the development process is one of the keys of the Agile methodologies. It is proven that the active customer involvement tends to give better results at the time of estimating (Furulund, 2007). In traditional development, where a detailed project planning is done in the beginning, it seems to be easier to give long term estimates. On the other hand, the iterative way of planning in Agile methods makes the project effort more difficult to predict in the early stages. The customer needs these estimates as soon as possible to prepare the budget, and there has to be an agreement between the customer and the development team in order to start working on the project. I find this critical process very interesting, and I will analyze the role of the customer, and the influence he has when making decisions.

These are my research questions.

RQ1 Which software effort estimation methods are Norwegian companies using?

RQ2 Do they combine different methods?

RQ3 Who takes part in the estimation process, and what is the role of each participant?

RQ4 To what extent can the customer influence the estimation process?

RQ5 Do Spanish companies estimate in the same way as Norwegian companies do?

2 Literature and Theory

In this section I will present literature and theories that are relevant to my research problem. I will first introduce the software development process, describing its activities and models. After this, I will go deeper into the topic of software estimation, and I will analyze the different software effort estimation approaches. An explanation of how to measure effort estimation accuracy in software projects will follow. In addition, an introduction to the agile planning process is presented, including techniques and best practices. To conclude this chapter, related work will be exposed.

2.1 Software Development Process

A software development process is a structure imposed on the development of a software product. Different software development methodologies can be used by these processes with the purpose of describing how to carry out the activities into the process.

2.1.1 Software development activities

The activities that most development processes include are planning, implementation, testing, documenting, deployment and maintenance (Boehm, 1988; Royce, 1970). The international standard for software life-cycle processes ISO/IEC 12207 tries to define a guideline for software development processes. Each process is described below.

2.1.1.1 Planning

A requirements specification is needed in order to start developing the software system, where the behavior of the system to be developed is described and specified.

In addition, the feasibility of the project has to be analyzed, where the effort and cost of the project may be estimated using different methods I am going to describe later.

The planning activity has as a main goal the design of the product. Ralph and Wand (2009) defined "design" as follows:

> "(noun) a specification of an object, manifested by some agent, intended to accomplish goals, in a particular environment, using a set of primitive components, satisfying a set of requirements, subject to some constraints;
>
> (verb, transitive) to create a design, in an environment (where the designer operates)" (p. 108)

2.1.1.2 Implementation

The part of the development process where developers program or write the code for the system.

2.1.1.3 Testing

Software testing is the process of validating and verifying that a software product meets the specified requirements, and satisfies the needs of the stakeholders.

Depending on the testing method employed, the software testing can be done at different points of the development process. According to Bertolino (2001), there are two classifications of testing levels. The first one uses the target of the test, and differentiates between unit testing, integration testing, and system testing. The other classification of testing levels is based on the testing objective. Acceptance testing, installation testing, alpha and beta testing, and regression testing are some of the examples.

The chosen development method can also affect the testing process. In traditional models the testing normally occurs after the implementation. In contrast, some newer agile approaches, for example Extreme Programming, include a practice where the unit tests are prepared before coding. This last practice is known as Test Driven Development (Beck, 2003).

2.1.1.4 Documenting

There are different types of documentation in software development. Requirements, architecture and design, technical documentation, and end-user manuals are some of these.

The internal design of the software is documented with the purpose of facilitating further development and maintenance. The amount of internal documentation to be generated will be determined by the chosen development model. Agile models normally generate less documentation than older models.

On the other hand, end-user documentation, or external documentation, is based on manuals or guides for the end-user, system administrators and support staff.

Documenting may include the writing of an Application Programming Interface (API), which can be auto-generated from source code comments, and presented in HyperText Markup Language (Friendly, 1996; Kramer, 1999). An API can be used both internally and externally.

2.1.1.5 Deployment and maintenance

Once the product is appropriately tested, approved for release, and sold or distributed into a production environment, the project starts the deployment stage. The main activities here are the installation and customization of the system.

In addition, software training may be performed in order to teach the customer to use the system and increase effectiveness when using it (Olfman and Bostrom, 2008; Webster and Martocchio, 1995).

Software maintenance is one of the most common activities in software engineering. It is the process of improving and optimizing the software product after been delivered to the end user where the main feature is the correction of faults. The international standard for software maintenance is ISO/IEC 14764.

2.1.2 Software development models

Software development models describe different ways to accomplish the activities of the development process. Different models have been created during the history of software development. The waterfall model (Royce, 1970), the spiral model (Boehm, 1988), or the agile methodologies such as XP (Beck, 1999) and Scrum (Schwaber and Beedle, 2002) are some of the most well-known models. The development team may choose a development model which can fit better with the project. A combination of models can be applied in order to satisfy their needs (Highsmith, 2000; Fowler, 2001; Boehm, 2002).

2.1.2.1 Waterfall model

The traditional waterfall model is a sequential model which follows a linear structure (Royce, 1970). It is based on the activities or phases discussed above, but with a slightly different naming. The activities in the waterfall model are requirements specification, analysis, program design, coding, testing and operations. Each phase should be finished before the next one starts. However, in practice, activities overlap and feed information to each other (Sommerville, 2004). See the original waterfall model in Figure 1.

The waterfall model was the first model presented by Royce (1970) in his article. However, he didn't really trust this model, and therefore came up with several others in the same article.

This is a quote from Royce (1970), being aware of the weaknesses of his model:

> "I believe in this concept, but the implementation described above is risky and invites failure."

As Royce (1970) explained, after the testing phase the requirements might need to be changed in order to fix any existing problem. This brings the development process back to the origin, which leads to a possible schedule and/or cost overrun of up to 100%. Figure 2 shows the problem of the waterfall model.

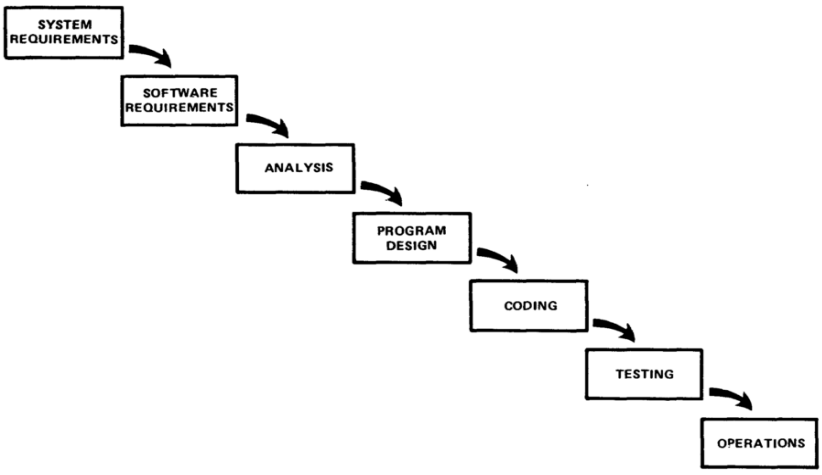


Figure 1 – The waterfall model presented by Royce (1970).

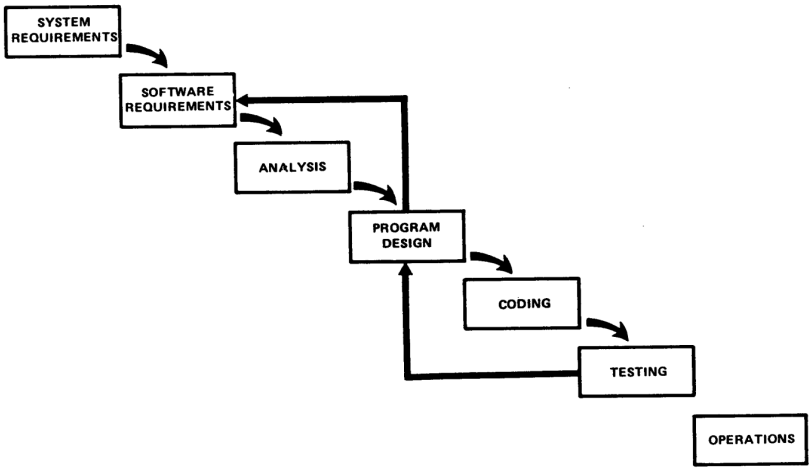


Figure 2 – The problem of the waterfall model presented by Royce (1970).

2.1.2.2 Spiral model

The spiral model of the software process (Figure 3) was originally designed by Boehm (1988). Instead of representing the software process as a linear sequence of activities, the process is represented as a spiral. Each loop in the spiral represents a phase of the software process. In the first loop, the most inner loop, the feasibility of the system is analyzed, in the next loop the requirements are specified, in the next one the system is designed, and the process continues with new loops until the system is delivered. Each loop in the spiral is divided into four sectors. Sommerville (2004) described them in this way:

1. *Goal setting* Specific goals for the current stage of the project are defined. Constraints are identified, and a detailed management plan is generated. Project risks are recognized. Depending on these risks alternative strategies may be planned.

2. *Risk assessment and reduction* A thorough analysis is conducted for each of the detected project risks. Different actions are chosen in order to reduce the risk. For example, a prototype may be developed when there is a risk that the requirements are not appropriate.

3. *Development and validation* Once risks are evaluated, a development model for the product is chosen.

4. *Planning* The project is reviewed and a decision has to be taken about whether to continue or not with a further loop of the spiral. In case the decision is to continue, plans for the next step of the project are drawn up.

The explicit recognition of risk makes this model special from other models.

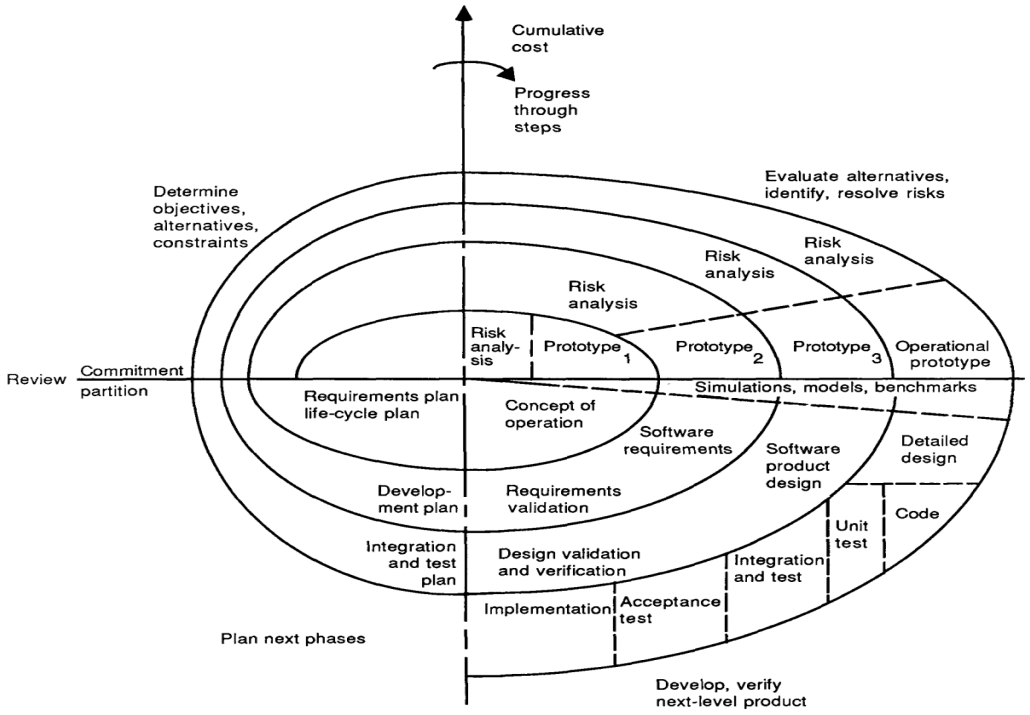


Figure 3 – Spiral model of the software process (Boehm, 1988).

2.1.2.3 Iterative and incremental development

The iterative development of software is a software development model based on iterations (Larman and Basili, 2003). In every iteration small amount of code is added to the product, helping in the detection of errors and preventing future system failures.

2.1.2.4 Agile software development

Agile software development makes reference to a group of software development methods based on iterative and incremental development, where requirements are set by the customer or product owner, and solutions are decided between self-organizing, cross-functional teams. These methods show a more people-centered viewpoint than traditional approaches.

The term "agile" was introduced in the Agile Manifesto (Beck et al., 2001), where they 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

Agile development encourages adaptive planning, evolutionary requirements and development, and a time-boxed iterative approach (Larman, 2004).

The use of short iterations lead to a continuous feedback from the customer and stakeholders, which makes iterative development encourage rapid and flexible response to change (Williams and Cockburn, 2003).

There are many different agile methods, XP (Extreme Programming) and Scrum are the ones with most followers and practitioners (Salo and Abrahamsson, 2008; Rodríguez et al., 2012). Dybå and Dingsøyr (2008) described the main agile development methods in their systematic review about empirical studies of agile software development (Figure 4).

2.2 Software Effort/Cost Estimation

Software development effort estimation is the process of forecasting the required effort for developing a software product.

The cost can be derived from the required effort and resources. Time, salaries, and software tools are just some examples of factors affecting the cost of a project.

Agile method	Description
Crystal methodologies	A family of methods for co-located teams of different sizes and criticality: Clear, Yellow, Orange, Red, Blue. The most agile method, Crystal Clear, focuses on communication in small teams developing software that is not life-critical. Clear development has seven characteristics: frequent delivery, reflective improvement, osmotic communication, personal safety, focus, easy access to expert users, and requirements for the technical environment
Dynamic software development method (DSDM)	Divides projects in three phases: pre-project, project life-cycle, and post project. Nine principles underlie DSDM: user involvement, empowering the project team, frequent delivery, addressing current business needs, iterative and incremental development, allow for reversing changes, high-level scope being fixed before project starts, testing throughout the lifecycle, and efficient and effective communication
Feature-driven development	Combines model-driven and agile development with emphasis on initial object model, division of work in features, and iterative design for each feature. Claims to be suitable for the development of critical systems. An iteration of a feature consists of two phases: design and development
Lean software development	An adaptation of principles from lean production and, in particular, the Toyota production system to software development. Consists of seven principles: eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity, and see the whole
Scrum	Focuses on project management in situations where it is difficult to plan ahead, with mechanisms for "empirical process control"; where feedback loops constitute the core element. Software is developed by a self-organizing team in increments (called "sprints"), starting with planning and ending with a review. Features to be implemented in the system are registered in a backlog. Then, the product owner decides which backlog items should be developed in the following sprint. Team members coordinate their work in a daily stand-up meeting. One team member, the scrum master, is in charge of solving problems that stop the team from working effectively
Extreme programming (XP; XP2)	Focuses on best practice for development. Consists of twelve practices: the planning game, small releases, metaphor, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, 40-h week, on-site customers, and coding standards. The revised "XP2" consists of the following "primary practices": sit together, whole team, informative workspace, energized work, pair programming, stories, weekly cycle, quarterly cycle, slack, 10-minute build, continuous integration, test-first programming, and incremental design. There are also 11 "corollary practices"

Figure 4 – Description of main agile development methods (Dybå and Dingsøyr, 2008).

Customers need effort and cost estimates of software projects for their investment analyses and budgets, while project managers use effort estimates for planning and controlling the software development process.

Several effort estimation approaches have been used in order to find the most accurate estimation method for a software project. Expert judgment, formal models and analogy are some of these approaches. However, there is not yet a perfect method that works for every project.

2.2.1 Software size

The size of the software is one of the basic factors that affect the effort and cost of the software projects.

Counting the source lines of code (LOC) has been the most common method for measuring the size of a software product among practitioners (Parthasarathy, 2003).

Another sizing method is analyzing the size of a software product in terms of its functional characteristics. The functional measurement of the software system can be done in early stages of the development process. It is easier to predict functionality than predicting the amount of LOC early in the project. Functional features can be measured using Function Points (FP).

The term FP was introduced by Albrecht (1979) when working at IBM. A FP is a unit of measurement that describes the amount of functionality provided by an information system to a user.

Gollapudi (2004) wrote an interesting article where he discusses the advantages and disadvantages of using lines of code and Function Points as sizing measures.

Among the advantages of using LOC, we find the ease of automating the counting of the lines, and the fact of being an intuitive and easy to understand metric.

However, it has even more disadvantages, which make the use of Function Points a better choice. Lack of accountability, lack of cohesion with functionality, no meaning in Object Oriented development, difference in developers' level of experience, or the difference of programming languages are some of these disadvantages.

According to Gollapudi, Function Point analysis provides the best objective method for sizing software projects, and for managing the size during development." (Gollapudi, 2004).

When measuring software size using Function Points, we find more advantages than disadvantages. FPs measure software from a functional perspective, which makes the size of the software easier to predict in early stages. The number of FPs for a project, no matter which development method or language have been used, will always be constant. This makes it useful for comparing productivity of different languages, environments, or tools within organizations. If a part of the code is going to be reused from an existing application or library, then the value of FPs of the reusable code will be deducted from the total count of FPs for the project. From the customer's point of view, Function Points are easy to understand due to their relation with the system requirements, which leads to an ease of contract negotiations. Function Point analysis is also good at handling volatility in the requirements. Whenever a feature is added or removed from the project plan, the FP total will be updated accordingly. Function Point analysis also allows us to use historical data in order to compare and predict estimates based on similar projects.

The downside of using Function Points instead of lines of code is that Function Points have to be counted manually, requiring experienced people and a high level of detail in the requirements, in order to generate accurate estimates.

2.2.2 Effort estimation approaches

Software effort estimation approaches have been categorized in many different ways (Boehm, 1981; Briand and Wieczorek, 2002; Jørgensen and Shepperd, 2007), but all of them distinguish expert judgment based methods from the ones based on formal estimation models. Jørgensen et al. (2009) debated the use of expert judgment and formal models trying to help software

project managers figure out under what conditions which method would be best. A combination of expert and formal model-based estimation techniques can be used in order to make a more accurate estimation.

One of the most recent classifications of effort estimation approaches is the one provided by Jørgensen and Shepperd (2007) in which the estimation approaches are sorted into twelve categories. Most relevant ones are described below, including some of the most known technique or models.

2.2.2.1 Parametric models

Parametric or algorithmic models are based on mathematical equations. These can be based on theory or collected data.

Most of the parametric cost models use regression techniques (Boehm et al., 2000).

COCOMO: The Constructive Cost Model (COCOMO) is an algorithmic model created by Boehm (1981). It is valid for estimating effort, cost and schedule for software projects. The model uses a basic regression formula, and takes parameters from historical project data and the future project to be estimated.

Basic COCOMO estimates software development effort and cost in function of the size. The estimated size has to be in thousands of source lines of code (KLOC).

In basic COCOMO the equations are,

$$\mathrm{E} = a * (\mathrm{KLOC})^b \quad \text{[man-months]}$$

$$\mathrm{D} = c * (\mathrm{E})^d \quad \text{[months]}$$

$$\mathrm{P} = \frac{\mathrm{E}}{\mathrm{D}} \quad \text{[people]}$$

where E is the effort applied in man-months, D is the development time in months, and P is the number of people required. The coefficients a,b,c and d change for different development modes such as well understood applications, more complex semidetached systems or embedded systems.

- For simple, well understood applications, $a = 2.4, b = 1.05, c = 2.5, d = 0.38$
- For more complex semidetached systems, $a = 3.0, b = 1.15, c = 2.5, d = 0.35$

- For embedded systems, $a = 3.6, b = 1.20, c = 2.5, d = 0.32$

This model is simple and easy to use, and it can only be used as a rough estimate because many affecting factors are not considered.

2.2.2.2 Analogy

In Analogy-based reasoning estimation methods, the new project to be estimated is compared with already finished similar projects. The gathered data of completed projects is necessary for the estimation.

Shepperd et al. (1996) and Shepperd and Schofield (1997) proved the validity of analogy-based estimation comparing it with different regression methods. They used a tool called ANGEL (ANaloGy softwarE tooL), developed by the authors, where they included several existing data sets with information about many completed projects. Their results showed that estimates based on analogy give more accurate results than regression-based estimates in most of the cases.

However, this technique has some disadvantages. As with algorithmic models, it is not clear for instance the effect of old data points. Companies develop and introduce new technologies over the years, which makes old data points become obsolete and therefore increasingly misleading.

2.2.2.3 Expert judgment

In expert judgment-based techniques one of more experts use their experience and knowledge to estimate the effort of a software project.

Jørgensen (2004, 2007*b*) defined judgment-based effort estimation to be based on an intuition-based final step. In other words, the last or final estimate is based on expert judgment. He considered as experts all individuals with competence in estimating software development effort, including software developers and students with previous experience.

A popular technique that uses expert judgment is the Wideband Delphi (Boehm, 1981). These are the steps you need to follow when using the mentioned technique:

1. The specifications and an estimation form are presented to each expert by the coordinator.
2. Coordinator calls a group meeting in which the experts discuss estimation issues with each other.
3. Forms are filled out anonymously by the experts.
4. A summary of the estimates is prepared and presented by the coordinator.

5. Coordinator calls a group meeting where the experts discuss the sections in which their estimates vary widely.
6. Experts fill out forms again, and steps 4 to 6 are iterated until there is an agreement in the estimates.

2.2.2.4 Function point analysis

Another size related approach is based on FPs. There are several recognized standards for sizing software based on FPs, being ISO/IEC 20926, the standard maintained by the IFPUG (The International Function Point Users Group), the most popular one among practitioners.

A variation of FPA is the Use Case Points (UCP) method, a method to estimate effort in man-hours taking the functional requirements of a system in terms of number of use cases as a parameter. Karner (1993) developed this method for estimating the size of object oriented software systems.

2.2.2.5 Neural network

Artificial neural networks are element processing nets that are able to learn the mapping between input and output data. Neural networks are adaptive systems that change their structure during a learning period based on given sample data. They try to solve complex input-output data relationships.

Many studies have tested the ability of neural networks in estimating software development effort, and have validated their effectiveness (Finnie et al., 1997; Gray and MacDonell, 1997; Shepperd and Kadoda, 2001; Heiat, 2002; Park and Baek, 2008).

2.2.2.6 Bayesian

Bayesian analysis is an inductive reasoning mode, which has been used in different scientific areas. A feature of the Bayesian approach is that it allows the use of both sample data and expert judgment information (Boehm et al., 2000).

Different Bayesian models have been created for estimating the software effort (Van Koten and Gray, 2006).

Pendharkar et al. (2005) compared the performance of the Bayesian models with neural network and regression tree forecasting models and showed that the Bayesian models are competitive models for forecasting software development effort. Bayesian Models are capable of providing uncertainty in the estimates, as well as handling missing data.

2.2.2.7 Combination of estimates

More than one approach can be used for estimating the same project. For example, applying expert judgment to estimates retrieved from a regression based model or a group estimation.

Combination of estimates applying different approaches improves the estimation accuracy (Winkler, 1989; Jørgensen et al., 2009).

2.2.3 Estimation accuracy

When trying to compare the effectiveness of the different methods, two common accuracy measures are the Mean Magnitude Relative Error (MMRE) and the PRED, both of them introduced by Conte et al. (1985).

$$\text{MRE} = \frac{|\text{Actual effort} - \text{Estimated effort}|}{\text{Actual effort}}$$

$$\text{MMRE} = \frac{1}{n}\sum_{i=1}^{n}\text{MRE}$$

$$\text{PRED}(r) = \frac{k}{n}$$

where k is the number of projects in a set of n projects whose MRE is $<= r$.

However, the MRE-based accuracy measures have been criticized, and a variety of measures have been proposed as alternative (Jørgensen, 2007*a*). Among these alternatives we have the Mean Balanced Relative Error (MBRE), Weighted Mean of Quartiles of relative errors (WMQ) or Mean Variation from Estimate (MVFE).

2.3 Agile Estimating and Planning

Cohn (2005) emphasizes the difference between planning and plans in his book. While plans are documents or figures, planning is an activity. He describes agile planning as an ongoing feature of the project, composed by easily changeable plans.

2.3.1 Planning levels

Agile teams complete a progressive planning process by planning at three different levels of the project. These levels are the release, the iteration, and the current day (Cohn, 2005).

Release planning includes user stories to be developed in the new release. It determines the scope, schedule, and resources for a project. The release plan is created at the beginning of the project, and has to be updated at the beginning of every iteration to reflect the current state of the project.

In the iteration planning, based on the work done in the last iteration, the user stories for the coming iteration are selected. The product owner will prioritize the user stories.

Also, there is a daily planning in Agile, normally based on daily stand-up meetings. The developers review what they have done, and plan what they are going to do next.

2.3.2 Estimating size and duration

In Agile, as well as in traditional development methods, estimates of size differ from estimates of duration. However, estimating duration requires estimating size first. See the process of estimating duration of an agile software project in Figure 5.

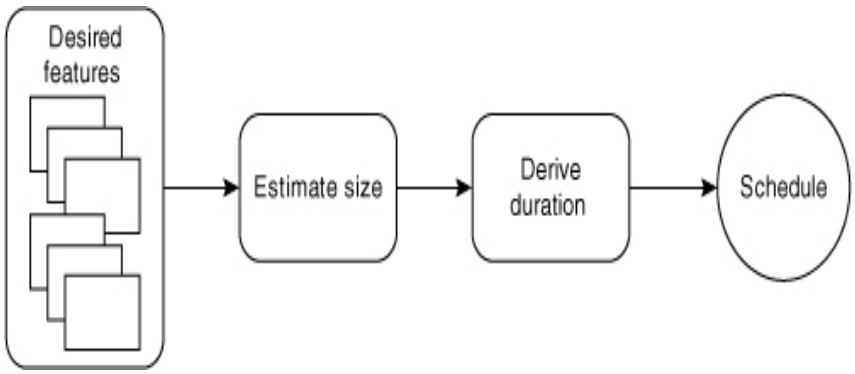


Figure 5 – Process of estimating duration of an agile project (Cohn, 2005).

The two alternative measure units of size for agile projects are story points and ideal days.

2.3.2.1 Story points

"Story points are a unit of measure for expressing the overall size of a user story, feature, or other piece of work" (Cohn, 2005, p 36). The story points are just relative values for measuring the effort for developing the user story. For example, a user story with eight story points is twice as big, or complex as a user story with four points.

Even though story points are not related to time, it is possible to estimate the duration using velocity.

2.3.2.2 Velocity

Velocity is a measure to determine the productivity of the team in an iteration, or how many story points the team is able to complete within an iteration. The mean velocity shows the average velocity of the project. This is the formula for calculating average velocity:

$$MV_n = \frac{1}{n} \sum_{i=1}^{n} V_i$$

where:

- $n =$ current iteration
- $V_i =$ Velocity of iteration i (number of story points achieved in the iteration)
- $MV_n =$ Average velocity for the n first iterations

We can estimate the total size of the project by summing the story points of all the user stories of a project. If we divide the size by velocity of the team we will have the estimated number of iterations needed to accomplish the project.

During the project the velocity helps us to recalculate the number of iterations, and to change the release plan if needed. Also it is an important parameter for the iteration planning, where the product owner cannot choose user stories exceeding the velocity of the team.

2.3.2.3 Ideal days

"Ideal time is the amount of time that something takes when stripped of all peripheral activities" (Cohn, 2005, p 43).

Estimating in ideal days is another size measure. This involves estimating only the time on which developers will work on the user stories without any distraction.

As it is a size measure, we can derive the duration of the project using velocity in the same way as we do with story points.

2.3.3 Estimation techniques

Cohn (2005) says that estimates have to be shared, in other words, the team who will do the work should estimate the user stories. Estimates are not created by a single expert.

Expert opinion, analogy, and disaggregation (splitting a story into smaller pieces) are the three main techniques for estimating.

2.3.3.1 Planning poker

Planning poker, a card game introduced by Grenning (2002), is the most accepted estimation technique for agile software development. This game combines expert opinion, analogy and disaggregation.

All the developers on the team participate in the planning poker. On agile projects normally the teams should be composed of less than 10 people. Otherwise, the team is going to be split into smaller groups to play the game. The product owner attends the planning poker, but cannot estimate.

At the estimation meeting, each participant has one deck of cards. All decks contain the same cards.

The meeting proceeds as follows:

1. The product owner reads the user story. There is a discussion clarifying the story if necessary.
2. Each developer chooses their estimate laying a card face down.
3. Everyone calls their cards at the same time by turning them over.
4. If there is agreement, the story is estimated. Write it down and start with the next story. If there is disagreement on the estimates, the players with highest and lowest estimates argue their choices and discuss.
5. Repeat the estimation process until a consensus is reached.

2.3.3.2 Estimation Scale

The story points represent abstract size of effort, and there is a need to follow some kind of scale for estimating them. As the measures are relative, the values in the scale can be numbers, or any kind of objects that are easy to compare. Some of the most successful scales are:

- Fibonacci sequence (1, 2, 3, 5, 8, 13, 21, 34,...)
- Other numeric scales (1 to 10) or (1, 2, 4, 8)
- T-shirt sizes (XS, S, M, L, XL, XXL, XXXL)

2.3.3.3 Wall estimation

Planning poker is a good technique for estimating user stories. However, in big projects where the backlog contains hundreds of user stories, the process of estimating these one by one might take too long. Therefore, in many cases, it is necessary to find a faster way.

Wall estimation, an agile estimation technique documented by Lacey (2012), appears to be a good solution to the previous problem.

Wall estimation allows teams to estimate stories in a relative way, initially without having to define the size of the stories in Story Points. It is also useful for the stakeholders at the time of prioritizing a large group of stories, where instead of having to decide if one story is slightly more important than another, they can sort and prioritize the stories in a more general way.

The team and the stakeholders carry out a meeting for about two to six hours, where the team puts the stories on the wall. They are sorted left to right depending on their size (left means smaller, right larger), and the stakeholders prioritize them from bottom to top, bottom meaning lower priority and top meaning higher priority.

In order to have more precise estimates, the team can now draw vertical lines for grouping the stories and assigning story points to them. It is also possible to use a different estimation unit rather than story points if preferred.

The result of the meeting should look like the wall in Figure 6, where the wall is separated in four quadrants based on size and priority of the stories. The stories in the top left are small and of high priority, and will therefore go to the top of the product backlog. The stories in the top right are large and high priority. Thus, these stories need to be split as soon as possible in order to include them in the next sprints. The lower left quadrant contains small stories with low priority. These will be at the bottom of the backlog. Finally, we have the lower right quadrant, which includes large stories that are low in priority. They will need to be broken down into smaller stories, but not until the stories with higher priority are done.

2.4 Related Work

This section contains summaries of relevant articles and surveys related to my work, as well as the scarce research about the customer's role in the estimation process.

2.4.1 Effort estimation surveys

Several surveys have been done related to effort estimation. Moløkken-Østvold and Jørgensen (2003) reviewed the most relevant software effort estimation surveys until date. Expert judgment-based estimation methods happened to be the most used ones in the software industry. As a

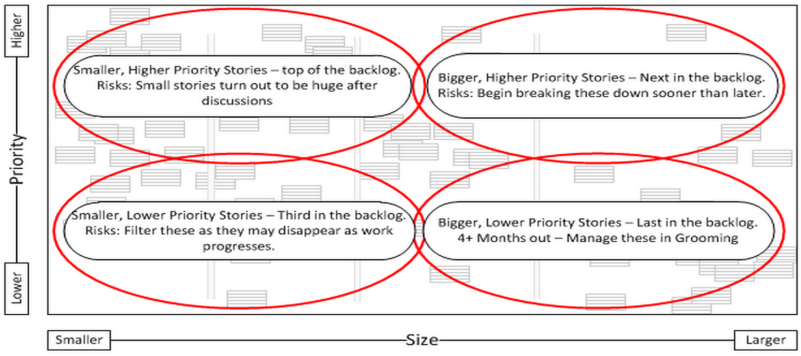


Figure 6 – Backlog wall divided into four quadrants using wall estimation technique (Lacey, 2012).

possible reason for these being the most used methods among practitioners, they claimed that there is no evidence that formal estimation models lead to better estimates.

In 2004, a survey on software estimation in Norwegian industry was carried out by Moløkken-Østvold, Jørgensen, Tanilkan, Gallis, Lien and Hove (2004). Their findings were similar to the previous mentioned study. Expert judgment-based estimations were the most popular, and the estimation accuracy when using formal methods did not show improvements over expert estimations. They also found that managers tend to think that the accuracy of their estimates is better than it in fact is.

Effort estimation accuracy in different development models has been analyzed (Moløkken-Østvold, Lien, Jørgensen, Tanilkan, Gallis and Hove, 2004; Moløkken-Østvold and Jørgensen, 2005). The results from those studies indicated that projects that are developed with an incremental model may be less likely to experience effort overruns than projects that implement sequential development models.

Later, in 2008, Yang et al. (2008) conducted a survey on software cost estimation in the Chinese software industry. The goal of the research was to investigate the state of software cost estimation in China, identify areas of potential improvement, and try to provide suggestions on how to improve the software cost estimation process and methods. Their study showed that large projects were more likely to experience cost and schedule overruns. When asked about satisfaction with the estimation techniques, most of the participants were neither satisfied nor dissatisfied. This study also showed that few organizations, a mere 15%, were using algorithmic model-based methods. The main causes for the low use of the previous techniques appeared to be the high adoption cost and the non-significant benefit once these are applied.

Trendowicz et al. (2011) performed surveys into the software industry with the purpose of analyzing current industrial practices related to estimating software development effort. Multi-

ple expert based techniques seemed to be the most accepted practice in the software industry.

2.4.2 Customer's role

Collaboration with the customer generates better results in software projects (Chow and Cao, 2008), both in agile and traditional development environments (Bakalova and Daneva, 2011).

Furulund (2007) analyzes and discusses the effect customer collaboration has on estimation accuracy. It shows that the projects with daily contact between contractors and customers have lesser effort overruns than other projects do. Moløkken-Østvold and Furulund (2007) point out that the current research into the field is insufficient, and emphasize the need of an extended set of project data for future analysis.

3 Research Methodology

Before choosing which research method, or methods, to apply to ones research project, it is essential that we have established our research questions. Only once the questions are settled, are we able to decide which method is most suitable to use, and assess if this method meets the requirements and purpose of the project. In other words, our choice of method is based on how we are best able to answer our research questions.

We have to consider what kind of data we wish to produce, and if we wish to gather the answers in the form of numbers and statistics, which would call for a quantitative approach, or if we wish to get answers in the form of people's thoughts and opinions which, in its turn, would call for the use of qualitative methods.

This chapter introduces the chosen methods used in my thesis, while describing their purpose, implementation and the results they produce.

3.1 Quantitative and Qualitative Methods

When conducting a research study there are two main types of data to gather. On the one hand you have the type of data that gives us results in the form of numbers, statistics and so forth. This is referred to as quantitative data and requires the use of quantitative methods. On the other hand you have the type of data that gives us results in the form of words, opinions and thoughts of someone that is an expert in the given research area. This is a form of qualitative data which requires a qualitative approach.

Often, these two different type of methods are being used separately depending on the type of data we wish to produce, which again depends on our research questions. There is, however, possible to combine the two methods in order to gather even more detailed data if desired.

3.1.1 Quantitative method

Quantitative data means data, or evidence, based on numbers (Oates, 2005). It can be generated from many types of research, but is mainly used in experiments and surveys (Section 3.2). Quantitative research comes in many forms, but most frequently used is measurements in numbers, statistics, quantities and others measurable units. We often wish to be able to generalize our findings to a large amount of people, and to look for patterns in the data and draw conclusions based on these patterns. To achieve this, the amount of respondents must be of a certain size, and they must also be in position to give relevant information to the field of research. In other words, we wish to reach a large group of relevant people and gather numerical data using,

for example, a survey.

There has been established a wide range of techniques for analyzing quantitative data (Oates, 2005). Among these are the use of graphs, tables or charts. The analyzing techniques allow you to see patterns in the answers, and from there draw conclusions that further can be generalized to a bigger population.

3.1.2 Qualitative method

As opposed to quantitative data, qualitative methods have a non-numeric approach. The quantitative methods aims to gather data in the form of words, images, sounds and so on (Oates, 2005). It is designed to gather as much in-depth information as possible from a few selected experts in the are of research. We deliberately seek out a small number of people who are able to give us valuable information in form of their thoughts, opinions, experience, feelings, and so on. These types of data are usually gathered in the form of interviews or free form surveys. They have a more open form which allows the participants to elaborate. Because these methods focuses on a limited amount of people, the results of qualitative research does not give a valid background for generalization.

Interviews are a common qualitative method, and comes in three different varieties: Structured, which consists of preplanned fixed questions, semi-structured, which has fixed questions with possibilities of further elaborations and follow-up questions, and finally unstructured interviews, which only has an area they want the participant to talk about, without leading them by asking preplanned questions, but rather executing the interview almost like a normal, free conversation about a given topic.

When it comes to analyzing qualitative results, one might find oneself with a very large amount of data. This cannot be processed by using standardized computer programs, but needs to be handled by the researcher in the way he sees most fit. Often it consists of reading through the data and extract the most valuable and relevant information regarding the research questions. To be able to do so, it is vital that the researcher is well prepared before starting the, say, interviews, and that he has a good and organized structure before conducting any data gathering.

3.2 Survey Research

The purpose of a survey is to obtain similar data from a large group of people, and try to find patterns that you can generalize to a larger population (Oates, 2005).

3.2.1 Survey design

Designing a good survey has always been a challenge. Therefore, many researchers have worked in this field trying to improve the survey design process (Singer and Presser, 1989; Fowler Jr, 1993, 1995; Oates, 2005). I followed their guidance and advice in order to write good questions.

As Oates (2005) explains, there are two types of data to collect from surveys. The first type is data that are directly related to the research questions. The second type of data are indirectly related to the research topics, and focus more on information about the respondents. These include questions about age, gender, years of experience, demographic data and so on. Data like these are important to consider when designing the questionnaire, proving helpful in organizing, filtering and finding patterns in the data at the time of analyzing the results.

The questions can be closed-ended or open-ended. A closed-ended question is a question format that limits respondents with a list of answer choices from which they must choose. Open questions, on the other hand, require a more in-depth and elaborated response, in which the participants are able to add information that they themselves find relevant. Open-ended questions are often shown as a statement which requires an answer. These are some examples of open-ended question:

- "Tell me about your experience with your new technologies."
- "Why did you choose this answer over the others?"

3.2.2 Pilot study

A pilot study is a small scale trial study conducted prior to a major study (Simon and Goes, 2011). It is usually used to pre-test or try out a research instrument. A reasonable number of participants to consider for a pilot study, is a sample size of 10–20% of the sample size for the actual study. Executing a pilot study is said to clearly increase the likelihood of success in the main study.

According to Simon and Goes (2011), the factors that can be resolved prior to the main study are as follows:

- Check that instructions are easy to understand.
- Check that researchers are sufficiently skilled for carrying the main study.
- Check the wording of a survey.
- Check the validity and reliability of the results.

- Check the statistical and analytical processes in order to make sure these are going to be effective.

It is important to make sure that the right questions are asked, so that I, in turn, am able to answer my research question.

3.2.3 Data processing and analysis

When analyzing and processing quantitative data, we wish to look for patterns and thereby be able to draw conclusions based on the results we find (Oates, 2005, p.245). There is a wide range of established techniques for doing so, and among these we find the use of tables, charts and graphs.

When it comes to analyzing and processing qualitative data on the other hand, we are dealing with peoples words, thoughts and opinions. Therefore we cannot draw conclusions, but rather get more insight and information from people who are experts in the field of question.

3.2.4 Reliability and validity

A study is reliable if any other researcher is able to perform exactly the same experiment, under the same circumstances, and generate the same results. Reliability focuses on consistency of a measurement (Shuttleworth, 2008).

On the other hand, validity refers to how well a study measures what it is meant to measure.

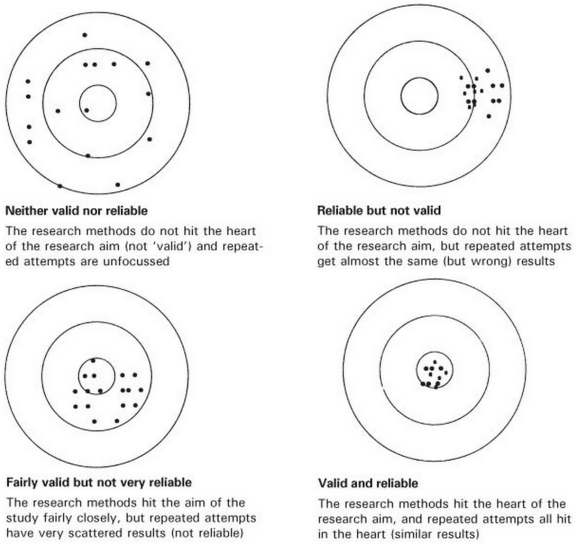


Figure 7 – Relation between reliability and validity (Precision Consulting, n.d.).

The relation between reliability and validity can be explained with the metaphor of the target presented in Figure 7. A study is neither valid nor reliable when the research methods do not hit the core of the research goal and, additionally, when the study is performed many times but yet produces inconsistent results. We say that a study is fairly valid, though not reliable, when the results gather in the middle of the research aim, but the repeated attempts of the research are spread across the target. A reliable, but not valid research, shows similar results when repeated, but these results do not aim at the center of the research target. A study is valid and reliable when the research methods hit the core of the research goal, and repeated attempts all show similar results.

A study can be reliable without being valid, and vice-versa. However, the ideal study should be both reliable and valid in order to be accepted and adopted by other researchers.

4 Research Design

We wish to avoid bias, misunderstandings, and most of all we want to get genuine answers to our research questions. The design depends on how and where we wish to execute the research, as well as the type of respondents we wish to reach. The goal is to make it as fitting, comfortable and convenient for the respondents as possible, and thus making it easier for them to participate and to answer truthfully. This will in return give the best data for further analysis.

4.1 The Survey

Empirical data will be collected through a web-based survey that includes carefully selected topics relevant for this thesis. The survey is intended to be answered by people involved in software projects in Norway and Spain.

The questions asked in the survey will help me to gather information about current software development methods, estimation techniques, the role of the participants, and the influence the customer has in the software estimation process both in Norway and Spain.

In terms of approach, the survey employs both qualitative and quantitative questions. This combined approach is used so as to overcome limitations of using either of the two. The quantitative approach focuses on obtaining numerical findings, while the qualitative approach focuses on personal opinions, observations, description and individual insights of the respondents.

The process of designing the self-made survey for this thesis, according to the research questions, is described below.

4.1.1 Intentions

To help me formulate the final questions, I define my question objectives. I also classify the questions by categories to make it easy to follow for the respondents. These categories are as follows: "Personal information and background", "Basic project information", "About the customer", "Software development methods", "Estimation methods", "The role of the participants" and "Closing".

When designing the survey I try to reach the whole range of roles within the participants. To do so, many of the questions are made to be answered by all of them, such as general questions about their education, work place, et cetera. However, I also need in-depth information from specific job positions, for example from developers, which I try to accomplish by creating different questions specified for certain working positions. Using LimeSurvey (Section 4.1.3) I will also be able to filter the questions to the point where if your response to a question implies

you are a developer, the following questions will be regarding your work in that exact position. This would make the survey more personalized in addition to reaching, and being relevant for, the whole spectrum of participants.

The first two categories, "Personal information and background" and "Basic project information", as well as "Software development methods", are designed for collecting basic information about the respondents. This is done in order to classify and filter data at a later point in the analysis. It also works as an introduction to the survey, and the quick and easy questions are good for getting the respondents started.

The core of the survey, and where I aim to get the most relevant information for the thesis, is in "Estimation methods". This category includes questions directly related to the research questions RQ1, RQ2, RQ3. It includes some direct questions about which methods the respondent used, as well as some open where they write information of their own. This combination is designed to get as much relevant information as possible.

Another important topic in this thesis is the role of the participants in the software estimation process. Therefore, all the questions related to the role of the participants are put in the associated category. Here there are questions designed directly for them, and about issues that they are best fit to answer and give information about.

I also separate the information about the customer into its own category. The reason for this is: First I wish to separate for organizational purposes. It would be useful to be able to look at the customers separately at a later point. Second, I wish to clarify to the respondents which topics are going to be included in the questionnaire. Naturally, their source of information would be of a different kind than that of, say, the developers, and thus I need them to answer questions that will be specific for customers.

4.1.2 Topics for analysis

The main topics that I analyze in the survey are as follows.

4.1.2.1 Software development methods

A brief classification of the software development methods that are being used today will give us a good overview and starting point for the thesis. It will also help us at the time of analyzing the results, giving us the chance to order or separate these results by methodologies. For example, differentiating between traditional and Agile methods could give us clearer results if needed.

4.1.2.2 Estimation methods

The literature shows the existence of several estimation methods. However, these methods have to be put into practice. Different companies might have their own ways of estimating. The possible options are using a single method, or combining different techniques. I intend to gather information about which estimation methods are being used in Norway and Spain, and how these are being used.

4.1.2.3 Participants' role

Related to the estimation methods, we have the participants of the estimation process. These are the ones who make the estimates by applying the chosen methods. In this study I also examine who participates in the estimation process, and to which extent they can influence the estimation process. I also wish to further analyze the customer's role in the estimation process. Specific questions regarding the customer are sent to all the participants. Customer's IT knowledge, communication frequency with the development team, and contract types are some of the examples.

4.1.3 Details about survey design

To get as full and informational answers as possible from the survey, I use a combination of question types. Some questions have different options where one or more have to be answered, whereas others are optional to answer or have an open field where the participant is able to elaborate. By using this combination of question types, I will be able to get detailed answers for my research questions.

4.1.4 Designing the survey

The survey can be found in Appendix A. In addition to the version written in English, the survey was also translated to Norwegian and Spanish in order to reach more respondents in both countries. The respondents chose their preferred language before starting the survey, but they were also able to switch languages at any point along the way if they wished to do so. This section explains how I created the survey, and gives detailed information about the chosen questions.

Personal information and background

The main purpose of this section of the survey is to create demographic variables. These variables might affect other variables, and therefore I found them necessary when designing this survey. They will be helpful later in the analysis for filtering and comparing results.

- *"In which country do you work?"*
  Demographic data. Necessary for separating answers from Norway and Spain. It is a single choice question.

- *"What is your age?"*
  Demographic data. I want to see if there is any difference between respondents of different ages regarding estimation. Single choice from given age ranges.

- *"Gender"*
  Demographic data. Single choice.

- *"Do you have a degree in higher education? If yes, specify the degree."*
  Demographic data. I also want to see if there is any connection between the level of education and the knowledge the respondents have about estimation practices. Single choice.

- *"How many years experience do you have in the IT field?"*
  Demographic data. This will be used in order to see if there are any differences between more and less experienced people. Single choice.

- *"Which is your current job title?"*
  Demographic data. The job title of the participants is essential at the time of analyzing the results. Some options will be presented, as well as an "Other"-option with a text box for specifying. Single choice.

Project information

This section intends to gather information about the most recent or current software project the respondents have worked on.

- *"Is the project you have in mind when answering this survey a: current project or an already finished project"*
  The main purpose of this question is to make sure that the respondents know they will be answering questions about a specific self-chosen project. Single choice.

- *"How long was the duration of the project? In case we are talking about the current project, how long do you think this is going to take?"*
  Project related variable. The duration of the project will be used in order to see if this is a crucial factor when selecting some estimation methods over others. Single choice.

- *"Was the project internal or external?"*
  Project related variable. A software project can either be external or internal depending on the stakeholders. External projects are the projects that are undertaken for an external client. Internal projects, on the other hand, do not have external clients, and are projects done to enhance the company's own business. Internal and external projects might have differences at the time of making estimates. Single choice.

- *"Was there any client or product owner involved in the project?"*
  Project related variable. Single choice.

- *"How many teams were involved in the software project?"*
  Project related variable. Single choice.

- *"Specify how many of the following were involved in the project. Developers, Managers Customer/Product owner, Other"*
  Project related variable. This will be used for counting the number of people with different roles working on the projects. I will also use this for counting the total amount of people working on the projects. The project size might play a significant role on how the estimates are done in software projects.

- *"Was there made any software estimations before or during the project?"*
  Project related variable. I use this for filtering survey questions related to estimation. If the respondents choose the "No, there was not made any estimation before or during the project"-option, they will not be asked questions related to estimation. Single choice.

- *"Which contract type was set up for the project?"*
  Regarding the type of contract, I distinguish among the following: Per hour (time and material), fixed price, target price (risk sharing between contractor and customer), and other. I also have an "I do not know"-option. It is a single choice question.

  A target price contract is a contract were the contractor and the customer share the risk of overruns (Furulund, 2007). The contract defines a target of effort, usually in time, and also a date for completing the project. If the project is completed before expected, the contractor and customer share the benefits. If more effort than what is defined in the contract is

needed, both the contractor and the customer share the overrun costs. The customer pays a percentage of the overrun costs, often 50%, while the contractor is responsible for the rest.

- *"Which company do you work for? In case you are a customer, write the name of the company that worked for you."*
  Different companies might use different estimation techniques. However, if there for example only are respondents from the same company, the results cannot be generalized to a higher population. I will use this question variable in order to make sure the previous problem does not occur.

About the customer

The purpose of this section is to analyze the customer, and to gain some variables for further analysis. This section is only shown to the respondents who answered "Yes" at question "Was there any client or product owner involved in the project?".

- *"Rate the customer on the following properties. Collaboration skills, IT competence, Decision making ability, Clarity of project goals"*
  Customer related variables. Ranged from "Very poor" to "Very good".

- *"If any, what has been the communication frequency with the customer/product owner?"*
  Customer and project related variable. I want to see if projects with high communication frequency have different results than the ones with low communication frequency.

Software development methods

The selected software development method affects the estimation process in software projects. This section aims to generate data related to software development methods.

- *"Which of the following software development methods do you have experience with? Waterfall model, Spiral model, Agile development methods, Other."*
  Development method related variable. Multiple choice question.

- *"Which of the following agile methods do you have experience with? Extreme Programming (XP), Kanban, Scrum, Other."*
  Development method related variable. Multiple choice. It will only be displayed when the participant states to have experience with Agile methods.

- *"Which software development method have you used in the most recent project? Waterfall model, Spiral model, Agile development methods, Other."*
  Development method related variable. Multiple choice. This question will help me to see the differences between various development methods regarding the estimation process.

- *"Which of the following agile method have you used in the most recent project? Extreme Programming (XP), Kanban, Scrum, Other."*
  Development method related variable. Multiple choice. It will be asked only to the participants who have used Agile methods in the selected project.

Estimation

This section of the survey includes questions directly related to the estimation process. The questions will be hidden from the respondents that claim to work on a project where there are not made any estimates before or during the project.

- *"Which of the following estimation techniques have you used in the most recent project? If you have used a combination of methods, please select all that apply."*
  This is a multiple choice question, with the following options.

  - Expert judgment (consider as experts all individuals with competence in estimating software development effort, including software developers and students with previous experience)
  - Analogy (comparing older projects)
  - Algorithmic models (COCOMO,...)
  - Neural networks
  - Bayesian analysis
  - Function Point analysis
  - Story Point estimation
  - Other

  I will be able to see which estimation methods or techniques are currently being used in the software business.

- *"Why did your team choose that/those estimation technique(s) over others?"*
  It is an open question, that will help me explain why some practitioners prefer or choose some estimation methods over others.

- *"Which of the following units of estimation did you use when estimating the most recent project?"*
  Multiple choice. I will be able to see in which estimation units companies are estimating their projects, and if they use a combination of these.

- *"Which of the following techniques did you use for helping the estimation across teams?"*
  This question is only asked to those working on multiple-team projects. This is interesting for the understanding of how companies estimate the total effort of a project, and therefore time and cost, when this is divided into smaller teams. The following three options are presented as multiple choice answers.

  - Option A: "First carrying out a joint planning poker session with members from all the teams in order to establish a common baseline for story points"
  - Option B: "Identifying stories in the backlog that are representative of each story point size and making those stories known to all of the teams"
  - Option C: "Normalization of Story Points (SAFe's approach,...)"

- *"What level of experience do/does the team(s) have in estimating user stories?"*
  This will only be asked if the respondent claims the use of agile methods in his project. I wish to investigate if my assumptions are correct regarding the following hypothesis. A high level of experience when it comes to estimating user stories, might lead to more accurate results and therefore to higher satisfaction levels. I will also use this variable, namely the level of experience, for filtering results and finding correlations with other variables, if any are found.

- *"Which of the following agile estimation techniques did you use during the most recent project?"*
  I want to know which techniques software companies are using when estimating agile projects. It is a multiple choice question, designed only for respondents working on agile projects.

- *"Which estimation scale did you use during the most recent project for estimating user stories? Specify the story point sizes."*
  This question is only asked to people who use story points as estimation unit. I want to see which scales are they using, and if there is any correlation between these and the level of experience of the team estimating user stories. I assume more experienced teams will not use T-Shirt sizes, since these units need to be converted to story points later.

- *"How successful were your estimations in meeting their goal?"*
  I want to know the level of success of the estimates in different projects. The options range from "Very unseccessful" to "Very successful".

- *"Overall, how satisfied are you with the estimation methods you have used in the most recent project?"*
  I wish to know something about the satisfaction level of the participants regarding the estimation methods used in their project. The options range from "Very dissatisfied" to "Very satisfied".

- *"If not satisfied, what would you change? Do you have any suggestions?"*
  I will use this open question, where the participants will be able to express their thoughts and suggestions, to determine what the they would change in the estimation process.

The role of the participants

In this section, participants will be asked about the role and the influence of different parties involved in the estimation process. Therefore, it will only be shown to respondents who work in projects where estimates are made.

- *"Who took part in the estimation process?"*
  I want to see who takes part in the estimation process, and verify that developers are part of this process in agile projects, as opposed to managers which should not be involved.

- *"Were the developers bound to follow estimations made by others in a higher position?"*
  I will also analyze if the developers are bound to estimates made by others in higher positions. It is a "Yes" or "No" question.

- *"To which extent does the developers'/managers'/customers' decision influence the final estimations?"*
  I also wish to investigate to which extent the decision of people in different positions influence the final estimates. This is separated into three questions, one regarding each of the following positions: Developer, manager, and customer. It will be asked to all of the respondents in order to have a better insight of the real influence. Possible answers are ranged from "Very low" to "Very high".

Closing

- *"If there is any other information about the project you would like to add, please write it here"*
  This is an open question where the respondents are able to add additional comments related to the project.

- *"If you would like to give further information that could be relevant to the study, I would appreciate your feedback"*
  I want to get feedback from the participants, and see if they can give me helpful advice for my research.

- *"You can also upload files with extra information."*
  I also want the participants to have an option to upload documents.

- *"If you are willing to do an interview please write your email address."*
  This question is made for being able to reach participants after the survey is done. In the case that I find some answers especially interesting I want to be able to contact them for further elaboration.

4.1.5 Summary

The survey is structured in such a manner that the participants are fully able to understand the goal of my research. It is divided into different topics, and some questions are directed towards specific groups or participants.

I use closed-ended questions in most of the cases where I want to be able to compare the answers. On the other hand, where I need a better insight from the participants' thoughts or ideas, I apply open-ended questions.

The survey is designed with the purpose of getting as much information as possible, so that I am able to answer my research questions.

5 Data Collection

To a large extent, this thesis is a comparative study. The main instrument used for gathering the data was thus a web-based extent survey.

5.1 Web-based Survey

I chose to use a web-based survey for the following reasons.

A web-based survey makes it possible to reach a large number of participants in a quick and non-expensive way. Since the participants in this case were people in the IT business, they were easily accessible because of their everyday use of computers, hence the use of the online questionnaire.

This kind of survey also minimizes the effort of gathering and sorting the data, as well as being easy for the users to complete.

5.1.1 Participants

The respondents were people working in the IT business in either Norway or Spain.

Collecting data from only one of the roles involved in the project, such as the project manager, could generate incomplete or biased information (Jørgensen and Moløkken-Ostvold, 2004). Taking this into account, I tried to get access to people with different roles within the projects, including project managers, developers and customers.

In order to gather realistic data, it is important to have as many participants as possible in both countries. I made use of the power of the social networks such as LinkedIN and Facebook to expand and share the questionnaire. I also sent it by email to University teachers, IT companies and friends with connection to the IT field, with the purpose that they would forward it to their contacts. On the last page of the survey the participants were asked to share the survey with their colleagues. Furthermore, an entry was published in the newsletter of the company in which I was working, with a description of the study and a link to the survey.

5.1.2 Survey tool

I found LimeSurvey (Schmitz, 2012) to be the most suitable tool for creating the questionnaire and collecting the results for this study. LimeSurvey is a free and open source online survey application, distributed under the GNU General Public License. It allows users to develop and publish online surveys, as well as collect responses, without doing any programming.

Once the survey was created, it was made publicly accessible during a 35 days period between the 26th of November and the 30th of December 2013.

The responses were automatically saved into a database, and later exported to the necessary format for analysis once the survey was closed. LimeSurvey offers a good solution for extracting the results in different formats including R, CSV, XLS, DOC and SPSS.

5.1.3 Processing and analyzing data

Based on recommendations from colleagues, in addition to my own research for finding the optimal analyzing tool, I decided to work with R, a free software programming language and software environment for statistical computing and graphics. R offers well documented, wide ranged options for statistics, including filtering, grouping, sorting and plotting data. It also has an active online community, which was essential for my choosing of this tool.

I chose to display the results in bar charts, pie charts, and tables in order to make the results easy to read and understand. Bar charts were used for displaying frequencies or trends. Although most of the bar charts consist of a single series, there were also cases where multiple series were plotted together. Pie charts, on the other hand, helped me when showing fractions or percentages of the total (Oates, 2005, p.249). Specific colors were chosen for representing different variables, which were applied throughout the thesis for ensuring consistency. For example, the orange color represented the results from Norway, whereas the blue was used for identifying the data from Spain. I used tables in order to show more details about specific variables such as age, gender, degree, or job position.

I created a script for compiling all the charts and pies with a single click, which saved me a lot of time.

5.1.4 Conducting the pilot study

Before publishing the definitive survey, I carried out a pilot study where I tested the survey. Five fellow students answered the questionnaire through LimeSurvey, while I was close to them following the process. This way I was able to locate questions that could be misunderstood, or any other problems the respondent might have while answering the questionnaire.

With the feedback I got from the testers, I reformatted some questions to avoid misinterpretations.

I also exported the results of the pilot study to a CSV file, and went through it to check if I was missing some vital points that I would need for this thesis. I did find a few weak points in the data, and therefore decided to add more questions in order to make the survey more complete.

5.1.5 Ethical considerations

As always when conducting a research study, ethical considerations have to be taken into account. It is important to be aware of all ethical aspects before, as well as during and after, the study is executed.

In the case of this thesis, which was a web-based comparative study, there were several things to consider. The first and foremost was anonymity of the participants. No details about their names, workplace, or any other personal information was neither asked for nor saved. However, at the end of the survey, participants had the option to write their email address in case they were willing to participate in a follow-up interview.

At the very beginning of the survey there was made a page with full information about the study. Here it was explained what the study was going to be used for, who had access to the data and for how long, and that it was anonymous to participate. In addition, it was written that you were free to quit the survey, and not deliver your answers, at any time. It thoroughly explained the purpose of the study, and when and how the given information would be used. My contact information was also written if there were any questions about the study.

6 Analysis

Analysis of data is the process of checking out, ordering, filtering, transforming, and modeling data with the purpose of discovering useful information, and suggesting conclusions.

This chapter shows the analysis of the survey. For each section of the survey, the results from Norway are presented first, followed by the ones from Spain. At the end of each section, a comparison between the two countries is presented.

The total number of respondents who finished the survey was 77, whereof 44 worked in Norway and 33 worked in Spain. My goal to reach as many different job positions as possible was accomplished, having an expected distribution of respondents from all the groups. However, there was an exception, that of the customers. I was only able to get two respondents in the customers position, making this data non-significant (Table 1).

	Total	**Total %**	**Norway**	**Norway %**	**Spain**	**Spain %**
All	77	100%	44	100%	33	100%
Software Developer	48	62.3%	26	59.1%	22	66.7%
Software Architect	7	9.1%	5	11.4%	2	6.1%
Team Manager	9	11.7%	5	11.4%	4	12.1%
Customer	2	2.6%	1	2.3%	1	3%
Other	11	14.3%	7	15.9%	4	12.1%

Table 1 – Participants' position

6.1 Personal Information and Background

6.1.1 Norway

The respondents were not asked for their specific age, but rather to choose from a given spectrum. Almost half of the participants were between the ages of 26 and 35. Following this age group were participants between 36 and 45 years of age, which were a quarter of the sample size. 18.2% were under 26 years old, whereas the 13.6% were between 46 and 55 years old. 90.9% of the respondents were male. See Table 2.

Almost all of the respondents had higher education. Of these, around one third had a Bachelor Degree, close to two thirds a Master Degree and only a minority had a PhD.

When the participants were asked for their years of experience in the IT field, 34.9% answered that they had been working within IT between four and 10 years. 30.2% had zero to three years of experience, whereas more experienced participants, with 11 to 20 years in the

	Male					Female					***Total***				
	<26	*26-35*	*36-45*	*46-55*	***Total***	*<26*	*26-35*	*36-45*	*46-55*	***Total***	*<26*	*26-35*	*36-45*	*46-55*	***Total***
Bachelor	4	5	2	2	13	0	0	0	0	0	4	5	2	2	**13**
Master	3	11	6	2	22	0	1	1	1	3	3	12	7	3	**25**
PhD	0	1	1	1	3	0	0	0	0	0	0	1	1	1	**3**
Other	1	1	0	0	2	0	0	1	0	1	1	1	1	0	**3**
Total	8	18	9	5	**40**	0	1	2	1	**4**	**8**	**19**	**11**	**6**	**44**

Table 2 – Participants' background information (Norway)

field, accounted for 25.6%. The participants with more than 20 years of experience represented 9.3% as shown in Figure 8.

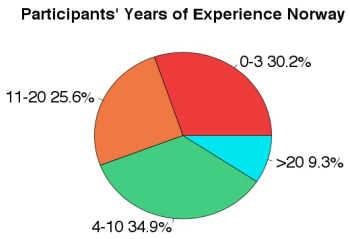


Figure 8 – Participants' Years of Experience (Norway).

59.1% were Software Developers, including all programmers, testers, database engineers, analysts, and user interaction designers. 11.4% were Software Architects, followed by the Team or Project Managers, which had the same percentage as the one previously mentioned. Only one of the respondents was a customer. 15.9% chose the option "Other" under the job positions question, of which included business architects, students, and information managers. See Table 2.

6.1.2 Spain

One third of the respondents were between the ages of 26 and 35. Following this age group were participants between 36 and 45 years of age, which represented a total of 30.3%. Participants under 26 years old were also 10 out of 33, whereas only two were between 46 and 55 years old. 87.9% of the respondents were male. See Table 3.

Almost all of the respondents had higher education. Around one quarter had a Bachelor Degree, almost two thirds had a Master Degree and only two a PhD (Table 3).

When the participants were asked for their years of experience in the IT field, 27.3% answered that they had been working within IT between four and 10 years. 42.4% had zero to three years of experience, whereas more experienced participants, with 11 to 20 years in the

	Male					Female					***Total***				
	<26	*26-35*	*36-45*	*46-55*	***Total***	*<26*	*26-35*	*36-45*	*46-55*	***Total***	*<26*	*26-35*	*36-45*	*46-55*	***Total***
Bachelor	4	4	0	0	8	0	0	0	0	0	4	4	0	0	**8**
Master	4	6	6	1	17	1	1	1	0	3	5	7	7	1	**20**
PhD	0	0	1	0	1	0	0	0	1	1	0	0	1	1	**2**
Other	1	0	2	0	3	0	0	0	0	0	1	0	2	0	**3**
Total	9	10	9	1	**29**	1	1	1	1	**4**	**10**	**11**	**10**	**2**	**33**

Table 3 – Participants' background information (Spain)

field, accounted for close to one quarter. The participants with more than 20 years of experience represented 6% as shown in Figure 9.

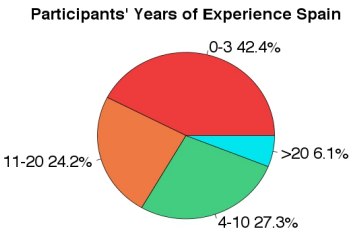


Figure 9 – Participants' Years of Experience (Spain).

They were also asked about their job position, or job title. 66.7% were developers. The second most answered option with 12.1% was Team Manager. Two of the participants were Software Architects, whereas only one of the respondents was a customer. Four participants chose the option "Other" under the job positions question, of which included a security analyst, a researcher, a university teacher, and a hardware developer. See Table 1.

6.1.3 Comparison

In both countries, most of the participants were between the ages of 26 and 45. However, the amount of people under age 26 was higher in Spain, whereas the amount of people older than 46 was higher in Norway (Tables 2 and 3).

Regarding the gender, there was a similar distribution between the countries. For every 10 participants, only one was female. See Tables 2 and 3.

When looking at the participants' educational degree (Tables 2 and 3), we see a similar distribution in Norway and Spain.

As seen in Figures 8 and 9, more than half of the respondents had four to 20 years of experience within IT in both Norway and Spain. However, respondents with less than three

years of experience represented a higher percentage in Spain (42.4%) comparing to Norway (30.2%).

The job position of the participants was distributed in an almost identical manner in both countries. See Table 1.

6.2 Project Information

The participants were asked to choose a project that they were currently working on, or a project that they had recently finished, when answering the next questions. In this manner, the answers were going to be related to specific projects and not to all projects that the participants were, or had been, working on.

The following section shows information about these projects.

6.2.1 Norway

Based on the participant's answers, 59.5% were working on internal projects, whereas 40.5% were involved in external projects (Figure 10).

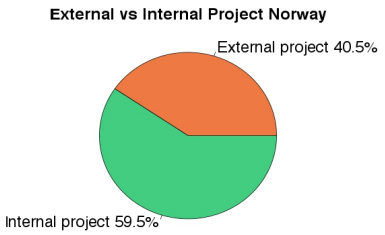


Figure 10 – Project External vs Internal (Norway).

The size of a project can be defined by factors such as duration, or the amount of people working on the project, among other elements. As we see in Figure 11, there is an even distribution in project duration (1-2 years with 25%, 7-12 months with 25%, and 4-6 months with 20.5%), excluding projects with a duration shorter than four months, and longer than two years, which had lower percentages.

When defining the project size by the number of people, we had the following classification. 40.9% were medium sized projects composed by six to 10 people. 31.8% of the projects were large (11 to 50 members). Projects with less than six people (small projects) were the 22.7%. Only two projects were classified as extra large (projects with more than 50 workers) in terms of number of people. See Figure 12a.

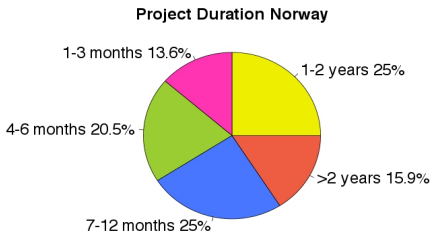


Figure 11 – Project Duration (Norway).

(a) People per project.

(b) Teams per project.

Figure 12 – Project Size (Norway)

Most of the projects, 60.5% to be precise, were composed by a single team. Bigger projects, however, were divided into smaller teams. Projects with two or three teams represented 20.9% of the total, while the projects with more than three teams represented the 18.6%. See Figure 12b.

In most cases we do see correlation between the previous discussed size measures. Projects with more people were the ones with more teams. The average number of people for single team projects was 7.5. Projects with two to three teams had a mean of 37 people. However, one of these projects had 234 workers. If we exclude this project, the average is 12.3 people per project. When the project was divided into more than three teams, the average number of workers was 22.8.

As shown in Figure 13, 29.5% of the projects had a contract based on hours worked. 15.9% were target price projects, followed by fixed price contract projects and other contract type projects, which showed the same percentage. The amount of respondents that did not know the contract type was 10 (22.7%). Two of the respondents that answered the option "Other", confirmed that they have used fixed price for each iteration/sprint, but without any long term contract.

A customer or a product owner was involved in 81.8% of the projects, against the 18.2% were they were not. See Figure 14.

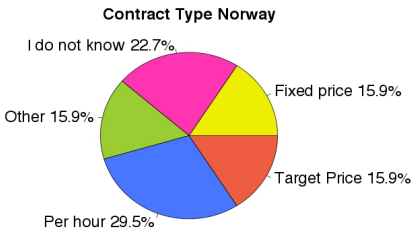


Figure 13 – Project Contract Type (Norway).

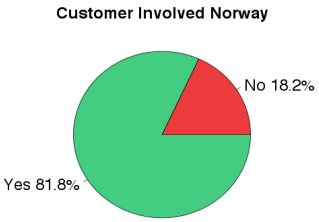


Figure 14 – Project Customer Involved (Norway).

A high rate of participants responded that they were using estimates in their software project. However, I wanted to differentiate these projects based on the time or phase of the development when the estimates are conducted.

68.2% of the total projects used estimation techniques both before and during the development process. 11.4% only estimated before the development process started, whereas the remaining 6.8% of the projects were only estimated during the development process and never up front (Figure 15). 13.6% did not use estimates.

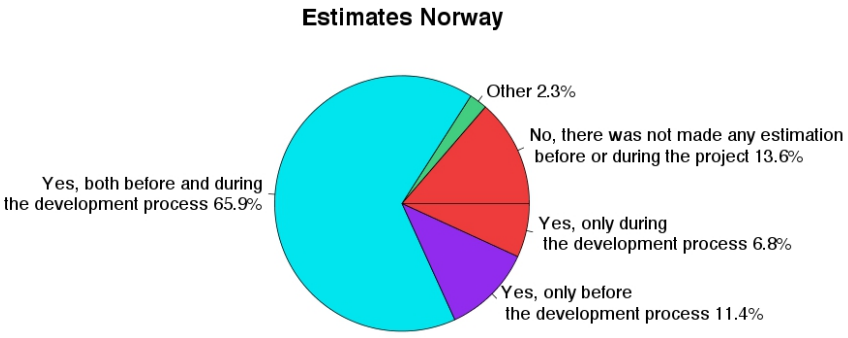


Figure 15 – Project Estimates (Norway).

6.2.2 Spain

54.5% of the projects were internal, whereas 45.5% were external (Figure 16).

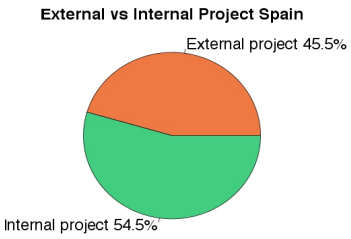


Figure 16 – Project External vs Internal (Spain).

As we see in Figure 17, many projects had a duration between seven and 12 months. 18.2% of the projects had a duration shorter than three months. Projects with a duration longer than two years made up 15.2% of the total, followed by projects that took between one to two years, which made up 12.1%. Only 9.1% of the projects had a duration of four to six months.

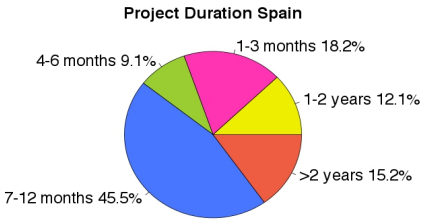


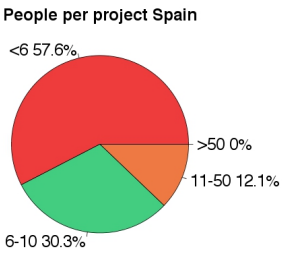
Figure 17 – Project Duration (Spain).

When analyzing the size of the project by the number of people involved, we have the following classification. Most of the projects were small projects with less than six people. Six to 10 people projects amounted for almost a third of the total, whereas projects with 11 to 50 workers only constituted 12.1%. There were no projects with more than 50 people involved. See Figure 18a.

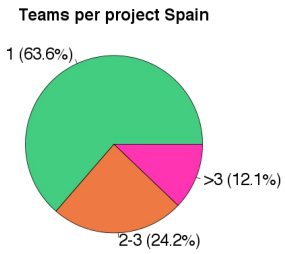
Most of the projects, 63.6% to be precise, were single team projects. 24.2% of the projects were divided into two or three teams, whereas the projects with more than three teams represented a 12.1%. See Figure 18b.

In Figure 18, we can see that the number of teams in a project, is directly related to the amount of people working on it.

As shown in Figure 19, 42.4% of the projects appeared to have a fixed price contract. Projects with a contract based on hours worked were 15.2%, whereas only one, giving a mere



(a) People per project.



(b) Teams per project.

Figure 18 – Project Size (Spain)

3%, had a target price deal. 12.1% answered the option "Other", and one of these respondents claimed the use of fixed price for each iteration/sprint, but without any long term contract. 27.3% did not know the contract type used in the project.

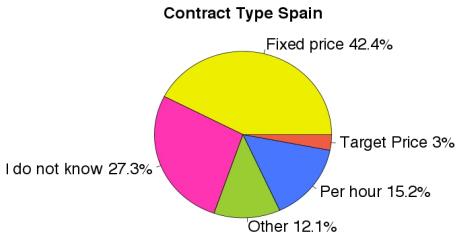


Figure 19 – Project Contract Type (Spain).

A customer, or a product owner, was involved in 66.7% of the projects, in contrast to the 33.3% where they were not. See Figure 20.

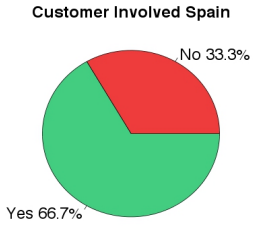


Figure 20 – Project Customer Involved (Spain).

In more than half of the projects, estimation techniques were used both before and during the development process. Close to one third were only estimated before the development process started, whereas 6.1% solely estimated during the development process and never up front. The remaining 6.1% of the projects were not estimated at all (Figure 21).

Figure 21 – Project Estimates (Spain).

6.2.3 Comparison

In both countries the respondents reported that they worked on external and internal projects to the same extent, with a slight tendency towards internal projects (Figures 10 and 16).

In Norway, more than half of the projects had a duration of more than one year, whereas in Spain these only amounted for 27.3% of the answers. Projects with a duration between seven and 12 months occurred quite often in Spain (45.5%). In fact, this was almost twice as much as in Norway, where these projects only accounted for 25%. Projects shorter than six months were around one third of the total in Norway, and around one quarter in Spain. See Figures 11 and 17.

When comparing the size of the projects by the amount of people working on them, Figures 12 and 18 show us that the projects in Norway consisted of more workers than the projects in Spain. Indeed, none of the projects in Spain had more than 50 people involved, whereas they in Norway amounted for 4.5% of the answers. Also, the number of projects with 11 to 50 people in Norway was almost three times higher than the projects in Spain. Therefore, small projects, with less than six people, in Spain were almost three times more frequent than in Norway.

The number of teams in which the projects were divided, was similar in both countries. Most of the projects were composed by a single team.

The contract type differed from project to project, and also from country to country. In Spain, fixed price contracts were the most popular, as opposed to Norway, where the most used contract type was based on hours. We can also see that target price contracts were popular in the last mentioned country. See Figures 13 and 19.

When including both of the countries, there was a customer or product owner involved in most of the projects (Figures 14 and 20). In Norway this was a fact in 81.8% of the cases, whereas in Spain the customer was involved 66.7% of the times.

Almost all of the projects, 93.9% in Spain and 86.4% in Norway, were estimated either before, during, or both before and during the development process. Most of these were estimated both before and during the development phase. In Norway this amounted for 68.2% of the cases, whereas in Spain it gave a result of 51.5%. The cases where the estimates only were made before the development process, were three times more frequent in Spain than in Norway. See Figures 15 and 21.

6.3 Development Methods

6.3.1 Norway

The participants were asked about their experience with different development methods.

On the one hand, 61.4% said to have previous experience with the Waterfall model. On the other hand, only 4.5% of respondents had had experience with the Spiral model. Most importantly though, is the high amount of people who had had experience with agile methods. They accounted for 95.5%, and a total of 42 out of 44 people. See Figure 22.

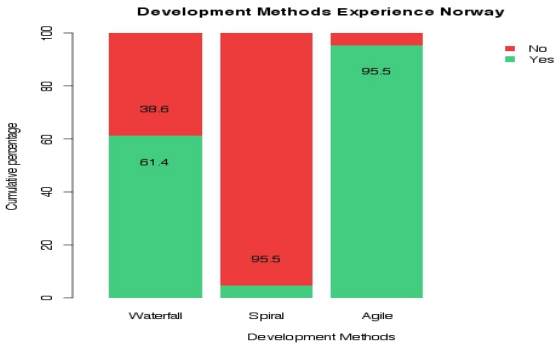


Figure 22 – Development Methods Experience (Norway).

As we can see in Figure 23, from the group of people who had had experience with agile methods, 97.6% had worked with Scrum. Furthermore, 47.6% had experience with Kanban, and 40.5% with eXtreme Programming (XP).

When the participants were asked for the development method used in their project of choice, 40 out of 44 stated that they had used agile methodologies during the development. Traditional methodologies such as Waterfall or Spiral models showed significantly lower rates of use, with a mere 9.1% and 2.3% of the projects respectively (Figure 24). It should be emphasized that these 9.1% who said they were using the Waterfall model, also stated to have used this method in combination with agile methodologies. Some of these respondents intro-

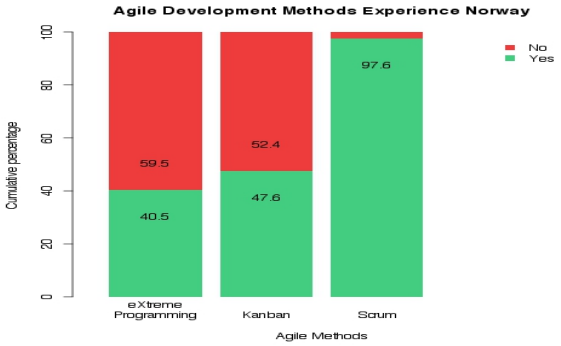


Figure 23 – Development Agile Methods Experience (Norway).

duced the terms "Scrummerfall" and "Water-scrum-fall", a combination of Waterfall and the well-known agile method Scrum.

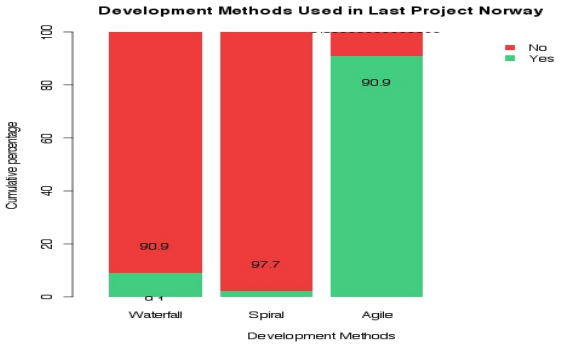


Figure 24 – Development Methods Used in Last Project (Norway).

From those who were using agile methods, 75% declared to be using Scrum, followed by Kanban, which was used in 27.5% of the projects. Only one of the participants stated to be using XP. Regarding to the combination of agile methods, five people out of 40 people (12.5%) said they had used Scrum and Kanban combined, whereas another respondent admitted to use a little from several methods, without specifying what these were. See Figure 25.

6.3.2 Spain

Most of the participants had had experience with agile methods, making up 54.5% of the total response. The same amount of respondents claimed to have experience with the traditional waterfall model. Only 18.2% of respondents had had experience with the spiral model. See Figure 26.

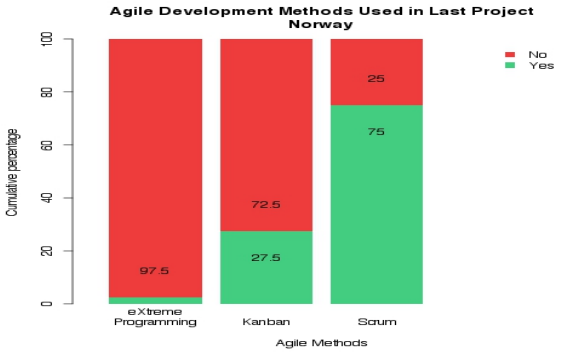


Figure 25 – Development Agile Methods Used in Last Project (Norway).

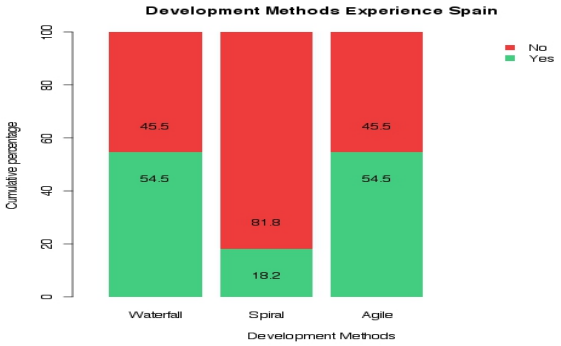


Figure 26 – Development Methods Experience (Spain).

As we can see in Figure 27, from those who had had experience with agile methods, 83.3% had worked with Scrum. Furthermore, 44.4% had experience with eXtreme Programming (XP), and 11.1% with Kanban.

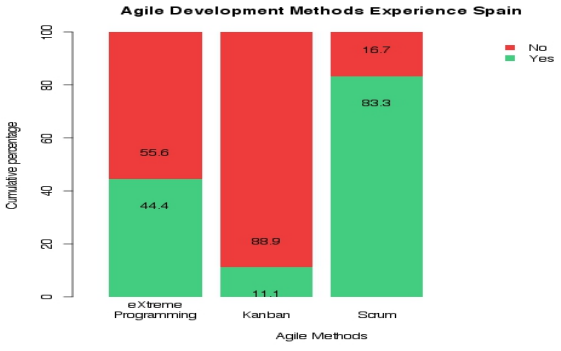


Figure 27 – Development Agile Methods Experience (Spain).

When the participants were asked for the development method used in their project (Fig-

ure 28), 36.4% stated that they had used the waterfall model during the development. The same amount of projects were developed using agile methods. The traditional spiral model was only used in 6% of the cases. One respondent claimed to be using a methodology called Métrica v.3. The remaining participants, accounting for 18.2%, did not select any of the options.

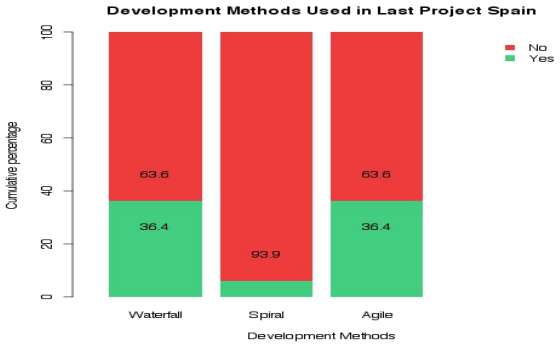


Figure 28 – Development Methods Used in Last Project (Spain).

From those who were using agile methods, 90.9% declared they were using Scrum, followed by XP, which was applied in 36.4% of the projects. Kanban was not used in any of the cases (Figure 29). From the previous, two of the respondents (18.2%) said they had used a combination of XP and Scrum.

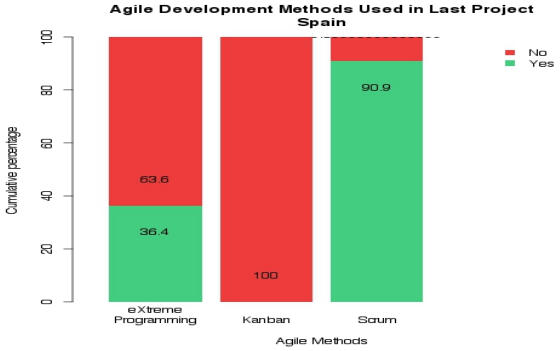


Figure 29 – Development Agile Methods Used in Last Project (Spain).

6.3.3 Comparison

When the participants were asked about their experience with different development methods, the main difference between the two countries was related to agile methods. Almost all of the participants in Norway had experience with agile methods, whereas in Spain these were only

54.5% (Figures 22 and 23). The experience with the spiral method, on the other hand, was four times higher in Spain.

As shown in Figures 26 and 27, Scrum appears to be the method most widely used among agile practitioners in both countries, with 97.6% in Norway and 83.3% in Spain. XP was experienced by around 40% in each country. However, we find the biggest difference with Kanban, which had a significantly lower response rate in Spain (11.1%) than in Norway (47.6%).

When comparing the development methods participants have used in their most recent project, we can see in Figures 24 and 25 that the use of agile methods is three times higher in Norway than in Spain. The traditional methods, on the other hand, were almost not used in Norway, whereas in Spain they are still a common practice, the waterfall method being used in more than one third of the projects.

Similar to experience, the most used agile method was Scrum both in Spain and Norway. Kanban was only used in Norway. 36.36% of the projects in Spain were developed using XP, whereas in Norway only 2.5% used this method. See Figures 28 and 29.

6.4 Estimation

The participants who confirmed the use of estimates in their project, which were 84.1% of the respondents in Norway (shown in Figure 15), and 93.9% in Spain (Figure 21), were further asked several specific questions regarding the estimation process.

6.4.1 Estimation methods

6.4.1.1 Norway

The respondents were shown a list of different estimation techniques, and asked to choose the ones they had used in their most recent project. In the case of participants using a combination of methods, they were asked to select all that applied.

72.9% of the projects were estimated using story points (see Figure 30), followed by Expert judgment based estimation, present in more than half of the projects. 11 out of 38 participants answered that they were using analogy as the estimation technique. Algorithmic models were only used in one project, which was the same percentage as Function Point analysis. Neural Networks and Bayesian Analysis were not selected by any of the respondents.

A combination of different techniques occurred in 19 of the projects, amounting for 51.5%. The estimates in eight of the projects (21.6%) were based on a combination of Expert Judgment and Story Point estimation. Five projects, making up for 13.5%, appeared to be using a combination of the previous two techniques, in addition to Analogy. Three projects, and thereby

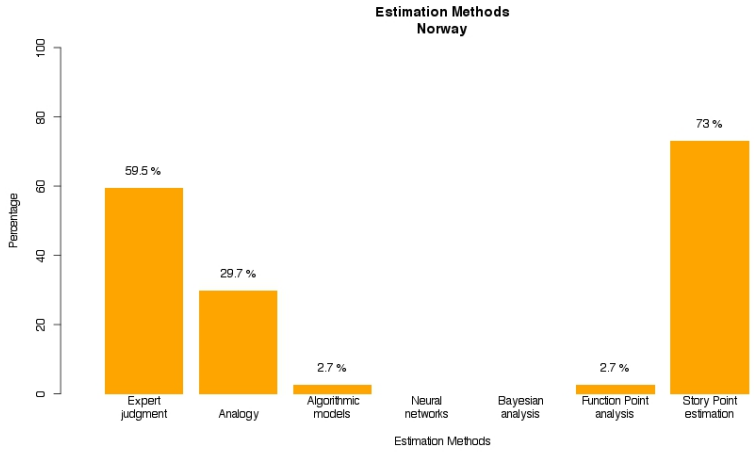


Figure 30 – Estimation Methods (Norway).

merely 8.1%, used Expert Judgment in combination with Analogy. The remaining technique combinations, which were applied in one project each, were as follows: Analogy with Function Point analysis, Expert Judgment with Story Points estimation in addition to Algorithmic models, and Analogy with Story Point estimation. One participant did not select any of the presented choices. See Figure 31.

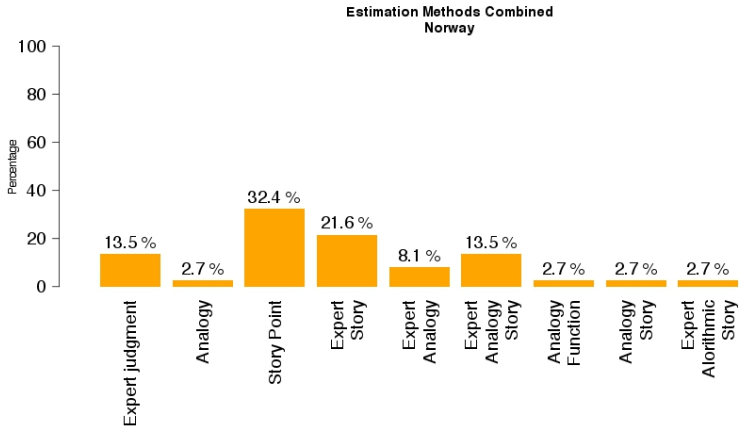


Figure 31 – Estimation Methods Combined (Norway).

The participants were also asked about their reasons for choosing some methods over others. The most relevant answers were as follows.

- Only expert judgment
  - "Simple and fast"

  - "Time was of the essence"

- Only Story Points
  - "From Scrum", "It is very common in scrum projects"
  - "Mandated by the customer"
  - "Little time consuming. Easy to get into. Easy to do many times."
  - "It was the one most discussed in the articles and theory on the curriculum"
  - "Give good estimates"
  - "Good and long experience with the use of Story Points"

- Expert judgment and Story Points
  - "By experience it seems to be good enough", "Extensive experience with similar tasks"
  - "This is what they usually use", "It is an established practice. We did not consider whether it should be done in other ways"
  - "Easy to calculate"
  - "In the customers experience this is thought to be fairly accurate"

- Expert judgment, Analogy and Story Points
  - "Fit with the scrum method"
  - "Easy to use"
  - "It is a popular method"

- Expert judgment and Analogy
  - "Because there was a similar project in the past that we used as a reference for our estimations"
  - "Experience and success in many previous projects using these methodologies"

- Analogy and Function Points
  - "Produced good results"

6.4.1.2 Spain

Expert judgment based estimation was used in 77.4% of the projects (see Figure 32), followed by Analogy based estimation, present in 45.2% of the projects. Six out of 33 participants (19.3%) answered that they were using algorithmic models as estimation technique. 12.9% used

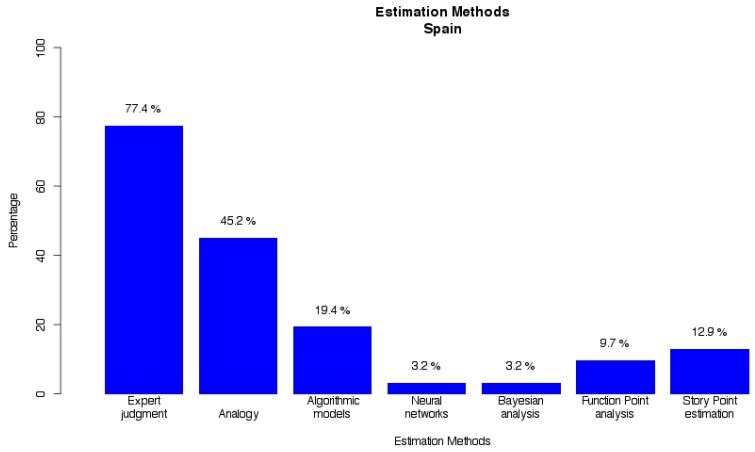


Figure 32 – Estimation Methods (Spain).

Story Point estimation, and 9.7% chose to estimate in Function Points. Bayesian analysis was only used in one project (3.2%), which had the same percentage as the use of Neural Networks.

The combined use of different techniques occurred in 16 of the projects (51.6%). One participant acknowledged the use of each and every presented method in his most recent project. The estimates in eight of the projects (25.8%) were based on a combination of Expert Judgment and Analogy. A combination of Expert judgment, Analogy, and Algorithmic Models was used in two projects (6.4%). The remaining projects all consisted of several different technique cominations. See Figure 33.

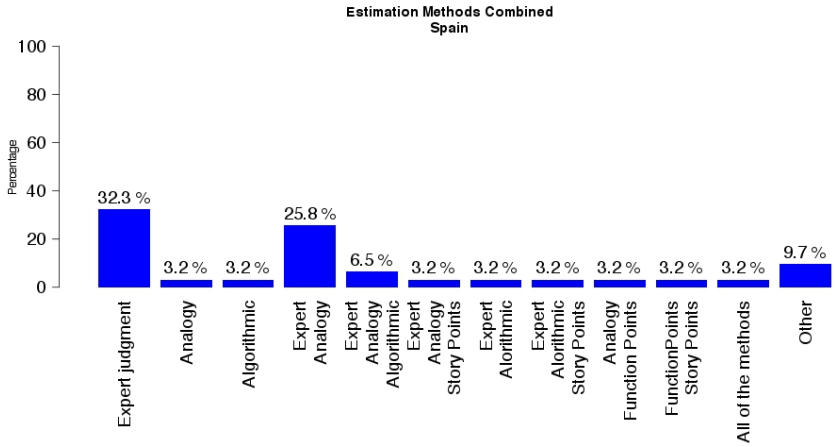


Figure 33 – Estimation Methods Combined (Spain).

When they further were asked about the reason for choosing their preferred method(s) over

others, we got the following answers.

- Only expert judgment
  - "Lack of resources"
  - "Previous experience", "Experience"
  - "Usual in the field", "We always use this kind of methods"
  - "The boss decided to use this method"
- Expert judgment and Analogy
  - "Proven to be good and agile enough"
  - "We have used these methods in other projects. Experience I guess"
  - "These are the most intuitive and direct methods"
  - "The project was too big for using agile methods"
- Expert judgment, Analogy and Story Points
  - "The client does not require an accurate estimate of the duration of the project, as long as there is continuous supervision. Therefore, we can make initial estimates and adjust these as we go along in the project if necessary. Moreover, the knowledge and experience of the team, due to previous similar projects, help us to make accurate estimates"
- Expert judgment, Analogy and Algorithmic Models
  - "Experience"

6.4.1.3 Comparison

The results in Figures 30 and 32 show that expert judgment based methods were the most popular in both Norway and Spain. However, it is noteworthy that the use of story points in Spain had a significantly lower percentage compared with Norway. I first thought this was related to the use of agile methods, which was almost double in Norway, but as it turned out this was not the only reason. Less than half of the people who were using agile methods in Spain estimated with story points, whereas 75% in Norway did the same. In addition, analogy was utilized in many projects, having a similar use rate in the two countries. Other methods such as algorithmic models, bayesian analysis, neural networks, or function point analysis, were close to unused.

Regarding the combination of methods, the only pattern we see is that the combination of expert judgment and analogy occurs quite frequently in both countries.

6.4.2 Estimation units

6.4.2.1 Norway

The projects were estimated using different estimation units. From Figure 30 we are able to read that the project was estimated using Story Points in 72.9% of the cases. In Figure 36 we can see that the second most used estimation unit was time (56.8%), this being based on hours, days, weeks, months, or even years. 16.2% of the projects were estimated in Ideal Days, an Agile estimation unit described in 2.3.2.3. LOC (lines of code)-estimate and Function Points analysis were only used in one project.

As we can see in Figure 34, many projects (35.1%) were estimated solely by story points, followed by projects where a combination of story points and time was used (27%). 18.9% chose time as a unique measure unit for their projects. Other combinations were as follows: story points and ideal days (2.7%); story points, ideal days and time (8.1%); only ideal days (5.4%); Function Points analysis, LOC, and time estimates (2.6%).

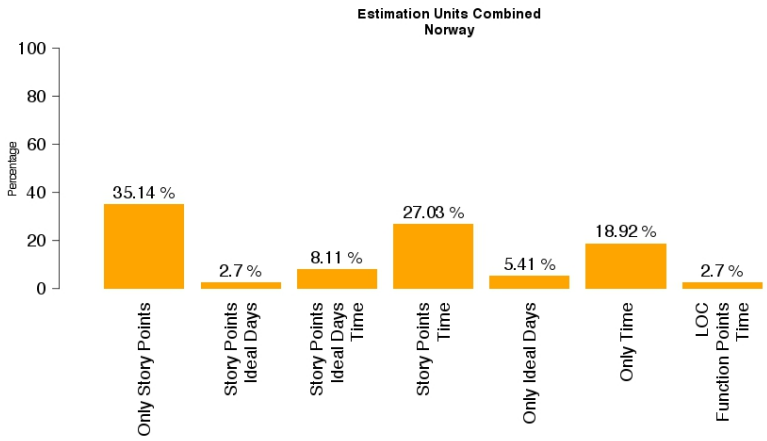


Figure 34 – Estimation Units Combined (Norway).

6.4.2.2 Spain

In Figure 36 we find that all of the participants, thus making out a total of 100%, used time as estimation unit. This may be based on hours, days, weeks, months, or even years. Other estimation units, on the other hand, showed lower rates of use. Story Points and LOC (lines of code) were used in 12.9% of the cases each. 9.7% of the projects were estimated in Ideal Days, whereas another 9.7% used Function Points as estimation units.

As shown in Figure 35, most of the projects (71%) were only estimated in time units. The rest of the projects were estimated in multiple units: Story points and time (3.2%); Story points, time, and lines of code (3.2%); Story points, ideal days, and time (3.2%); Story points, ideal days, time, and function points (3.2%); Story points, time, and function points (3.2%); Time and LOC (9.7%).

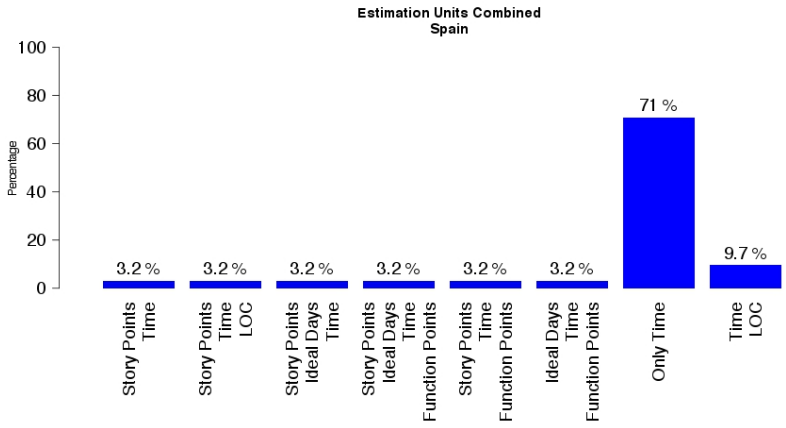


Figure 35 – Estimation Units Combined (Spain).

6.4.2.3 Comparison

Figure 36 shows the comparison between Norway and Spain at the time of analyzing the estimation units in which the projects were estimated. All the projects in Spain were estimated by time or duration, whereas this only happened in half of the cases in Norway. Projects in Norway tended to be estimated by relative measures (story points or ideal days) rather than by time. I want to clarify that this does not mean that the projects' duration was not estimated by time at all. It is a normal practice to derive the time using the team's velocity, as explained in Section 2.3.2.2.

6.4.3 Agile estimating

6.4.3.1 Norway

Regarding agile estimation techniques (Figure 39), Planning Poker appeared to be the most used technique in agile projects with 69.4%, followed by the Wall Estimation (Section 2.3.3.3), which was utilized in 19.4% of the cases. The remaining 11.1% did not select any of the previous options.

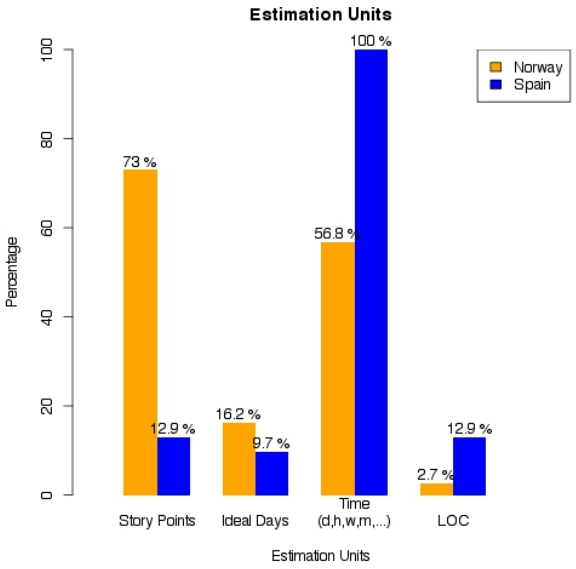


Figure 36 – Estimation Units (Norway vs Spain).

As shown in Figure 40, from the participants who used Story Points for estimating their user stories, 81.8% applied Fibonacci numbers as estimation scale, whereas T-Shirt sizes and other numeric scales were only used by 9.09% of the respondents respectively.

When the participants working in agile projects where estimation is used were asked about the experience level of the team regarding estimating user stories, 40% answered that they had an average level. Furthermore, the experience was distributed normally, or with a Gaussian distribution, around an average. See figure 37.

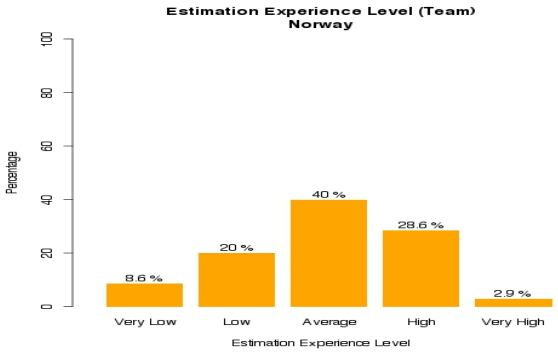


Figure 37 – User Story estimation experience level (Team) (Norway).

The participants who were working on a project with multiple teams, were asked a specific question regarding the estimation across teams. More specifically, they were asked for the tech-

niques they had used for helping estimation across teams. The possible answers are presented on page 34. As we can see in Figure 41, the technique in Option A was implemented in 10 out of these 15 projects (66.7%), whereas seven projects (46.7%) used the technique presented in Option B, and only three (20%) used the approach in Option C. Some projects used a combination of multiple techniques.

6.4.3.2 Spain

Regarding agile estimation techniques (Figure 39), Planning Poker appeared to be the most used technique accounting for 30% of the answers among the respondents. Following this was Wall Estimation which was utilized in 20% of the cases. However, the remaining 50% did not select any of the previous options. The question was only asked to people involved in agile projects with estimates, which were 10 out of 33 respondents.

As shown in Figure 40, from the participants who used Story Points when estimating their user stories, 25% applied Fibonacci numbers as estimation scale, whereas other numeric scales were used by the majority with 75% response rate. T-Shirt sizes were not used in any of the projects.

When the participants working in agile projects with estimations were asked about the experience level of the team regarding estimating user stories, 44.4% answered that they had an average level, whereas another 44.4% claimed to have a high level of experience. Contrarily, 11.1% of the respondents claimed that their team had very low experience. See figure 38.

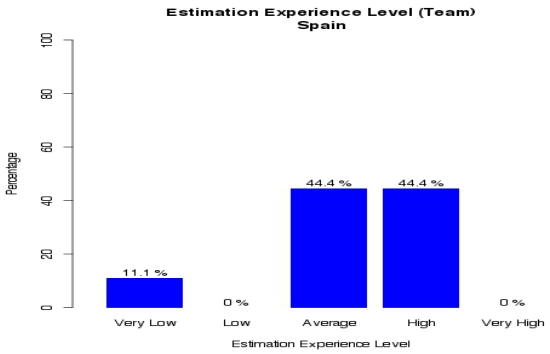


Figure 38 – User Story estimation experience level (Team) (Spain).

As we can see in Figure 41, the technique in Option A was implemented in four out of 11 projects (36.4%) with multiple teams, whereas another four projects (also making out 36.4%) used the technique presented in Option B. The C-option was not selected by any of the respondents. The presented options are described on page 34.

6.4.3.3 Comparison

If we have a look at Figure 39, we see a big difference between Norway and Spain in the use of planning poker as estimation technique. It appeared to be an established technique in Norway, but we cannot say the same for Spain, where only one third of the agile projects used planning poker when estimating.

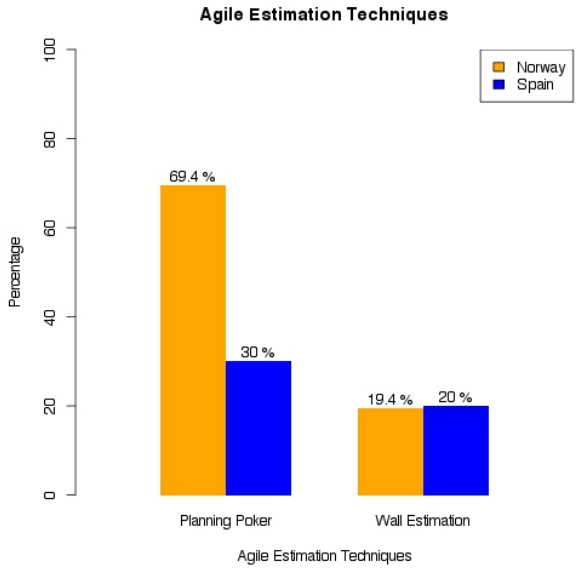


Figure 39 – Estimation Agile Techniques (Norway vs Spain).

Regarding story point estimation scales (Figure 40), we also observe differences in the results of the two countries. Fibonacci sequence was used in almost 75% of the agile projects in Norway, whereas it happened to be less popular in Spain by only being used in 25% of the cases. Instead, other numeric scales were applied in Spain. Only in a minority of the projects were T-shirt sizes being utilized, and these cases only occurred in Norway.

Estimation across agile teams is a challenging task. As we can see in Figure 41, carrying out a joint planning poker session with members from all the teams was the standardized way in Norway (66%), in addition to identifying stories in the backlog that were representative of each story point size and making those stories known to all of the teams (46.7%). In Spain, on the other hand, both techniques were below 40%. It is also good to mention that SAFe's approach for normalizing story points was solely applied in Norway, and only in 20% of the cases.

The respondents in Spain claimed to have a slightly higher level of experience than did the ones in Norway. However, the difference was not significant.

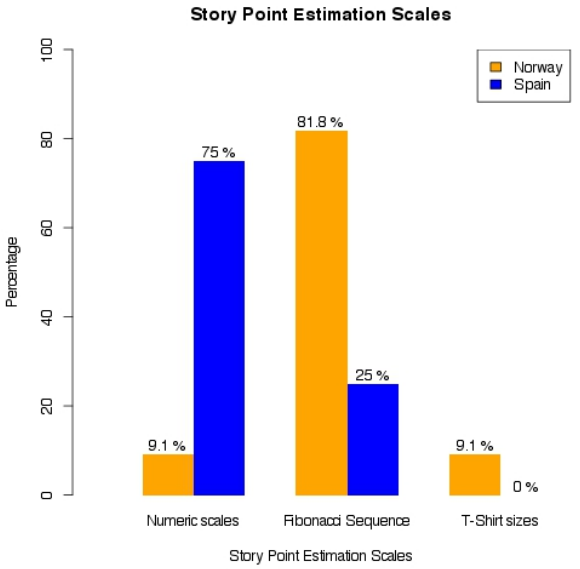


Figure 40 – Story Point estimation scales (Norway vs Spain).

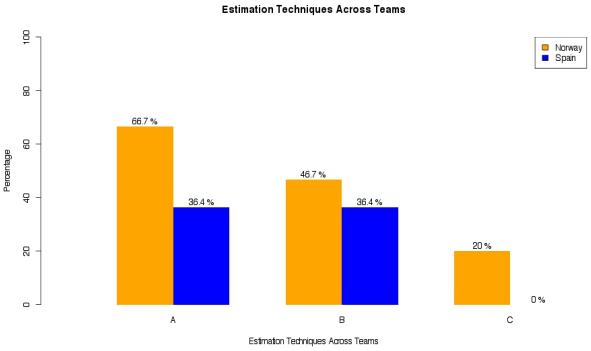


Figure 41 – Estimation Across Teams (Norway vs Spain).

6.4.4 Success and satisfaction

6.4.4.1 Norway

When the participants who had used estimates in their project were asked about the success of the estimates, the majority claimed them to be successful. In addition, two people thought their estimates were very successful. 10.8% declared their estimates as unsuccessful, and the remaining 29.7% admitted that their estimates were neither successful or unsuccessful. See Figure 42.

They were also asked for their overall satisfaction on the utilized estimation methods. The results in Figure 43 show that success and satisfaction were mutually related, overall giving identical results.

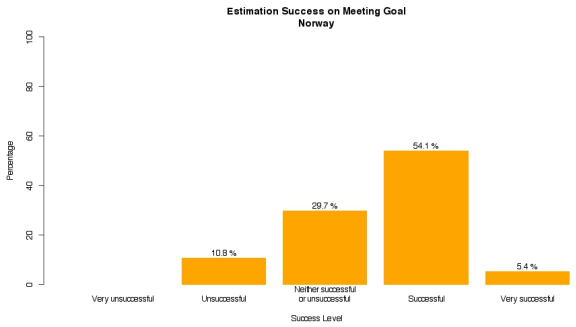


Figure 42 – Estimation Success On Meeting Goal (Norway).

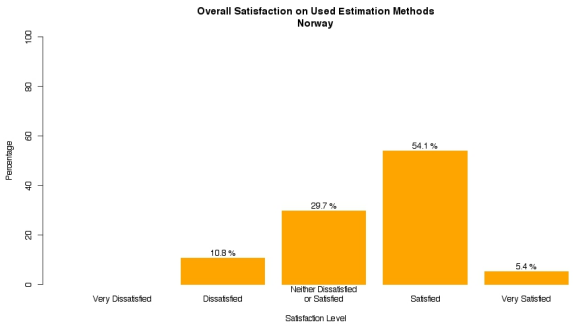


Figure 43 – Overall Satisfaction on Used Estimation Methods (Norway).

In the case that they were not fully satisfied, I asked the participants for suggestions and possible changes regarding the estimation process.

Two respondents warned about the risk of fixed price contracts in agile projects:

- "Try and avoid this, it puts a lot of strain on the teams."
- "We end up taking a high risk for this kind of work, as customers normally want fixed price projects. I believe we should not offer fixed prices for this kind of work."

A scrum practitioner admitted that they tend to think in hours instead of size when estimating user stories, and he suggested more practice as a solution.

A team manager suggested to stop estimating, and delivering the project in smaller batches instead. Another respondent also thought the estimates were pointless, claiming that the tasks presented at the planning meetings were rarely the tasks that they ended up performing. This

might be related to the answer of another respondent, who said user stories were too imprecise, and therefore it was hard to track changes and deviations.

One of the participants, who had been working as a developer for many years, wished to reduce the duration and frequency of the estimation meetings, which he found to be time consuming and too often.

6.4.4.2 Spain

When the participants who had used estimates in their project were asked about the success of the estimates, the majority described it as successful. In addition, one person thought their estimates were very successful. 16.7% declared their estimates as unsuccessful, and the other 23.3% admitted that their estimates were neither successful nor unsuccessful. See Figure 44.

They were also asked about their overall satisfaction on the used estimation methods. 51.6% appeared to be satisfied, whereas only 6.5% were dissatisfied. 35.5% thought their estimates were neither successful nor unsuccessful. The results in Figure 45 show that success and satisfaction were almost mutually related, though with some deviations.

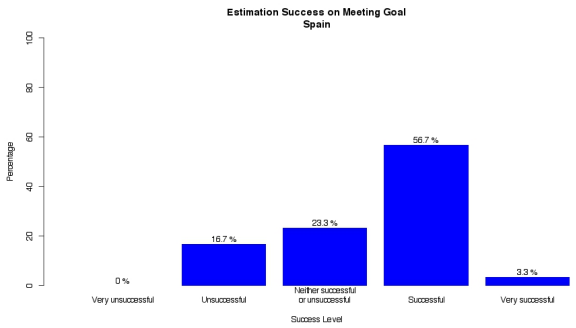


Figure 44 – Estimation Success On Meeting Goal (Spain).

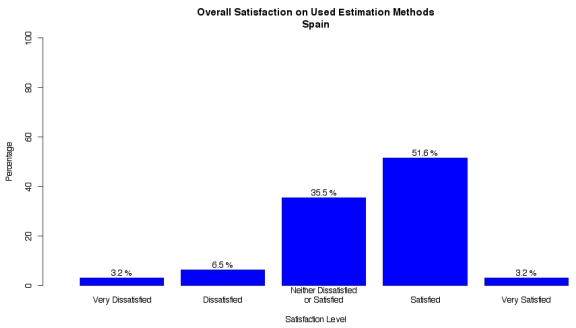


Figure 45 – Overall Satisfaction on Used Estimation Methods (Spain).

One participant emphasized the importance of the communication between the team and the customer, and wished for communication improvement in his project.

Two respondents answered that they would like to use agile methods instead of traditional ones, the success of agile methods being the main reason for this.

Finally, a software architect who had previous experience with Function Points and analogy based estimation techniques, requested the use of COCOMO, and an improvement on the use of analogy.

6.4.4.3 Comparison

The success of the estimates turned out to be notably similar in both Norway and Spain, with a tendency towards successful. See Figure 42 and Figure 44.

When it comes to satisfaction level, we also have significantly similar results in both countries. Overall, the satisfaction on the used estimation methods appeared to be high. Figure 43 and Figure 45.

6.5 Customer

6.5.1 Norway

As previously shown in Figure 14, a customer, or a product owner, was involved in 36 out of 44 projects (81.8%). A section in the survey called "About the Customer" was answered by those participants who stated to be working on a project where a customer was involved.

The communication frequency between the client and the software development team varied from project to project. However, most of the projects showed frequent communication between the two. 54.3% had daily communication, and 40% kept in contact on a weekly basis. Only 5.7% of the projects had monthly communication. See Figure 46.

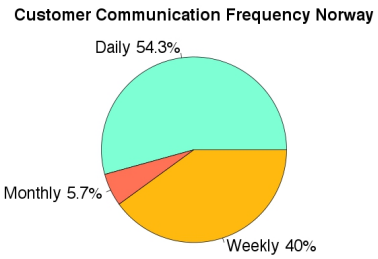


Figure 46 – Customer Communication Frequency (Norway).

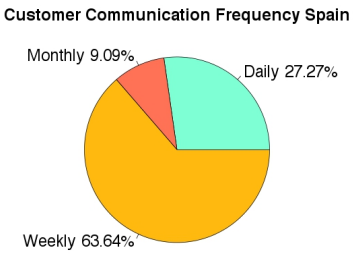


Figure 47 – Customer Communication Frequency (Spain).

When comparing the communication frequency with the success of the estimates (see Table 4), we can see that projects with a daily communication between the team and the customer tend to be more successful than those based on a weekly or monthly basis.

The survey participants were asked to rate the customer on the following properties: Collaboration Skills, IT Competence, Decision Making Ability, and Clarity of Project Goals. The options were "Very Poor", "Poor", "Average", "Good", and "Very Good".

The results showed that more people tended to answer that the collaboration skills of the customers were good (30.6%) and very good (30.6%), rather than poor (19.4%) and very poor

	Norway			Spain		
	Daily	*Weekly*	*Monthly*	*Daily*	*Weekly*	*Monthly*
Very unsuccessful	0	0	0	0	1	1
Unsuccessful	0	1	0	0	4	1
Neither successful nor unsuccessful	3	5	1	2	2	1
Successful	10	8	1	2	6	0
Very successful	2	0	0	0	0	0

Table 4 – Communication Frequency against success of the estimates (Norway vs Spain).

(0%). See Figure 48.

Figure 48 – Customer Collaboration Skills (Norway).

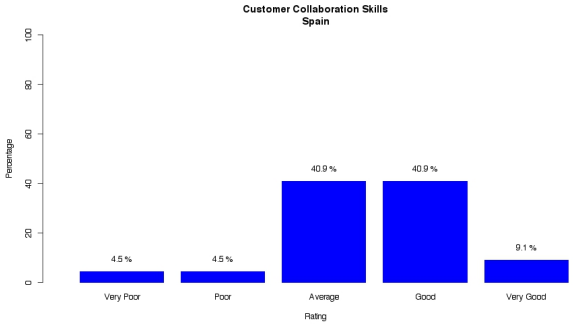


Figure 49 – Customer Collaboration Skills (Spain).

When comparing the collaboration skills of the customer with the success of the estimates (see Table 5), we do not see any reasonable difference between the options.

	Norway					Spain				
	Very poor	*Poor*	*Average*	*Good*	*Very good*	*Very poor*	*Poor*	*Average*	*Good*	*Very good*
Very unsuccessful	0	0	0	0	0	0	0	0	0	0
Unsuccessful	0	0	0	0	0	0	0	3	2	0
Neither successful nor unsuccessful	0	2	3	3	1	0	0	3	2	1
Successful	0	5	3	6	5	0	1	1	5	1
Very successful	0	0	0	0	2	0	0	1	0	0

Table 5 – Collaboration skills against success of the estimates (Norway vs Spain).

Similar results were found when analyzing the customers' IT competence (Figure 50), showing a tendency towards good (25.7%) and very good (29.6%). However, another 29.6% agreed that customers' IT competence was poor.

Figure 50 – Customer IT Competence (Norway).

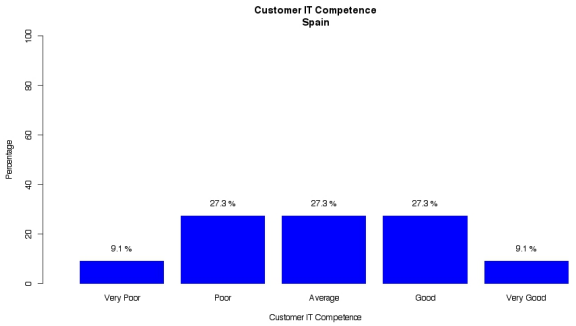


Figure 51 – Customer IT Competence (Spain).

Success of the estimates in relation with customers' IT competence was also analyzed (see Table 6). No trend was discovered between these two variables.

	Norway					Spain				
	Very poor	*Poor*	*Average*	*Good*	*Very good*	*Very poor*	*Poor*	*Average*	*Good*	*Very good*
Very unsuccessful	0	0	0	0	0	0	0	0	0	0
Unsuccessful	0	0	0	0	0	1	0	1	3	0
Neither successful nor unsuccessful	0	2	1	5	1	0	2	3	1	0
Successful	1	7	4	3	4	0	4	1	1	2
Very successful	0	0	0	0	2	0	0	1	0	0

Table 6 – IT competence against success of the estimates (Norway vs Spain).

The decision making ability of the customers also tended towards good (27.8%) and very good (22.2%), rather than poor (5.6%) and very poor (8.3%). However, a high rate of respondents (33.3%) declared the decision making ability of the customer to be average. See Figure 52.

The success of the estimates in relation with the decision making ability of the customer was also considered (see Table 7). When the decision making ability was very good, the success of the project showed a tendency towards very successful. In spite of this, in the projects where the decision making ability of the client was claimed to be very poor, the estimates were still successful.

Figure 54 shows the clarity of project goals that the customers "held" or "possessed". The results showed a tendency towards good (27.8%) and very good (19.4%), even tough a significant number of participants claimed it to be average (25%) or even poor (22.2%).

The success of the project showed the best results when the goal clarity of the customer was very good (see Table 8). However, in the projects where the goal clarity was said to be very

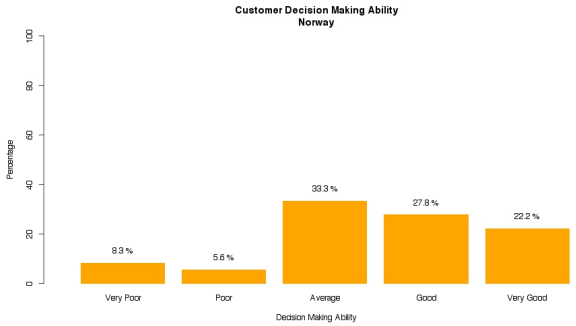


Figure 52 – Customer Decision Making Ability (Norway).

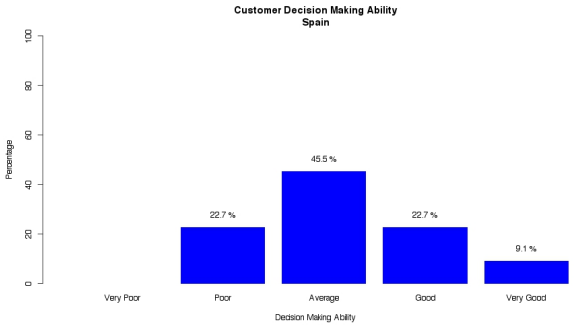


Figure 53 – Customer Decision Making Ability (Spain).

	Norway					Spain				
	Very poor	*Poor*	*Average*	*Good*	*Very good*	*Very poor*	*Poor*	*Average*	*Good*	*Very good*
Very unsuccessful	0	0	0	0	0	0	0	0	0	0
Unsuccessful	0	0	0	0	0	0	1	2	2	0
Neither successful nor unsuccessful	1	0	4	2	2	0	0	4	2	0
Successful	2	2	7	5	3	0	1	4	1	2
Very successful	0	0	0	0	2	0	1	1	0	0

Table 7 – Decision making ability against success of the estimates (Norway vs Spain).

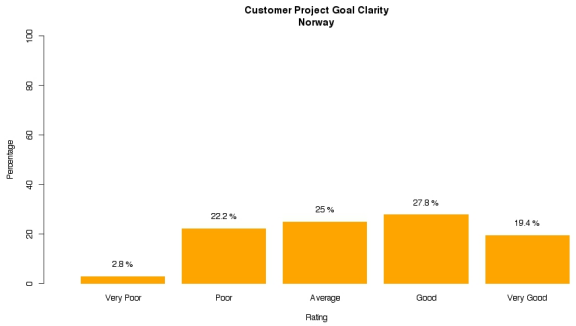


Figure 54 – Customer Project Goal Clarity (Norway).

Figure 55 – Customer Project Goal Clarity (Spain).

poor, the estimates were still successful. Based on this, we are therefore not able to draw any conclusions.

	Norway					Spain				
	Very poor	*Poor*	*Average*	*Good*	*Very good*	*Very poor*	*Poor*	*Average*	*Good*	*Very good*
Very unsuccessful	0	0	0	0	0	0	0	0	0	0
Unsuccessful	0	0	0	0	0	0	1	2	2	0
Neither successful nor unsuccessful	0	3	3	1	2	0	2	2	1	1
Successful	1	5	4	6	3	0	3	3	0	2
Very successful	0	0	0	0	2	0	1	0	0	0

Table 8 – Goal clarity against success of the estimates (Norway vs Spain).

6.5.2 Spain

As previously shown in Figure 20, a customer, or a product owner, was involved in 22 out of 33 projects (66.7%). A section in the survey called "About the Customer" was answered by those participants who stated that they were working on a project where a customer was involved.

The communication frequency between the client and the software development team varied from project to project. However, most of the projects showed frequent communication between the two. 27.3% had daily communication, and 63.6% kept in contact on a weekly basis. Only 9% of the projects had monthly communication (see Figure 47).

We can see that projects with a daily communication between the team and the customer tend to be more successful than those based on a weekly or monthly basis (see Table 4).

The survey participants were asked to rate the customer on the following properties: Collaboration Skills, IT Competence, Decision Making Ability, and Clarity of Project Goals. The options were "Very Poor", "Poor", "Average", "Good", and "Very Good".

The results (Figure 49) showed that more people tended to answer that the collaboration skills of the customers were good and very good, rather than poor and very poor. The rest of the respondents (40.9%) thought the collaboration skills of the customers were average.

In most of the cases where the collaboration skills of the customer were good, the project was successful, whereas when it was claimed to be average, the project had a tendency towards unsuccessful. One participant answered that his project was a success even though the collaboration skills of the client were thought of as poor (see Table 5). With that said, we do not see any trends between these two variables.

When analyzing the customers' IT competence (Figure 51), there were no obvious patterns. The respondents claimed that the customers' IT competence were either poor, average, or good, amounting for 27.3% in each of the mentioned cases. Two participants (9.1%) answered that it was very good, whereas another two thought it was very poor.

No pattern was discovered between the success of the estimates and customers' IT compe-

tence (see Table 6).

The decision making ability of the customers was average in 45.5% of the cases. Five people (22.7%) stated that the customers' decision making ability was poor, whereas another five respondents(22.7%) thought this was good. Only 9.1% answered this to be very good. See Figure 53.

When comparing the decision making ability of the client with the success of the estimates, we do not see any tendencies in the data (see Table 7), and therefore we could say that there is no obvious relation between these two variables.

Figure 55 shows the clarity of project goals that the customers "held" or "possessed". The results showed a tendency towards poor, even tough a significant number of participants claimed it to be average or even good and very good.

The success of the project showed the best results when the goal clarity of the customer was very good, but in fact also when it was poor (see Table 8). Therefore, we could say that success and goal clarity are not related.

6.5.3 Comparison

The communication frequency was daily in most of the projects in Norway, followed by projects with weekly contact. In Spain, on the other hand, the communication with the customer was on a weekly basis in most of the cases, followed by projects with daily communication. Projects where the communication with the customer was on a monthly basis were few in both countries. See Figures 46 and 47.

As shown in Figures 48 and 49, the collaboration skills of the customers appeared to be better in Norway, where more participants selected the "Very good"-option, than in Spain. In contrast, 19.4% thought the collaboration skills were poor or very poor in Norway, whereas in Spain these were only 9.1%.

Regarding customers' IT competence (Figures 50 and 51), we see that the customers in Norway had an overall slightly better IT competence than the clients in Spain.

The decision making ability of the customer also differed between Norway and Spain. It showed that the customers from Norway had a slightly better decision making ability than its counterpart. See Figures 52 and 53.

When analyzing the clarity of project goals that the customer held (Figures 54 and 55), we can see that the customers in Spain had better goal clarity than the ones in Norway.

6.6 Roles

This section of the survey was related to the roles of the people working in the project during the estimation process.

The estimation process normally involves people with different job positions. This is in order to have a better insight of the scope of the project. Therefore, the participants of the survey were asked about who was involved in the estimation process in their projects.

To clarify, it is necessary to say that participating in the estimation process might not mean the act of estimating itself, as it might mean to only take part in the estimation meetings. Also, the term "final estimates" can be further elaborated. Final estimates are estimates related to the total cost of the project.

6.6.1 Norway

As we can see in Figure 58, the developers were part of the estimation process in almost all of the cases. Following them were the team managers or leaders, who took part in more than half of the projects. Software architects helped estimating in 45.9% of the projects, whereas the customer was part of the estimation process only in a quarter of the cases.

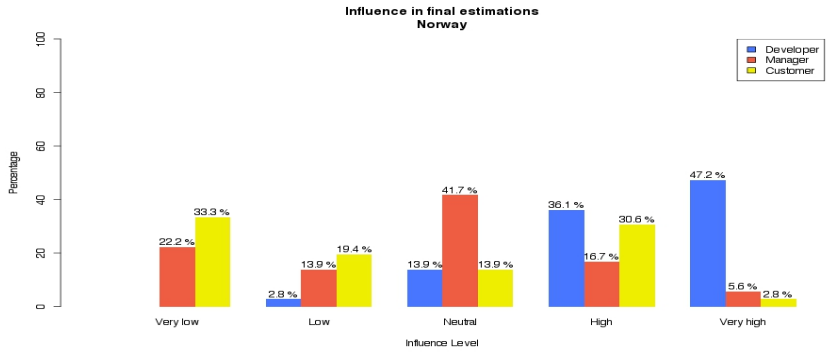


Figure 56 – Influence in final estimations (Norway).

They were also asked about the influence of different people in the project, defined by the others' job position (Figure 56). A high amount of respondents said that the developers were the ones who had the most influence in the final estimates. 47.2% found the influence of the developers to be very high, and 36.1% answered it to be high.

The influence of the managers appears to be neutral in 41.7% of the cases. However, the results show that the managers had different influence level from project to project. They turned out to be very low in 22.2% of the projects, which was in contrast with the 16.7% who found the influence of the managers to be high.

In the case of the customer, we can also see a contrast in the results. It is important to highlight the 33.3% who said that the influence of the customer in the final estimations was very low, against the 30.6% who said that is was high. It is also interesting to take into account, that in only one of the cases (2.8%) the influence of the customer regarding the final estimates appeared to be very high.

In the closing of the section, the participants were asked if the developers were bound to estimations made by others in higher positions. In almost all of the cases the developers were not bound to estimations made by others. Only 11.4% said the opposite. Figure 62.

6.6.2 Spain

As we can see in Figure 58, the developers and team managers were part of the estimation process in 67.7% of the cases. Software architects helped estimating in 38.7%, whereas the customer was part of the estimation process only in 25.8% of the projects.

Figure 57 reveals that the influence of the different job positions was approximately flat in each satisfaction level. However, the satisfaction level varied. Close to 50% of the respondents thought that the influence level of all the parts involved was high, meaning that there was no position that had more control regarding the estimates.

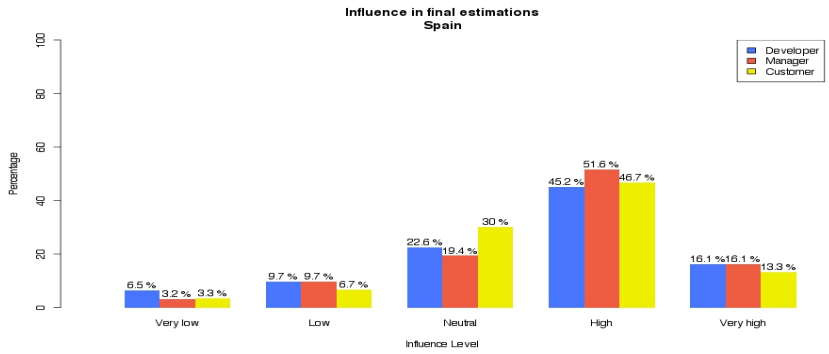


Figure 57 – Influence in final estimations (Spain).

In the closing of the section, the participants were asked if the developers were bound to estimations made by others in higher positions. Almost an equal amount of the participants answered "Yes" (44.8%) compared to the ones who answered "No" (55.2%). See Figure 62.

6.6.3 Comparison

In both countries developers were on top of the list when it comes to participating in the estimation process (Figure 58). However, there was a noteworthy difference between the countries.

In Norway, cases where the developers did not participate rarely occurred, whereas in Spain, on the other hand, more than a quarter of the projects did not include developers in the estimation process. Regarding the involvement of other job positions, there was not significant difference between the countries.

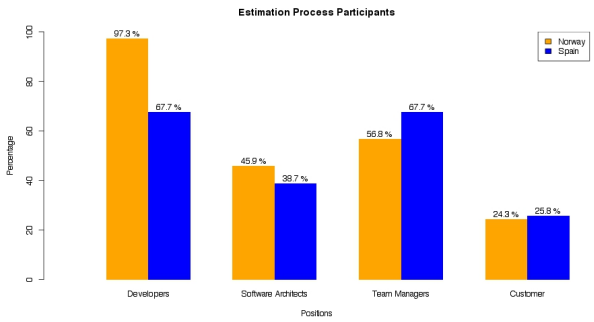


Figure 58 – Estimation Process Participants (Norway vs Spain).

In the sample from Spain, developers turned out to have a high influence in the final estimates, while in Norway they had a very high influence. See Figure 59.

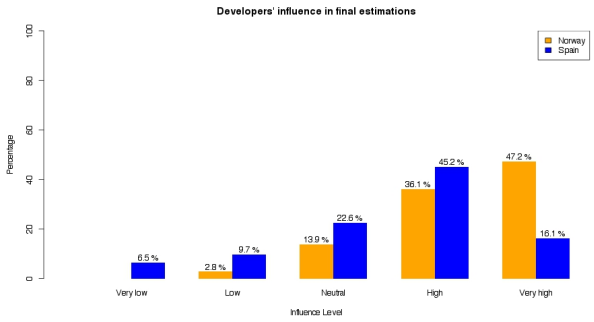


Figure 59 – Developers' influence in final estimations (Norway vs Spain).

The influence of the managers differed in Norway and Spain (Figure 60). In Norway, their influence was overall low, whereas in Spain, it was more towards high.

Regarding customers' influence on the final estimates, the results from Spain showed a higher influence than the results from Norway. In Norway it did not seem to be an agreement between the respondents, whose answers were spread between the options from very low to high. See Figure 61.

As we can see in Figure 62, when analyzing if the developers were bound to estimates made by others in a higher position, the results from Spain and Norway showed significant

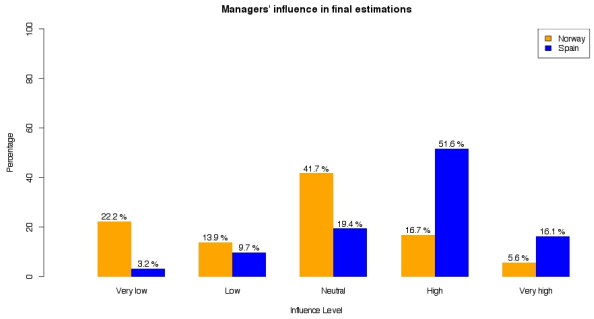


Figure 60 – Managers' influence in final estimations (Norway vs Spain).

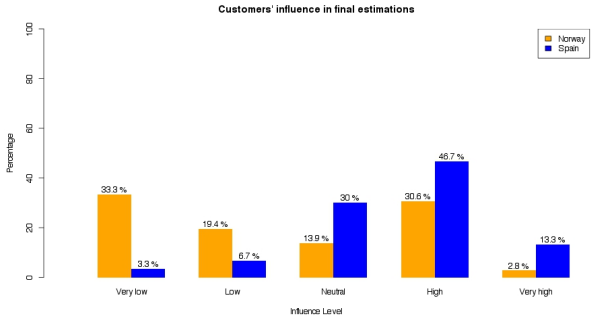


Figure 61 – Customers' influence in final estimations (Norway vs Spain).

differences. In Spain, "Yes" and "No" answers were almost fifty fifty, whereas in Norway the case in which the developers were bound to estimates made by others was rare.

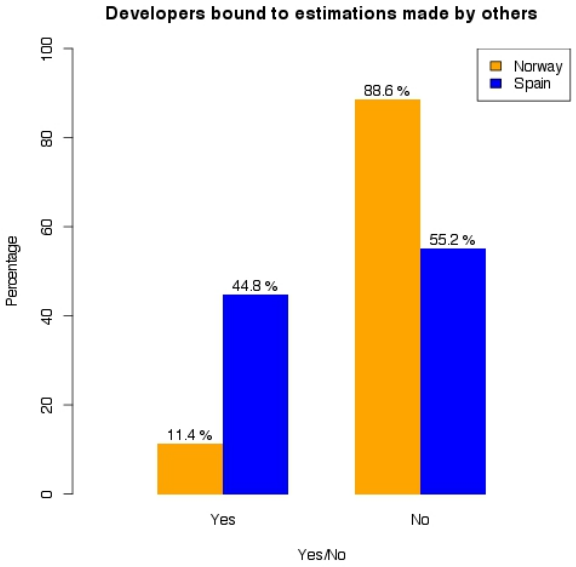


Figure 62 – Developers bound to estimations made by others (Norway vs Spain).

7 Discussion

In this chapter I will examine my findings, and compare them with the findings of other researches (section 2.4). I want to see whether they correlate or differ.

Each of the research questions will be discussed in a sequential manner. In addition, the validity and reliability of my study will be reviewed.

7.1 RQ1: Which software effort estimation methods are Norwegian companies using?

Estimation methods based on expert judgment and analogy were the most popular within the sample of Norwegian companies. The most used method was story point estimation, which does indeed involve both expert judgment and analogy.

This could be caused by the fact that most of the projects used agile development methods, and estimating story points is, as a matter of fact, a common practice in agile. The use of these methods may also be the result of them showing high levels of success and satisfaction. These methods are built on the known notion that comparing tasks is considerably easier than estimating time.

Other main estimation methods introduced by the literature, such as parametric models, bayesian analysis, neural networks, or function point analysis, showed a particularly low usage within the Norwegian companies.

My discoveries are in correlation with other studies conducted in the same field of research. That is to say, these studies support my findings in that expert judgment based methods were the most popular in the software industry (Moløkken-Østvold and Jørgensen, 2003; Moløkken-Østvold, Jørgensen, Tanilkan, Gallis, Lien and Hove, 2004; Trendowicz et al., 2011).

The scarce use of algorithmic model-based methods that my study revealed, did not come as a surprise as these results were similar to the ones found in the study carried out by Yang et al. (2008).

There seems to be an agreement among researchers on why expert judgment based techniques predominate the ones based on algorithmic models. This agreement says that there is not enough evidence to support the theory that formal models lead to better estimates (Moløkken-Østvold and Jørgensen, 2003; Yang et al., 2008). Based on the point of view of the respondents in my study, expert judgment-based estimates were thought of as successful, and thus no other formal estimation models were considered.

Regarding agile techniques, planning poker was used in the majority of the projects. The

literature states that planning poker is a common practice in agile development. Most of the projects in the Norwegian sample were estimated using the Fibonacci sequence as the story point scale. A typical planning poker card deck normally has the numbers of the Fibonacci sequence. Based on my experience, the use of Fibonacci sequence simplifies the estimation of bigger tasks. It avoids the uncertainty of whether a task is an 8 or a 9, an instead lets you chose between an 8 or a 13 which, without a doubt, is less complicated.

7.2 RQ2: Do they combine different methods?

Assuming that story point estimation includes both expert judgment and analogy, there were few cases where only one single method was used for generating estimates. In most of the projects, different combinations of expert judgment, analogy and story point estimation were common.

My study shows that the reason for using these methods is their simplicity and their success on giving good estimates, as well as being a convention in agile development. However, the accuracy of the estimates was not analyzed in this study, and therefore the success of the estimates was entirely dependent on the respondents' opinion.

Previous studies (Winkler, 1989; Jørgensen et al., 2009) have claimed that when a combination of different estimation methods was used, the estimates were more accurate and therefore more successful. In regards to my findings, I am not able to say that the combination of methods gives better results. This is because the majority of the projects in my study utilized a combination of different approaches, and thus I did not have enough data to compare these with. Nonetheless, the combination of methods did generally produce good results.

7.3 RQ3: Who takes part in the estimation process, and what is the role of each participant?

It is necessary to emphasize that participating in the estimation process might not mean the act of estimating itself, as it could mean to only participate in the meetings.

Development Team

The development team, which includes all programmers, testers, database engineers, analysts, user interaction designers and so on, took part in the estimation process in virtually all of the Norwegian projects considered in this thesis. According to my study, developers were the ones who influenced the final estimates the most, and they were generally not bound to estimates

made by others in higher positions. Agile software development states that developers are the ones who estimate, and Norway does not seem to be an exception in this regard. With that said, cautiousness is needed before we draw any definite conclusions since "taking part in the estimation process" might not mean estimating, but taking part in the meetings.

Team Manager or Scrum Master

The manager, or Scrum Master in scrum projects, should be the facilitator of the team, and therefore he needs to be part of the estimation meetings according to agile development. However, he should not estimate the tasks unless he is also a developer, which might be the case in some projects. In my study, around half of the Norwegian participants said that the manager did not take part in the estimation process. It is not clear though if the managers did not attend the meetings, or if they did not estimate. In regards to how managers influenced the final estimates, overall results showed that their influence was neutral in most of the projects.

Customer

In the agile world, the customer should be part of the estimation meetings to answer any questions the developers might have, but as in the case of the managers, they do not estimate. My study showed that only one quarter of the Norwegian projects included the customer in the estimation process. Again, it is not clear if this means being a part of the meetings, or estimating the tasks.

7.4 RQ4: To what extent can the customer influence the estimation process?

I explained the involvement of the customer when answering RQ3 in section 7.3. Additionally, I will focus this section towards how the customer might influence the estimation process.

Influence on the selection of methods

Some respondents declared that they chose specific estimation methods mandated by the customer. This clearly shows the high influence a customer might have in some projects, actually deciding which methods will be utilized. Nonetheless, this was not a general rule, but rather a limited and interesting observation on when the estimation process had in fact been influenced by the customer.

Influence in the final estimates

The contradictory results in regards to customers' influence in the final estimates did not lead me to any clear conclusions. In some projects it appeared to be high, whereas it was low in others. Data was gathered by a single question: "To which extent does the customers' decision influence the final estimates". The way it was created was seemingly not the best option, and I see that I instead should have defined the meaning of "final estimates" beforehand, in order to avoid confusion.

Collaboration and communication frequency

Collaboration with the customer is key for producing better results in software projects (Chow and Cao, 2008; Bakalova and Daneva, 2011). According to my study, projects with a daily communication between the team and the customer tend to be more successful than those based on a weekly or monthly basis. Similar results were also found by Furulund (2007), who observed that projects with daily contact between contractors and customers had lesser effort overruns than other projects did.

The rapid feedback from the customer helps the team adapt to changes. It also makes it easier for the team to always have a full grasp of the goal or scope of the project at any given time of the development process. After all, the customer is the one who finances the project, and because of this decides what needs to be done. Hence, he should be close to the development team during the whole project.

Customer skills

Communication frequency is however not the only feature regarding the customer and the project. Customer skills such as IT competence, goal clarity, collaboration skills or decision making ability, might also influence the success of the estimates. In my study, there was not found any patterns that could lead us to claim that there is a relation between customer skills and success of the estimates.

7.5 RQ5: Do Spanish companies estimate in the same way as Norwegian companies do?

The results of this question were presented and analyzed in the comparison section for each specific topic in Chapter 6. In this section I will, based on my results, explain why the estimation practice differed between the projects in Norway and Spain.

The differences between the countries in regards to the estimation practice were significant. The phase of the project in which the estimates were made, the chosen estimation approaches and techniques, the units in which the projects were estimated, the agile estimating techniques, and also the people who participated in the estimation process, differed. Each of them so to a different extent in the two countries.

The only explanation I can see for this is the use of agile methods, and the adaptation to new techniques. The chosen development methods influence the estimation process in a substantial manner. Estimating projects where traditional development methods are used, requires the estimates to be done before the development process, whereas in agile projects estimation is a continuous process that goes throughout the project, and involves estimating in each development cycle or sprint. Surprisingly, agile methods were used three times more in Norway than in Spain, and according to my study traditional methods were outdated in the Norwegian companies. In Spain, on the other hand, traditional methods, such as the waterfall method, were still practiced in more than one third of the projects. It is also surprising that approximately half of the Spanish participants did not have any experience with agile methods. In regards to new agile estimating techniques, planning poker and Fibonacci sequence are well known terms in the agile estimation world. Therefore, we could say that the adaptation to new techniques has succeeded in Norway, as opposed to Spain where it has yet to be applied with sufficient strength.

The success of the estimates on meeting their goal, and the overall satisfaction on the used estimation methods were significantly similar in both countries. Therefore, the estimation problem does not seem to be a concern for the majority of the respondents, which could be the reason why they do not try different techniques. If something works, why should we change it? However, there have been established facts that projects developed with sequential development models are more likely to experience effort overruns than projects where incremental models are used (Moløkken-Østvold, Lien, Jørgensen, Tanilkan, Gallis and Hove, 2004; Moløkken-Østvold and Jørgensen, 2005). This should serve as a starting point for companies, specially in Spain, to at least give agile methods a try.

7.6 Reliability and Validity

In this section I will use Shuttleworth (2008) when discussing the reliability and validity of my research study.

7.6.1 Reliability

In this section I intend to answer whether the research conducted in this thesis is reliable or not.

The theory of survey research (Chapter 3), and the details of how I designed my survey (Chapter 4), will help others to repeat the research study, in order to investigate if similar results will be produced. The process of gathering data was transparent, avoiding any kind of manipulation from the researcher. Therefore, if a new research was to be conducted within the same population, population signifying the respondents of my survey, it is likely that the results would be similar.

7.6.2 Validity

Am I measuring what I think I am measuring in my research study? Answers to this question will be presented in the following section.

External validity

Campbell et al. (1963) described external validity as follows.

> *"External validity asks the question of generalizability: To what populations, settings, treatment variables and measurement variables can this effect be generalized?"*

Randomization of participants improves the external validity of a research. For this thesis, my goal was to select the participants in the most arbitrary manner possible.

However, the results showed that there was a notable difference in age distribution between the countries. Participants from Spain were overall younger than those from Norway. This may be related to the way I recruited them. As a result of my previous working experience in Norway, I had earned a broader network of more experienced contacts in the IT industry. Though in Spain, people were mostly recruited through classmates I had studied with or via University professors, making this network predominantly younger.

The differences between the project sizes in the two countries might also be due to the previous issue of the respondents overall age group. Participants in the Spanish selection were relatively junior in the IT business, and as a result they did not work in big companies. The Norwegian participants however, had more working experience and thus worked in bigger companies. My assumption with regards to the project size is that bigger companies take on more sizable projects than smaller companies do.

With this being said, the results from Spain can not be generalized to a larger population. The study of the Norwegian sample on the other hand, shows a higher external validity and is therefore more likely to be generalized to the population of Norway.

Internal validity

Could there be an alternative cause, or causes, that explain my observations and results?

The questions from the survey, including the introduction message, are added as appendix to enhance the validity and reliability of the research.

The pilot study I performed before the final survey also reinforces the internal validity of this thesis. It was used in order to avoid any unexpected cause that could affect the results in the final study. Nonetheless, the pilot study failed to detect a mistake I made when creating filters in some of the questions. Instead of hiding the estimation section when the respondent selected "No, there was not made any estimates before or during the project", the filter also hid this section from the respondents who chose the "Other (Specify)"-option. I first discovered this mistake at the time of analyzing the results. Fortunately, only one participant's answers were affected by my lapse.

The general lack of knowledge among the respondents regarding estimation might also have affected the results. I base this statement on the inconsistency in some answers from a selection of the participants.

In hindsight I realize that I ought to have given more details and explanations related to certain terms that might have been unclear for the respondents. These include ideal days, wall estimation, and others.

The participants were asked about the success of the estimates on meeting their goal, without this being defined beforehand. The options ranged from "Very unsuccessful" to "Very successful". I do not think this is the way of measuring the success of the estimations, and instead more detailed data should have been asked related to the accuracy of the estimates (Section 2.2.3). Alternatively, I could have asked the participants an open question, making them elaborate as to what makes the estimates a success in their opinion.

The question "Who took part in the estimation process?" was also poorly formulated, which might have led to confusion as to the meaning of the phrase "taking part in the estimation process". Instead, I should have divided the question in two, making them "Who took part in the estimation meetings?" and "Who estimated the tasks?".

In order to avoid biased answers, the survey must be responded by people in all job positions. In this study, I aimed to achieve this by spreading the survey as wide as I could. However, I was only able to reach one customer in each country, which biased some results due to the lack of

customers' opinions.

8 Conclusion

This chapter will close the thesis with a summary of the findings, and finally it will present ideas for further research on the topic.

8.1 Summary of the Main Findings

Expert judgment and analogy-based estimation approaches were the most popular among the sample of Norwegian companies. The most used method was story point estimation, which does indeed involve both expert judgment and analogy. My discoveries were in correlation with other studies conducted in the same field of research.

Assuming that story point estimation includes both expert judgment and analogy, in Norway there were few cases where only one single method was used for generating estimates. In most of the Norwegian projects, different combinations of expert judgment, analogy, and story point estimation were common. The reason for using these methods is their simplicity and their success on giving good estimates, as well as being a convention in agile development.

Regarding agile techniques in Norway, planning poker was used in the majority of the projects, as well as the Fibonacci sequence. These are well known techniques in the agile world.

According to the results from Norway, developers were the ones who influenced the final estimates the most, and therefore they took part in the estimation process in virtually all of the projects. In regards to how managers influenced the final estimates, overall results showed that their influence was neutral in a majority of the cases. When it comes to the customer, they were only included in the estimation process in one quarter of the Norwegian projects.

When analyzing the communication frequency between the team and the customer, projects with a daily communication tended to be more successful than the ones based on a weekly or monthly basis. This was true in both the Norwegian and Spanish samples.

Nonetheless, there were significant differences between Norway and Spain in regards to the estimation practice. The phase of the project in which the estimates were made, the chosen estimation approaches and techniques, the units in which the projects were estimated, the agile estimating techniques, and also the people who participated in the estimation process, differed. Each of them so to a different extent in the two countries. The only explanation I can see for this is the use of agile methods in Norway, and the slow adaptation to new techniques in Spain.

8.2 Further Research

The research on the software estimation process has not yet reached its full potential, and the need for further research on how to improve the ability to produce accurate estimates is still at large.

This thesis shows detailed information about the estimation process, but there is still more data to be analyzed in regards of finding patterns and correlations between different variables. For example, it would be interesting to see the differences, or similarities, between developers, managers and customers when it comes to defining to which extent others can influence the estimation process.

The role of the customer also needs to be further analyzed, since there are few studies in regards to this. In my research study, my aim was to get as much information related to customers as possible, but I failed in trying to reach them. I suggest that future researchers establish a direct contact with customers. In the case of this study, the customers were reached by requesting the development team or the managers to contact the customer on my behalf by forwarding the survey.

Future investigators should also clearly define beforehand what "success of the estimates" means. This definition should involve the opinion of all the project stakeholders, and their respective points of view. In other words, what the customer considers as success may differ from what the team finds successful, and this is an issue that needs to be addressed.

References

Albrecht, A J (1979), Measuring application development productivity', presented at ibm applications development symposium, *in* '7. ISO/IEC 14143-1:1998. 'Software Measurement – Functional Size Measurement – Part 1 Definition of Concepts', pp. 2–11.

Bakalova, Zornitza and Maya Daneva (2011), A comparative case study on clients participation in a'traditional'and in an agile software company, *in* 'Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement', ACM, pp. 74–80.

Beck, Kent (1999), 'Embracing change with extreme programming', *Computer* **32**(10), 70–77.

Beck, Kent (2003), *Test-driven development: by example*, Addison-Wesley Professional.

Beck, Kent, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries et al. (2001), 'Manifesto for agile software development', *The Agile Alliance* pp. 2002–04.

Bertolino, Antonia (2001), 'Software testing', *SWEBOK* p. 69.

Boehm, Barry (1981), *Software engineering economics*, Prentice-Hall advances in computing science and technology series, Prentice-Hall.

Boehm, Barry (1988), 'A spiral model of software development and enhancement', *Computer* **21**(5), 61–72.

Boehm, Barry (2002), 'Get ready for agile methods, with care', *Computer* **35**(1), 64–69.

Boehm, Barry, Chris Abts and Sunita Chulani (2000), 'Software development cost estimation approaches—a survey', *Annals of Software Engineering* **10**(1), 177–205.

Briand, Lionel C and Isabella Wieczorek (2002), 'Resource estimation in software engineering', *Encyclopedia of Software engineering* .

Campbell, Donald Thomas, Julian C Stanley and Nathaniel Lees Gage (1963), *Experimental and quasi-experimental designs for research*, Houghton Mifflin Boston.

Chow, Tsun and Dac-Buu Cao (2008), 'A survey study of critical success factors in agile software projects', *Journal of Systems and Software* **81**(6), 961–971.

Cohn, M. (2005), *Agile Estimating and Planning*, Pearson Education.

Conte, Samuel D., Hubert E. Dunsmore and Vincent Yun Shen (1985), 'Software effort estimation and productivity', *Advances in Computers* **24**, 1–60.

Dybå, Tore and Torgeir Dingsøyr (2008), 'Empirical studies of agile software development: A systematic review', *Information and software technology* **50**(9), 833–859.

Finnie, GR, GE Wittig and JM Desharnais (1997), 'A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models', *Journal of Systems and Software* **39**(3), 281–289.

Fowler Jr, Floyd J (1993), *Survey research methods*, Vol. 1, Sage.

Fowler Jr, Floyd J (1995), *Improving survey questions: Design and evaluation*, Vol. 38, Sage.

Fowler, Martin (2001), Is design dead?, *in* 'Extreme programming examined', Addison-Wesley Longman Publishing Co., Inc., pp. 3–17.

Friendly, Lisa (1996), The design of distributed hyperlinked programming documentation, *in* 'Hypermedia Design', Springer, pp. 151–173.

Furulund, Kristian Marius (2007), Empirical research on software effort estimation accuracy, PhD thesis, Master Thesis, Department of Informatics, University of Oslo.

Gollapudi, K (2004), 'Function points or lines of code?-an insight', *Global Microsoft Business Unit, Wipro Technologies* .

Gray, Andrew R and Stephen G MacDonell (1997), 'A comparison of techniques for developing predictive models of software metrics', *Information and Software Technology* **39**(6), 425–437.

Grenning, James (2002), 'Planning poker or how to avoid analysis paralysis while release planning', *Hawthorn Woods: Renaissance Software Consulting* **3**.

Heiat, Abbas (2002), 'Comparison of artificial neural network and regression models for estimating software development effort', *Information and Software Technology* **44**(15), 911–922.

Highsmith, James A (2000), *Adaptive software development*, Dorset House.

ISO/IEC 12207 (2008), Systems and software engineering software life cycle processes, ISO, Geneva, Switzerland.

ISO/IEC 14764 (2006), Software engineering – software life cycle processes – maintenance, ISO, Geneva, Switzerland.

ISO/IEC 20926 (2009), Software and systems engineering – software measurement – ifpug functional size measurement method, ISO, Geneva, Switzerland.

Jørgensen, Magne (2004), 'A review of studies on expert estimation of software development effort', *Journal of Systems and Software* **70**(1), 37–60.

Jørgensen, Magne (2007*a*), A critique of how we measure and interpret the accuracy of software development effort estimation, *in* 'First International Workshop on Software Productivity Analysis and Cost Estimation. Information Processing Society of Japan, Nagoya', Citeseer.

Jørgensen, Magne (2007*b*), 'Forecasting of software development work effort: evidence on expert judgement and formal models', *International Journal of Forecasting* **23**(3), 449–462.

Jørgensen, Magne, Barry Boehm and Stan Rifkin (2009), 'Software development effort estimation: Formal models or expert judgment?', *Software, IEEE* **26**(2), 14–19.

Jørgensen, Magne and Kjetil Moløkken-Ostvold (2004), 'Reasons for software effort estimation error: impact of respondent role, information collection approach, and data analysis method', *Software Engineering, IEEE Transactions on* **30**(12), 993–1007.

Jørgensen, Magne and Martin Shepperd (2007), 'A systematic review of software development cost estimation studies', *Software Engineering, IEEE Transactions on* **33**(1), 33–53.

Karner, Gustav (1993), 'Resource estimation for objectory projects', *Objective Systems SF AB* **17**.

Kramer, Douglas (1999), Api documentation from source code comments: a case study of javadoc, *in* 'Proceedings of the 17th annual international conference on Computer documentation', ACM, pp. 147–153.

Lacey, Mitch (2012), 'Estimating'. [Online; posted January-2012].
URL: http://msdn.microsoft.com/en-us/library/hh765979.aspx

Larman, Craig (2004), *Agile and iterative development: a manager's guide*, Addison-Wesley Professional.

Larman, Craig and Victor R Basili (2003), 'Iterative and incremental developments. a brief history', *Computer* **36**(6), 47–56.

Métrica v.3 (2001), Metodología de planificación, desarrollo y mantenimiento de sistemas de información, Ministerio de Hacienda y Administraciones Públicas.
URL: *http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html*

Moløkken-Østvold, Kjetil, Anette Lien, Magne Jørgensen, Sinan Tanilkan, Hans Gallis and Siw Hove (2004), 'Does use of development model affect estimation accuracy and bias?', *Product Focused Software Process Improvement* pp. 17–29.

Moløkken-Østvold, Kjetil and Kristian Marius Furulund (2007), The relationship between customer collaboration and software project overruns, *in* 'Agile Conference (AGILE), 2007', IEEE, pp. 72–83.

Moløkken-Østvold, Kjetil and Magne Jørgensen (2003), A review of software surveys on software effort estimation, *in* 'Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on', IEEE, pp. 223–230.

Moløkken-Østvold, Kjetil and Magne Jørgensen (2005), 'A comparison of software project overruns-flexible versus sequential development models', *Software Engineering, IEEE Transactions on* **31**(9), 754–766.

Moløkken-Østvold, Kjetil, Magne Jørgensen, Sinan S Tanilkan, Hans Gallis, Anette C Lien and SW Hove (2004), A survey on software estimation in the norwegian industry, *in* 'Software Metrics, 2004. Proceedings. 10th International Symposium on', IEEE, pp. 208–219.

Oates, Briony J (2005), *Researching information systems and computing*, SAGE Publications Limited.

Olfman, L and Robert P Bostrom (2008), 'End-user software training: an experimental comparison of methods to enhance motivation', *Information Systems Journal* **1**(4), 249–266.

Park, Heejun and Seung Baek (2008), 'An empirical validation of a neural network model for software effort estimation', *Expert Systems with Applications* **35**(3), 929–937.

Parthasarathy, A. (2003), *Practical Software Estimation: Function Point Methods for Insourced and Outsourced Projects*, Pearson Education.
URL: *http://books.google.no/books?id=e5NXNXC4JCcC*

Pendharkar, Parag C, Girish H Subramanian and James A Rodger (2005), 'A probabilistic model for predicting software development effort', *Software Engineering, IEEE Transactions on* **31**(7), 615–624.

Precision Consulting (n.d.), 'Reliability vs validity'. [Online; viewed November-2014].
URL: http://precisionconsultingcompany.com/wp-content/themes/swatch/images/designandco_144_la_0.jpg

Ralph, Paul and Yair Wand (2009), 'A proposal for a formal definition of the design concept', *Design requirements engineering: A ten-year perspective* pp. 103–136.

Rodríguez, Pilar, Jouni Markkula, Markku Oivo and Kimmo Turula (2012), Survey on agile and lean usage in finnish software industry, *in* 'Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement', ACM, pp. 139–148.

Royce, Winston W (1970), Managing the development of large software systems, *in* 'proceedings of IEEE WESCON', Vol. 26, Los Angeles.

Salo, Outi and Pekka Abrahamsson (2008), 'Agile methods in european embedded software development organisations: a survey on the actual use and usefulness of extreme programming and scrum', *Software, IET* **2**(1), 58–64.

Schmitz, Carsten (2012), 'Limesurvey: An open source survey tool'.
URL: http://www.limesurvey.org

Schwaber, Ken and Mike Beedle (2002), *Agile software development with Scrum*, Vol. 18, Prentice Hall PTR Upper Saddle Riverˆ eNJ NJ.

Shepperd, Martin and Chris Schofield (1997), 'Estimating software project effort using analogies', *Software Engineering, IEEE Transactions on* **23**(11), 736–743.

Shepperd, Martin, Chris Schofield and Barbara Kitchenham (1996), Effort estimation using analogy, *in* 'Proceedings of the 18th international conference on Software engineering', IEEE Computer Society, pp. 170–178.

Shepperd, Martin and Gada Kadoda (2001), 'Comparing software prediction techniques using simulation', *Software Engineering, IEEE Transactions on* **27**(11), 1014–1022.

Shuttleworth, Martyn (2008), 'Validity and reliability'. [Online; viewed November-2014].
URL: https://explorable.com/validity-and-reliability

Simon, M. and J. Goes (2011), *Dissertation and Scholarly Research: Recipes for Success: 2011 Edition*, Dissertation & Scholarly Research, CreateSpace Independent Publishing Platform.
URL: http://books.google.no/books?id=PYF-KQEACAAJ

Singer, Eleanor Ed and Stanley Ed Presser (1989), *Survey research methods: A reader.*, University of Chicago Press.

Sommerville, I. (2004), *Software engineering*, International computer science series, Pearson/Addison-Wesley.

Trendowicz, Adam, Jürgen Münch and Ross Jeffery (2011), 'State of the practice in software effort estimation: a survey and literature review', *Software Engineering Techniques* pp. 232–245.

Van Koten, Chikako and AR Gray (2006), 'Bayesian statistical effort prediction models for data-centred 4gl software development', *Information and Software Technology* **48**(11), 1056–1067.

Webster, Jane and Joseph J Martocchio (1995), 'The differential effects of software training previews on training outcomes', *Journal of Management* **21**(4), 757–787.

Williams, L. and A. Cockburn (2003), 'Agile software development: it's about feedback and change', *Computer* **36**(6), 39 –43.

Winkler, Robert L (1989), 'Combining forecasts: A philosophical basis and some current issues', *International Journal of Forecasting* **5**(4), 605–609.

Yang, Da, Qing Wang, Mingshu Li, Ye Yang, Kai Ye and Jing Du (2008), A survey on software cost estimation in the chinese software industry, *in* 'Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement', ACM, pp. 253–262.

Appendix A Survey Design

A COMPARATIVE STUDY OF THE SOFTWARE ESTIMATION PRACTICE BETWEEN NORWAY AND SPAIN

The purpose of this study is to analyze the current state of the software estimation practice in Norway and Spain, and try to find out if there are any differences or similarities between how it is done in these two countries.

It will be a part of my Master's Thesis in Information Science which I am taking at the University of Bergen(Norway).

All the questions should be answered according to the MOST RECENT PROJECT in which you have participated. It can be the last finished project, but also the current project if you are working on your first project or you are working on a long project for more than 4 months.

Please, feel free to choose the language you prefer from the dropdown menu on top of the page in order to carry out the questionnaire.

The questionnaire contains seven sections: (1) personal infromation and background, (2) basic project information, (3) about the customer, (4) software development methods, (5) estimation methods, (6) the role of the participants, and (7) the closing. The approximate time of completion is 15 minutes.

Please only answer this survey once.

There are 41 questions in this survey

Personal information and background

In this section you will be asked for personal information and background.

1 []In which country do you work? *

Please choose **only one** of the following:

- ○ Norway
- ○ Spain

2 []What is your age?

Please choose **only one** of the following:

- ○ Under 26 years old
- ○ 26-35 years old
- ○ 36-45 years old
- ○ 46-55 years old
- ○ 56 years or older

3 []Gender

Please choose **only one** of the following:

- ○ Female
- ○ Male

4 []Do you have a degree in higher education?

Please choose **only one** of the following:

- ○ Yes
- ○ No

5 []Specify the degree.

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '4 [Q0004]' (Do you have a degree in higher education?)

Please choose **only one** of the following:

- ○ Bachelor degree (3 years)
- ○ Master degree (5 years)
- ○ PhD

6 []How many years experience do you have in the IT field?

Please choose **only one** of the following:

- ○ 0-3
- ○ 4-10
- ○ 11-20
- ○ >20

7 []Which is your current job title?

Please choose **only one** of the following:

- ○ Software developer (all programmers, testers, database engineers, analysts, user interaction designers,...)
- ○ Software architect
- ○ Team Manager
- ○ Customer
- ○ Other [ ]

Basic Project Information

The following questions are related to the most recent or current project you have worked on.

8 []Is the project you have in mind when answering this survey a:

Please choose **only one** of the following:

- ○ Current project
- ○ Already finished project

9 []How long was the duration of the project? In case we are talking about the current project, how long do you think this is going to take?

Please choose **only one** of the following:

- ○ 1-3 months
- ○ 4-6 months
- ○ 7-12 months
- ○ 1-2 years
- ○ >2 years

10 []Was the project internal or external?

Please choose **only one** of the following:

- ○ Internal project
- ○ External project
- ○ Other [ ]

11 []Was there any client or product owner involved in the project? *

Please choose **only one** of the following:

- ○ Yes
- ○ No

12 []How many teams were involved in the software project?

Please choose **only one** of the following:

- ○ 1
- ○ 2-3
- ○ >3

13 []Specify how many of the following were involved in the project.

Please write your answer(s) here:

Developers	
Managers	
Customer/Product owner	
Other	

14 []Was there made any software estimations before or during the project? *

Please choose **only one** of the following:

- ○ Yes, only before the development process
- ○ Yes, only during the development process
- ○ Yes, both before and during the development process
- ○ No, there was not made any estimation before or during the project
- ○ Other (Specify)

Make a comment on your choice here:

15 []Which contract type was set up for the project?

Please choose **only one** of the following:

- ○ Per hour
- ○ Fixed price
- ○ Target price (risk sharing between contractor and customer)
- ○ I do not know
- ○ Other

16 []Which company do you work for? In case you are a customer, write the name of the company that worked for you.

Please write your answer here:

About the customer

17 []Rate the customer on the following properties.

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '11 [Q0011]' (Was there any client or product owner involved in the project?)

Please choose the appropriate response for each item:

	Very poor	Poor	Average	Good	Very good
Collaboration skills	○	○	○	○	○
IT-competence	○	○	○	○	○
Decision making ability	○	○	○	○	○
Clarity of project goals	○	○	○	○	○

18 []If any, what has been the communication frequency with the customer/product owner?

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '11 [Q0011]' (Was there any client or product owner involved in the project?)

Please choose **only one** of the following:

- ○ Daily
- ○ Weekly
- ○ Monthly
- ○ None
- ○ Other [ ]

Software development methods

19 []Which of the following software development methods do you have experience with?

Please choose **all** that apply:

- ☐ Waterfall model
- ☐ Spiral model
- ☐ Agile development methods
- ☐ Other: _____

20 []Which of the following agile methods do you have experience with?

Only answer this question if the following conditions are met:
Answer was at question '19 [Q0019]' (Which of the following software development methods do you have experience with?)

Please choose **all** that apply:

- ☐ Extreme Programming (XP)
- ☐ Kanban
- ☐ Scrum
- ☐ Other: _____

21 []Which software development method have you used in the most recent project?

Please choose **all** that apply:

- ☐ Waterfall model
- ☐ Spiral model
- ☐ Agile development methods
- ☐ Other: _____

22 []Which of the following agile method have you used in the most recent project?

Only answer this question if the following conditions are met:
Answer was at question '21 [Q0021]' (Which software development method have you used in the most recent project?)

Please choose **all** that apply:

- ☐ Extreme Programming (XP)
- ☐ Kanban
- ☐ Scrum
- ☐ Other: _____

Estimation methods

23 []Which of the following estimation techniques have you used in the most recent project? If you have used a combination of methods, please select all that apply.

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please choose **all** that apply:

- ☐ Expert judgment (consider as experts all individuals with competence in estimating software development effort, including software developers and students with previous experience)
- ☐ Analogy (comparing older projects)
- ☐ Algorithmic models (COCOMO,...)
- ☐ Neural networks
- ☐ Bayesian analysis
- ☐ Function Point Analysis
- ☐ Story Point estimation
- ☐ Other: _____

24 []Why did your team choose that/those estimation technique(s) over others?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please write your answer here:

25 []Which of the following units of estimation did you use when estimating the most recent project?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please choose **all** that apply:

- ☐ Story Points
- ☐ Ideal days
- ☐ Time (hours, days, weeks, months,...)
- ☐ Lines of Code
- ☐ Other: [ ]

26 []Which of the following techniques did you use for helping the estimation across teams?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?) *and* Answer was NOT '1' at question '12 [Q0012]' (How many teams were involved in the software project?)

Please choose **all** that apply:

- ☐ First carrying out a joint planning poker session with members from all the teams in order to establish a common baseline for story points
- ☐ Identifying stories in the backlog that are representative of each story point size and making those stories known to all of the teams
- ☐ Normalization of Story Points (SAFe's approach,...)
- ☐ Other: [ ]

27 []What level of experience do/does the team(s) have in estimating user stories?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?) *and* Answer was at question '21 [Q0021]' (Which software development method have you used in the most recent project?)

Please choose **only one** of the following:

- ○ Very high
- ○ High
- ○ Average
- ○ Low
- ○ Very low

28 []Which of the following agile estimation techniques did you use during the most recent project?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?) *and* Answer was at question '21 [Q0021]' (Which software development method have you used in the most recent project?)

Please choose **all** that apply:

- ☐ Planning poker
- ☐ Wall estimation
- ☐ Other: _____

29 []Which estimation scale did you use during the most recent project for estimating user stories? Specify the story point sizes.

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?) *and* Answer was at question '25 [Q0025]' (Which of the following units of estimation did you use when estimating the most recent project?)

Please choose all that apply and provide a comment:

☐ Numeric scales (1 to 10) or (1, 2, 4, 8)	
☐ Fibonacci sequence (1, 2, 3, 5, 8, 13, 21, 34,...)	
☐ T-shirt sizes (XS, S, M, L, XL, XXL, XXXL)	

30 []How successful were your estimations in meeting their goal?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please choose **only one** of the following:

- ○ Very successful
- ○ Successful
- ○ Neither successful or unsuccessful
- ○ Unsuccessful
- ○ Very unsuccessful

31 []Overall, how satisfied are you with the estimation methods you have used in the most recent project?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please choose **only one** of the following:

- ○ Very Satisfied
- ○ Satisfied
- ○ Neither Dissatisfied or Satisfied
- ○ Dissatisfied
- ○ Very Dissatisfied

32 []If not satisfied, what would you change? Do you have any suggestions?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please write your answer here:

The role of the participants

33 []Who took part in the estimation process?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please choose **all** that apply:

- ☐ Developers
- ☐ Software architects
- ☐ Team Managers
- ☐ Customer
- ☐ Other: _____

34 []Were the developers bound to follow estimations made by others in a higher position?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please choose **only one** of the following:

- ○ Yes, the project's up front estimation was already done.
- ○ No, the developers were not bound to estimations made by others.
- ○ Other _____

35 []To which extent does the developers decision influence the final estimations?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please choose **only one** of the following:

- ○ Very high
- ○ High
- ○ Neutral
- ○ Low
- ○ Very low

36 []To which extent does the managers decision influence the final estimations?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please choose **only one** of the following:

- ○ Very high
- ○ High
- ○ Neutral
- ○ Low
- ○ Very low

37 []To which extent does the customers decision influence the final estimations?

Only answer this question if the following conditions are met:
Answer was NOT 'Other (Specify)' at question '14 [Q0014]' (Was there made any software estimations before or during the project?)

Please choose **only one** of the following:

- ○ Very high
- ○ High
- ○ Neutral
- ○ Low
- ○ Very low

Closing

38 []If there is any other information about the project you would like to add, please write it here:

Please write your answer here:

39 []If you would like to give further information that could be relevant to the study, I would appreciate your feedback.

Please write your answer here:

40 []You can also upload files with extra information.

Kindly attach the aforementioned documents along with the survey

41 []If you are willing to do an interview please write your email address.

Please write your answer here:

Thank you very much for completing this questionnaire!

You will find the results of this study in my Master's Thesis. The delivery date of the Thesis is the 1st of June of 2014.

Please, feel free to contact me at any time if you have any questions or suggestions. Email: psa017@student.uib.no

It would be great if you could share the questionnaire with your contacts.
Link: https://bigfoot.uib.no/limesurvey/index.php/821936/lang-en

29/12/2013 – 19:35

Submit your survey.
Thank you for completing this survey.

EN KOMPARATIV STUDIE MELLOM NORGE OG SPANIA OM SOFTWARE-ESTIMERING

Formålet med denne studien er å analysere den nåværende tilstanden til software-estimeringspraksisen i Norge og Spania, og prøve å finne ut av om det er noen likeheter eller forskjeller i hvordan praksisen er i disse to landene.

Studien vil være en del av min Masteroppgave i Informasjonsvitenskap ved Universitetet i Bergen, Norge.

Alle spørsmålene bør besvares med tanke på det SISTE PROSJEKTET du har deltatt i. Det kan være et allerede fullført prosjekt, men også et pågående arbeid dersom du har jobbet med det pågående prosjektet i mer enn 4 måneder.

Når du svarer på spørreskjemaet står du fri til å velge ditt foretrukne språk fra rullgardinmenyen på toppen av siden.

Spørreskjemaet består av syv seksjoner: (1) bakgrunn og personlig informasjon, (2) grunnleggende prosjektinformasjon, (3) om kunden, (4) software-utviklingsmetoder, (5) estimeringsmetoder, (6) rollen til deltakerne, (7) avrunding. Omtrentlig gjennomføringstid er 15 minutter.

Vennligst bare besvar dette spørreskjemaet en gang.

Det er 41 spørsmål i denne undersøkelsen.

Personlig informasjon og bakgrunn

I denne delen vil du bli spurt om bakgrunn og personlig informasjon.

1 []I hvilket land jobber du? *

Velg kun en av følgende:

- ○ Norge
- ○ Spania

2 []Hva er din alder?

Velg kun en av følgende:

- ○ Under 26 år
- ○ 26-35 år
- ○ 36-45 år
- ○ 46-55 år
- ○ 56 år eller eldre

3 []Kjønn

Velg kun en av følgende:

- ○ Kvinne
- ○ Mann

4 []Har du en grad i høyere utdanning?

Velg kun en av følgende:

- ○ Ja
- ○ Nei

5 []Spesifiser graden

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var 'Ja' ved spørsmål '4 [Q0004]' (Har du en grad i høyere utdanning?)

Velg kun en av følgende:

- ○ Bachelorgrad (3 år)
- ○ Mastergrad (5 år)
- ○ Doktorgrad

6 []Hvor mange års erfaring har du innen IT-området?

Velg kun en av følgende:

- ○ 0-3
- ○ 4-10
- ○ 11-20
- ○ >20

7 []Hva er din nåværende jobbtittel?

Velg kun en av følgende:

- ○ Software-utvikler (alle programmerere, testere, databaseingeniører, analytikere, brukerinteraksjondesignere,...)
- ○ Software-arkitekt
- ○ Teamleder
- ○ Kunde
- ○ Annet [ ]

Grunnleggende prosjektinformasjon

8 []Er prosjektet du har valgt å bruke i denne undersøkelsen et:

Velg kun en av følgende:

- ○ Nåværende/pågående prosjekt
- ○ Avsluttet prosjekt

9 []Hvor lenge varte prosjektet? Dersom vi snakker om et pågående prosjekt, hvor lenge antar du at det vil vare?

Velg kun en av følgende:

- ○ 1-3 måneder
- ○ 4-6 måneder
- ○ 7-12 måneder
- ○ 1-2 år
- ○ >2 år

10 []Var prosjektet internt eller eksternt?

Velg kun en av følgende:

- ○ Internt prosjekt
- ○ Eksternt prosjekt
- ○ Annet [ ]

11 []Var det involvert klienter eller produkteiere i prosjektet? *

Velg kun en av følgende:

- ○ Ja
- ○ Nei

12 []Hvor mange team var involvert i prosjektet?

Velg kun en av følgende:

- ○ 1
- ○ 2-3
- ○ >3

13 []Spesifiser hvor mange av de følgende som var involvert i prosjeketet:

Vennligst skriv her:

Utviklere	
Ledere	
Kunde/ Product owner	
Andre	

14 []Ble det gjort software-estimering før eller under prosjektet? *

Velg kun en av følgende:

- ○ Ja, kun før utviklingsprosessen
- ○ Ja, kun under utviklingsprosessen
- ○ Ja, både før og under utviklingsprosessen
- ○ Nei, det ble ikke utført estimeringer før eller under prosjektet
- ○ Annet (spesifiser)

Legg til en kommentar til svaret her:

15 []Hva slags type kontrakt ble laget for prosjektet?

Velg kun en av følgende:

- ○ Per time
- ○ Fastpris
- ○ Target pris (risikovurdering mellom leverandør og kunde)
- ○ Vet ikke
- ○ Annet

16 []Hvilket firma jobber du for? Dersom du er kunde, skriv navnet på firmaet som utførte din bestilling.

Vennligst skriv her:

Om kunden

17 []Ranger kunden ut ifra følgende egenskaper

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var 'Ja' ved spørsmål '11 [Q0011]' (Var det involvert klienter eller produkteiere i prosjektet?)

Vennligst velg passende svar til hvert alternativ:

	Veldig dårlig	Dårlig	Gjennomsnittlig	God	Veldig god
Samarbeidsevner	○	○	○	○	○
IT-kompetanse	○	○	○	○	○
Beslutningstaking	○	○	○	○	○
Klarhet i prosjektmål	○	○	○	○	○

18 []Hva har vært kommunikasjonsfrekvensen mellom kunden og produkteieren?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var 'Ja' ved spørsmål '11 [Q0011]' (Var det involvert klienter eller produkteiere i prosjektet?)

Velg kun en av følgende:

- ○ Daglig
- ○ Ukentlig
- ○ Månedlig
- ○ Ingen
- ○ Annet [ ]

Software-utviklingsmetoder

19 []Hvilke(n) av følgende software-utviklingsmetoder har du erfaring med?

Vennligst velg alle som passer:

- ☐ Fossefallsmodellen
- ☐ Spiralmodellen
- ☐ Smidige utviklingsmetoder
- ☐ Annet: [ ]

20 []Hvilke(n) av følgende smidige metoder (agile methods) har du erfaring med?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var ved spørsmål '19 [Q0019]' (Hvilke(n) av følgende software-utviklingsmetoder har du erfaring med?)

Vennligst velg alle som passer:

- ☐ Extreme Programming (XP)
- ☐ Kanban
- ☐ Scrum
- ☐ Annet: [ ]

21 []Hvilken software-utviklingsmetode ble brukt i ditt siste prosjekt?

Vennligst velg alle som passer:

- ☐ Fossefallsmodellen
- ☐ Spiralmodellen
- ☐ Smidige utviklingsmetoder
- ☐ Annet: [ ]

22 []Hvilken smidige metode (agile method) ble brukt i ditt siste prosjekt?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var ved spørsmål '21 [Q0021]' (Hvilken software-utviklingsmetode ble brukt i ditt siste prosjekt?)

Vennligst velg alle som passer:

- ☐ Extreme Programming (XP)
- ☐ Kanban
- ☐ Scrum
- ☐ Annet: [ ]

Estimeringsmetoder

23 []Hvilke(n) av følgende estimeringsteknikker ble brukt i ditt siste prosjekt? Dersom du har benyttet en kombinasjon, vennligst velg alle de relevante.

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Vennligst velg alle som passer:

- ☐ Ekspertvurderinger (eksperter er individer med kompetanse i estimering av software, inkludert programvareutviklere og studenter med tidligere erfaring)
- ☐ Analogi (sammenligner eldre prosjekter)
- ☐ Algoritmemodeller (COCOMO,...)
- ☐ Neurale nettverk
- ☐ Bayesisk analyse
- ☐ Function Point analyse
- ☐ Story Point-estimering
- ☐ Annet: _____

24 []Hvorfor valgte ditt team akkurat denne/disse estimeringsteknikken(e)?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Vennligst skriv her:

25 []Hvilke av følgende estimeringsenheter ble brukt under estimeringen av ditt siste prosjekt?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Vennligst velg alle som passer:

- ☐ Story Points
- ☐ Ideelle dager
- ☐ Tid (timer, dager, uker, måneder,...)
- ☐ Kodelinjer
- ☐ Annet: [ ]

26 []Hvilke(n) av følgende teknikker ble brukt for å hjelpe estimeringen mellom teamene?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?) *og* Svaret var IKKE '1' ved spørsmål '12 [Q0012]' (Hvor mange team var involvert i prosjektet?)

Vennligst velg alle som passer:

- ☐ Først utføre en samlet "planning poker"-økt med medlemmer fra alle teamene, for å etablere en felles baseline for story points
- ☐ Identifisere historier i backlog som er representative for hver story point-størrelse, og fortelle disse historiene til alle teamene
- ☐ Normalisasjon av Story Points (SAFe's approach,...)
- ☐ Annet: [ ]

27 []Hvor mye erfaring har teamet/ teamene i å estimere brukerhistorier?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?) *og* Svaret var ved spørsmål '21 [Q0021]' (Hvilken software-utviklingsmetode ble brukt i ditt siste prosjekt?)

Velg kun en av følgende:

- ○ Veldig mye
- ○ Mye
- ○ Gjennomsnittlig
- ○ Lite
- ○ Veldig lite

28 []Hvilke(n) av følgende smidige teknikker ble brukt i ditt siste prosjekt?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?) *og* Svaret var ved spørsmål '21 [Q0021]' (Hvilken software-utviklingsmetode ble brukt i ditt siste prosjekt?)

Vennligst velg alle som passer:

- ☐ Planning poker
- ☐ Wall estimation
- ☐ Annet: _____

29 []Hvilken estimeringsskala ble brukt for å estimere brukerhistorier i ditt siste prosjekt? Spesifiser "story point"-størrelse.

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?) *og* Svaret var ved spørsmål '25 [Q0025]' (Hvilke av følgende estimeringsenheter ble brukt under estimeringen av ditt siste prosjekt?)

Vennligst velg de alternativene som passer, og legg til en kommentar:

☐ Numeriske skalaer (1 to 10) eller (1, 2, 4, 8)	
☐ Fibonacci-sekvens (1, 2, 3, 5, 8, 13, 21, 34,...)	
☐ T-skjortestørrelse (XS, S, M, L, XL, XXL, XXXL)	

30 []Hvor vellykket var estimeringen i å møte målene?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Velg kun en av følgende:

- ○ Veldig vellykket
- ○ Vellykket
- ○ Hverken vellykket eller mislykket
- ○ Mislykket
- ○ Veldig mislykket

31 []Hvor fornøyd er du totaltsett med de valgte estimeringsmetodene i ditt siste prosjekt?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Velg kun en av følgende:

- ○ Veldig fornøyd
- ○ Fornøyd
- ○ Hverken fornøyd eller misfornøyd
- ○ Misfornøyd
- ○ Veldig misfornøyd

32 []Hvis du ikke er fornøyd, hva ville du endret? Har du noen forslag?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Vennligst skriv her:

Rollen til deltakerne

33 []Hvem deltok i estimeringsprosessen?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Vennligst velg alle som passer:

- ☐ Utvilkere
- ☐ Software-arkitekter
- ☐ Teamledere
- ☐ Kunde
- ☐ Annet: _____

34 []Var utviklerne bundet til å følge estimeringer gjort av noen i en høyere stilling?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Velg kun en av følgende:

- ○ Ja, prosjektets estimeringer var gjort på forhånd.
- ○ Nei, utviklerne måtte følge estimeringer laget av andre.
- ○ Annet _____

35 []I hvilken grad påvirker utviklernes valg de endelige estimeringene?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Velg kun en av følgende:

- ○ I veldig høy grad
- ○ I høy grad
- ○ Nøytral
- ○ I lav grad
- ○ I veldig lav grad

36 []I hvilken grad påvriker lederen de endelige estimeringene?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Velg kun en av følgende:

- ○ I veldig høy grad
- ○ I høy grad
- ○ Nøytral
- ○ I lav grad
- ○ I veldig lav grad

37 []I hvilken grad påvriker kunden de endelige estimeringene?

Svar kun på dette hvis følgende betingelser er oppfylt:
Svaret var IKKE 'Annet (spesifiser)' ved spørsmål '14 [Q0014]' (Ble det gjort software-estimering før eller under prosjektet?)

Velg kun en av følgende:

- ○ I veldig høy grad
- ○ I høy grad
- ○ Nøytral
- ○ I lav grad
- ○ I veldig lav grad

Avrunding

38 []Hvis det er annen informasjon om prosjektet du ønsker å tilføye, vennligst skriv det her:

Vennligst skriv her:

39 []Dersom du ønsker å tilføye annen informasjon som kan være relevant for studien, vil jeg sette pris på din tilbakemelding.

Vennligst skriv her:

40 []Her kan du laste opp filer med tilleggsinformasjon.

Kindly attach the aforementioned documents along with the survey

41 []Hvis du er villig til å delta i et intervju, vennligst skriv din e-postadresse her.

Vennligst skriv her:

Tusen takk for at du fullførte denne undersøkelsen!

Du kan finne resultatene av denne studien i min Mastergrad. Leveringsdato for oppgaven er 1. juni 2014.

Ta gjerne kontakt dersom du har spørsmål eller forslag til min oppgave. E-postadresse: psa017@student.uib.no

Jeg hadde satt pris på om du kan dele dette spørreskjemaet med dine kontakter.
Link: https://bigfoot.uib.no/limesurvey/index.php/821936/lang-nb

29.12.2013 – 19:35

Send undersøkelse.
Takk for at du fullførte denne undersøkelsen.

ESTUDIO COMPARATIVO DEL PROCESO DE ESTIMACION DEL SOFTWARE ENTRE NORUEGA Y ESPAÑA

La finalidad de este estudio es analizar el estado actual de las prácticas de estimación del software en Noruega y España, y tratar de encontrar/averiguar si existen diferencias o similitudes entre ambos países.

Este estudio será parte de mi Tesis de Maestría en Ciencias de la Información, que estoy cursando en la Universidad de Bergen (Noruega).

Todas las preguntas deben ser contestadas de acuerdo con el PROYECTO MÁS RECIENTE en el que haya participado. Puede ser el último proyecto acabado, o el proyecto actual si está trabajando en su primer proyecto o en un proyecto de más de 4 meses de duración.

Por favor, completa el cuestionario solo una vez.

Por favor, eliga el idioma que desee del menú desplegable en la parte superior de la página para llevar a cabo la encuesta.

El cuestionario contiene siete secciones: (1) infromación personal, (2) información básica del proyecto, (3) acerca del cliente, (4) métodos de desarrollo de software, (5) métodos de estimación, (6) el rol de los participantes, y (7) finalizando. El tiempo aproximado de finalización es de 15 minutos.

Hay 41 preguntas en esta encuesta

Información personal

En esta sección se le pedirá información personal

1 []¿En que país trabajas? *

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Noruega
- ○ España

2 []Edad

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Menos de 26 años
- ○ 26-35 años
- ○ 36-45 años
- ○ 46-55 años
- ○ 56 años o más

3 []Sexo

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Femenino
- ○ Masculino

4 []¿Tiene un título de educación superior?

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Sí
- ○ No

5 []Especifica el tipo de título

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue 'Sí' en la pregunta '4 [Q0004]' (¿Tiene un título de educación superior?)

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Diplomatura, Ingeniería Técnica o Grado (3-4 años)
- ○ Licenciatura, Ingeniería Superior o Master (5 años)
- ○ Doctorado (PhD)

6 []¿Cuántos años de experiencia tiene en el campo de las Tecnologías de Información(Informática)?

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ 0-3
- ○ 4-10
- ○ 11-20
- ○ >20

7 []Puesto de trabajo actual

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Desarrollador de software (Programador, tester, ingeniero de bases de datos, diseñador,...)
- ○ Arquitecto de software
- ○ Jefe de grupo/equipo (Team manager)
- ○ Cliente/Product owner
- ○ Otro [ ]

Información básica del proyecto

Las siguientes preguntas están relacionadas con el proyecto más reciente en el que haya trabajado.

8 []¿Es el proyecto elegido el proyecto en el que estás trabajando actualmente o un proyecto ya terminado?

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Proyecto actual
- ○ Proyecto finalizado

9 []¿Cuál fue/será la duración del proyecto?

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ 1-3 meses
- ○ 4-6 meses
- ○ 7-12 meses
- ○ 1-2 años
- ○ >2 años

10 []¿Fue/Es un proyecto interno o externo?

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Proyecto interno
- ○ Proyecto externo
- ○ Otro [ ]

11 []¿Participó/Participa algún cliente o "Product Owner" en el proyecto? *

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Sí
- ○ No

12 []¿Cuántos equipos/grupos participaron/participan en el proyecto?

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ 1
- ○ 2-3
- ○ >3

13 []Especifique cuántos de los siguientes estuvieron/están involucrados en el proyecto.

Por favor, escriba su(s) respuesta(s) aquí:

Desarrolladores	
Jefes de equipo (Team managers)	
Cliente/Product Owner	
Otros	

14 []¿Se ha estimado el software antes o durante el proyecto? *

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Sí, solo antes del proceso de desarrollo
- ○ Sí, solo durante el proceso de desarrollo
- ○ Sí, tanto antes como durante el proceso de desarrollo
- ○ No, no se hizo ninguna estimación antes ni durante el proyecto
- ○ Otro (Especifica)

Comente su elección aquí:

15 []¿Qué tipo de contrato se estableció para el proyecto?

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Por horas
- ○ Precio fijado
- ○ Precio basado en el objetivo (distribución de riesgos entre el empresario y el cliente)
- ○ No lo se
- ○ Otro

16 []¿Para qué empresa trabajas? En el caso de ser un cliente, escriba el nombre de la empresa que trabajó para usted.

Por favor, escriba su respuesta aquí:

Acerca del cliente

17 []Califica al cliente en las siguientes propiedades.

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue 'Sí' en la pregunta '11 [Q0011]' (¿Participó/Participa algún cliente o "Product Owner" en el proyecto?)

Por favor, seleccione la respuesta apropiada para cada concepto:

	Muy mala	Mala	Aceptable	Buena	Muy buena
Habilidad de colaboración	○	○	○	○	○
Competencia en Informática	○	○	○	○	○
Capacidad a la hora de tomar decisiones	○	○	○	○	○
Claridad de los objetivos del proyecto	○	○	○	○	○

18 []En el caso de que la hubiera, ¿cuál ha sido/es la frecuencia de comunicación con el cliente/product owner?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue 'Sí' en la pregunta '11 [Q0011]' (¿Participó/Participa algún cliente o "Product Owner" en el proyecto?)

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Diariamente
- ○ Semanalmente
- ○ Mensualmente
- ○ Ninguna
- ○ Otro [ ]

Métodos de desarrollo de software

19 []¿Con cuál de los siguientes métodos de desarrollo de software tiene experiencia previa?

Por favor, marque las opciones que correspondan:

- ☐ Modelo en cascada (Waterfall)
- ☐ Desarrollo en espiral (Spiral model)
- ☐ Métodos de desarrollo ágiles
- ☐ Otro: [ ]

20 []¿Con cuál de los siguientes métodos ágiles tiene experiencia previa?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue en la pregunta '19 [Q0019]' (¿Con cuál de los siguientes métodos de desarrollo de software tiene experiencia previa?)

Por favor, marque las opciones que correspondan:

- ☐ Extreme Programming (XP)
- ☐ Kanban
- ☐ Scrum
- ☐ Otro: [ ]

21 []¿Qué métodos de desarrollo de software ha utilizado en el proyecto más reciente?

Por favor, marque las opciones que correspondan:

- ☐ Modelo en cascada (Waterfall)
- ☐ Desarrollo en espiral (Spiral model)
- ☐ Métodos de desarrollo ágiles
- ☐ Otro: [ ]

22 []¿Cuál de los siguientes métodos ágiles ha utilizado en el proyecto más reciente?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue en la pregunta '21 [Q0021]' (¿Qué métodos de desarrollo de software ha utilizado en el proyecto más reciente?)

Por favor, marque las opciones que correspondan:

- ☐ Extreme Programming (XP)
- ☐ Kanban
- ☐ Scrum
- ☐ Otro: [ ]

Métodos de estimación

23 []¿Cuáles de las siguientes técnicas de estimación ha utilizado en el proyecto más reciente? Si ha utilizado una combinación de métodos, por favor selecciona los que correspondan.

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor, marque las opciones que correspondan:

- ☐ Opinión de expertos (considere como expertos a todas las personas con competencia en la estimación de software, incluidos los desarrolladores de software y estudiantes con experiencia previa)
- ☐ Analogía (comparando proyectos anteriores)
- ☐ Modelos algorítmicos (COCOMO,...)
- ☐ Redes neuronales
- ☐ Análisis bayesiano
- ☐ Análisis de puntos de función (Function Point Analysis)
- ☐ Estimación de puntos de historia (Story point estimation)
- ☐ Otro: _____

24 []¿Cuál fue/es la razón por la cual su equipo escogió/ha escogido este/estos metodo(s) antes que otro(s)?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor, escriba su respuesta aquí:

25 []¿Cuáles de las siguientes unidades de estimación se utilizaron al estimar el proyecto más reciente?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor, marque las opciones que correspondan:

- ☐ Puntos de historia (Story points)
- ☐ Días ideales (Ideal days)
- ☐ Tiempo (horas, días, semanas, meses,...)
- ☐ Líneas de código (LOC)
- ☐ Otro: [ ]

26 []¿Cuál de las siguientes técnicas habéis utilizado para gestionar la estimación entre diferentes equipos?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?) *Y* La respuesta fue NO '1' en la pregunta '12 [Q0012]' (¿Cuántos equipos/grupos participaron/participan en el proyecto?)

Por favor, marque las opciones que correspondan:

- ☐ Organizar una sesión de "planning poker" conjunta con miembros de todos los grupos con el fin de establecer una base común para puntos de la historia
- ☐ Identificar "user stories" en el backlog que representen cada uno de los tamaños de los "story points", y ponerlos a disposición de todos los participantes en el proyecto
- ☐ Normalización de los "story points" (por ejemplo, el planteamiento de SAFe)
- ☐ Otro: [ ]

27 []Nivel de experiencia del equipo estimando historias de usuario(User Stories).

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?) *Y* La respuesta fue en la pregunta '21 [Q0021]' (¿Qué métodos de desarrollo de software ha utilizado en el proyecto más reciente?)

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Muy alto
- ○ Alto
- ○ Medio
- ○ Bajo
- ○ Muy bajo

28 []¿Cuál de las siguientes técnicas de estimación ágiles se utilizaron durante el proyecto más reciente?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?) *Y* La respuesta fue en la pregunta '21 [Q0021]' (¿Qué métodos de desarrollo de software ha utilizado en el proyecto más reciente?)

Por favor, marque las opciones que correspondan:

- ☐ Planning poker
- ☐ Wall estimation
- ☐ Otro: _____

29 []¿Qué escala de estimación su utilizó durante el más reciente proyecto al estimar las historias de usuario(User Stories)? Especifique el tamaño de los "story points".

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?) *Y* La respuesta fue en la pregunta '25 [Q0025]' (¿Cuáles de las siguientes unidades de estimación se utilizaron al estimar el proyecto más reciente?)

Por favor, seleccione todas las opciones que correspondan y escriba un comentario:

☐ Escalas numéricas. Por ejemplo, (1 a 10) o (1,2,4,8,...)	
☐ Secuencia Fibonacci (1, 2, 3, 5, 8, 13, 21, 34,...)	
☐ Tallas de camiseta (XS, S, M, L, XL, XXL, XXXL)	

30 []Evalúa el grado de éxito de las estimaciones del proyecto.

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Muy exitosa
- ○ Exitosa
- ○ Mediocre
- ○ Fracaso
- ○ Fracaso total

31 []En general, ¿cómo de satisfecho está usted con los métodos de estimación que se han utilizado en el proyecto más reciente?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Muy satisfecho
- ○ Satisfecho
- ○ Ni satisfecho ni insatisfecho
- ○ Insatisfecho
- ○ Muy insatisfecho

32 []Si no está satisfecho, ¿qué cambiaría? ¿Tiene alguna sugerencia?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor, escriba su respuesta aquí:

El rol de los participantes

33 []¿Quién participó en el proceso de estimación?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor, marque las opciones que correspondan:

- ☐ Desarrolladores
- ☐ Arquitectos de software
- ☐ Jefes de equipo (Team managers)
- ☐ Cliente/Product Owner
- ☐ Otro: _____

34 []¿Estaban los desarrolladores obligados a seguir estimaciones realizadas por otros cargos más altos?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Sí, la estimación por adelantado del proyecto ya estaba hecha.
- ○ No, no estaban obligados a seguir estimaciones realizadas por otros.
- ○ Otro _____

35 []¿En qué medida influye la decisión de los desarrolladores/programadores en las estimaciones finales?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Muy alta
- ○ Alta
- ○ Neutral
- ○ Baja
- ○ Muy baja

36 []¿En qué medida influye la decisión de los managers en las estimaciones finales?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Muy alta
- ○ Alta
- ○ Neutral
- ○ Baja
- ○ Muy baja

37 []¿En qué medida influye la decisión del cliente en las estimaciones finales?

Sólo conteste esta pregunta si se cumplen las siguientes condiciones:
La respuesta fue NO 'Otro (Especifica)' en la pregunta '14 [Q0014]' (¿Se ha estimado el software antes o durante el proyecto?)

Por favor seleccione **sólo una** de las siguientes opciones:

- ○ Muy alta
- ○ Alta
- ○ Neutral
- ○ Baja
- ○ Muy baja

Finalizando

38 []Si desea añadir información adicional sobre el proyecto, puede hacerlo a continuación:

Por favor, escriba su respuesta aquí:

39 []Si puede aportar más información relevante al estudio que estoy realizando, le agradecería su colaboración.

Por favor, escriba su respuesta aquí:

40 []Si lo desea, también puede ajuntar archivos.

Kindly attach the aforementioned documents along with the survey

41 []Si está dispuesto a hacer una entrevista para poder analizar mejor el proceso de estimación, por favor escriba su dirección de correo electrónico.

Por favor, escriba su respuesta aquí:

Muchas gracias por completar el cuestionario!

Encontrará los resultados de este estudio en mi Tesis de Maestría. La fecha de entrega de la tesis es el 1 de junio de 2014.

Por favor, no dude en ponerse en contacto conmigo en caso de tener cualquier pregunta o sugerencia. Email: psa017@student.uib.no

Sería genial si pudiera difundir/compartir el cuestionario con sus contactos!
Link: https://bigfoot.uib.no/limesurvey/index.php/821936/lang-es

29/12/2013 – 19:35

Enviar su encuesta.
Gracias por completar esta encuesta.