Università degli Studi di Padova

Department of Mathematics “Tullio Levi-Civita”

Master’s degree in Computer Science

Challenges in customer-supplier relationship management in bespoke software development: a survey

Supervisor
Prof. Tullio Vardanega

Undergraduate
Alessio Trevisan

Academic Year 2022-2023

Alessio Trevisan: *Challenges in customer-supplier relationship management in bespoke software development: a survey,* Master's degree, © September 2023.

What we observe is not nature in itself but nature exposed to our method of questioning.

— Werner Karl Heisenberg

Abstract

Bespoke software development needs the engaged involvement of customers in each phase of the development life cycle, as the client's perspective and contribution are essential to the success of the project. At present, most of the software houses (organisations specialised in building software products) that work on commissioned supply adopt agile methodologies in the intent of achieving customer loyalty and of lowering time to market. Frequently however, the relationship between customers and suppliers may be undermined by a range of latent factors whose detection should need thorough attention and mitigation to keep the project afloat. Such problems as domain language barriers, unclear definition of scope, requirements, milestones and responsibilities, passive and defensive attitude in place of collaborative approaches, and different levels of expertise between teams and customers, may all spawn divergence in scope, loss of quality, and incorrect estimations. This work presents a survey that investigates the main factors of influence on breakages in the relationship between customers and suppliers, addressing the perspectives of the two sides of the fence.

Acknlowledgements

First of all I would like to express my deepest gratitude to all the people and organisations that have contributed to the development of this thesis, from Moku and the organisations that offered their contribution to Prof. Tullio Vardanega, thesis supervisor, who offered a valuable contribution of supervision and assistance throughout the project.

In conclusion, I would like to thank my friends and family, who have always supported and believed in me throughout my university career.

Padova, September 2023 Alessio Trevisan

Contents

List of Figures

List of Tables

Chapter 1

Common issues experienced by bespoke software development companies

This chapter introduces the case study for this thesis, describing all possible topics of research involved and the related state of the art. In addition, all essential background information will be provided to the reader.

1.1 Context overview

When in need of purchasing new software products, customer may opt for off-the-shelf artifacts or may consider the idea of developing a tailor-made solution based on their specific needs. The former solution usually grants a solid product that has been developed in order to respond to a shared necessity of the market, thus is has been tested on a large installed base for years. This benefit is proportional to the number of users, even though the coverage of the user tests is usually not homogeneous. In fact, common flows (e.g. the parts of the applications that are widely used by the majority of the users) tend to be highly more "tested" in comparison with edge cases or additional features.

Furthermore, off-the-shelf products are usually developed within the supplier's company itself, with shorter development cycles and without the need of interacting frequently with lots of stakeholders. These products are capable of granting a specific set of functionalities that are available at a given price.

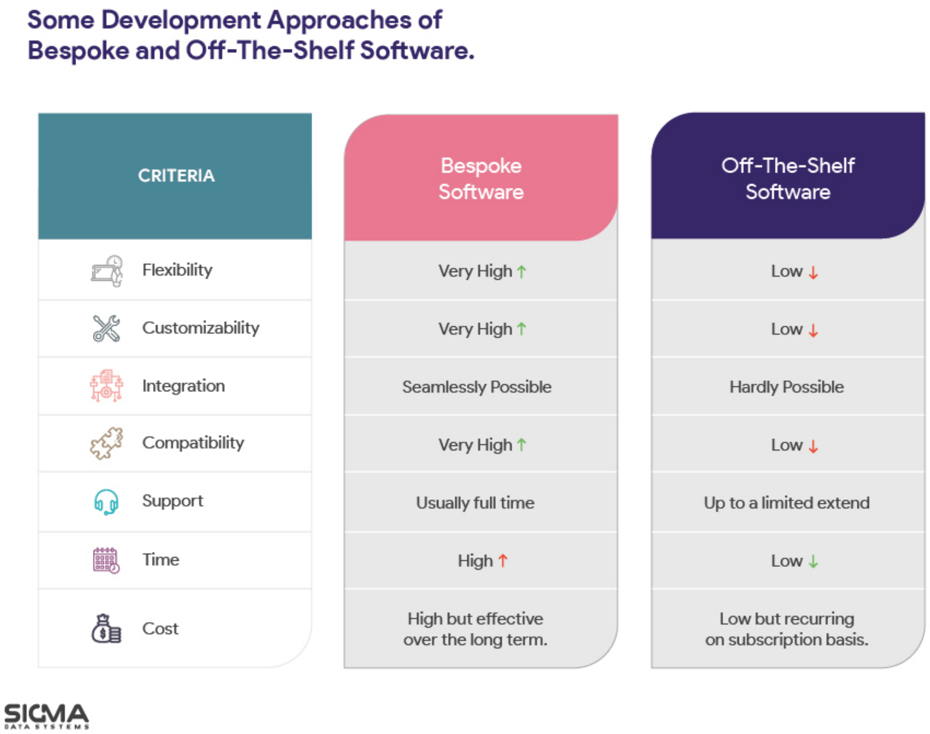
Conversely, for their intrinsic nature, off-the-shelf products present a reduced level of customisation to the needs of specific customers, or simply there might not yet exist products that capture and meet the actual client needs. For example, companies that rely on projects with a high degree of innovation and risk (e.g. startups) or need to integrate/update pre-existing services usually may not find adequate products.

In these cases, therefore, bespoke software development solutions become essential. Unlike the former scenario, where customers act like passive figures that select the most suitable software for them, this approach requires active engagement of the client during the entire development of the product.

In custom-made projects, sometimes the client may already have an internal team of designers and developers who are able to ensure the correct realisation of the product or eventual integrations with other services. However, in far more common scenarios, customers do not have the resources or the expertise that would allow them to undertake the project internally, so they usually have to outsource the development of the project by relying on external providers such as software houses or freelance developers.

This scenario inevitably incurs distinct overheads that may cause well-known problems in software engineering, as it requires the alignment of groups of people with different cultures, expertise and needs towards a common goal.

In 2021, the custom software development market generated revenues of USD 24.46 billion expected to expand at a compound annual growth rate (CAGR) of 22.3% from 2022 to 2030 [11], thus this sector can be considered as a relevant scenario of interest.



Some Development Approaches of Bespoke and Off-The-Shelf Software.

CRITERIA	Bespoke Software	Off-The-Shelf Software
Flexibility	Very High ↑	Low ↓
Customizability	Very High ↑	Low ↓
Integration	Seamlessly Possible	Hardly Possible
Compatibility	Very High ↑	Low ↓
Support	Usually full time	Up to a limited extend
Time	High ↑	Low ↓
Cost	High but effective over the long term.	Low but recurring on subscription basis.

SIGMA DATA SYSTEMS

Figure 1.1: Bespoke vs off-the-shelf software, *Source:* [19]

1.2 Bespoke software development

Bespoke software development, also referred to as custom or tailor-made software, is defined as *the creation of a software product based on the needs of a customer*. Its main characteristic is represented by the need for outsourcing the development of the project, generally entrusted to one or more software houses.

The development of a software project is a complex job. It is reasonable to assume that the complexity increases according to the number of parties involved, as the coordination of different organisations adds overhead.

In order to standardise many relevant aspects of the development of a software project, an international standard for software life cycle processes has been proposed. This standard is called **ISO/IEC:12207.** In this thesis, the reference version will be the latest, 12207:2017[1].

In particular, *this document provides a common process framework for describing the life cycle of systems* created by humans, adopting a Software Engineering approach.

The ISO standard describes the processes needed to establish a collaborative effort that follows a structured development process from the initial concept to the final product and its maintenance.

In particular, it aims to deliver a product that meets the requirements of the various stakeholders involved in the project, considering both their technical and business needs.

The standard describes the life cycle of the system from its conception of ideas to the retirement of a system. Furthermore, it provides the necessary processes to acquire and supply systems, and helps to improve communication and cooperation between the parties involved in the creation, use and management of modern software systems, so that they can work together in a unified and coherent manner.

ISO/IEC:12207 can be used in different ways. This thesis will refer to the acquirer-supplier mode, as guidance in developing agreements between the acquirer and the supplier. The standard defines the processes and activities that can be selected, negotiated, agreed to, and performed as reported in the agreement.

The ISO defines the concepts of acquirer, user, customer, supplier, and process. These terms will be widely referred to in this thesis.

The *organisation executing the acquisition process* is called the **acquirer**. Users, instead, can be the organisation or individuals that directly benefit from the utilisation of the product or service, while customer refers to the user and acquirer collectively. However, in this thesis, the concept of **customer** will be referred to as a *synonym of acquirer.*

Suppliers are instead defined as organisations or individuals that enter into an agreement with the acquirer for the supply of a product or service.

At last, a **process** is defined as a *set of interrelated or interacting activities that transform input into output.* In particular, every process that will be object of interest for this thesis is defined by

- a title
- its purpose
- its outcomes
- its activities
- its tasks

In particular, the document refers to the concept of **software life cycle processes**, which are all the *processes that can be applied to the life cycle of a software*, regardless of the software development life cycle model that is selected for the project.

The ISO classifies software life cycle processes into four main categories. The processes that will be relevant for this case study will be described in the following paragraphs

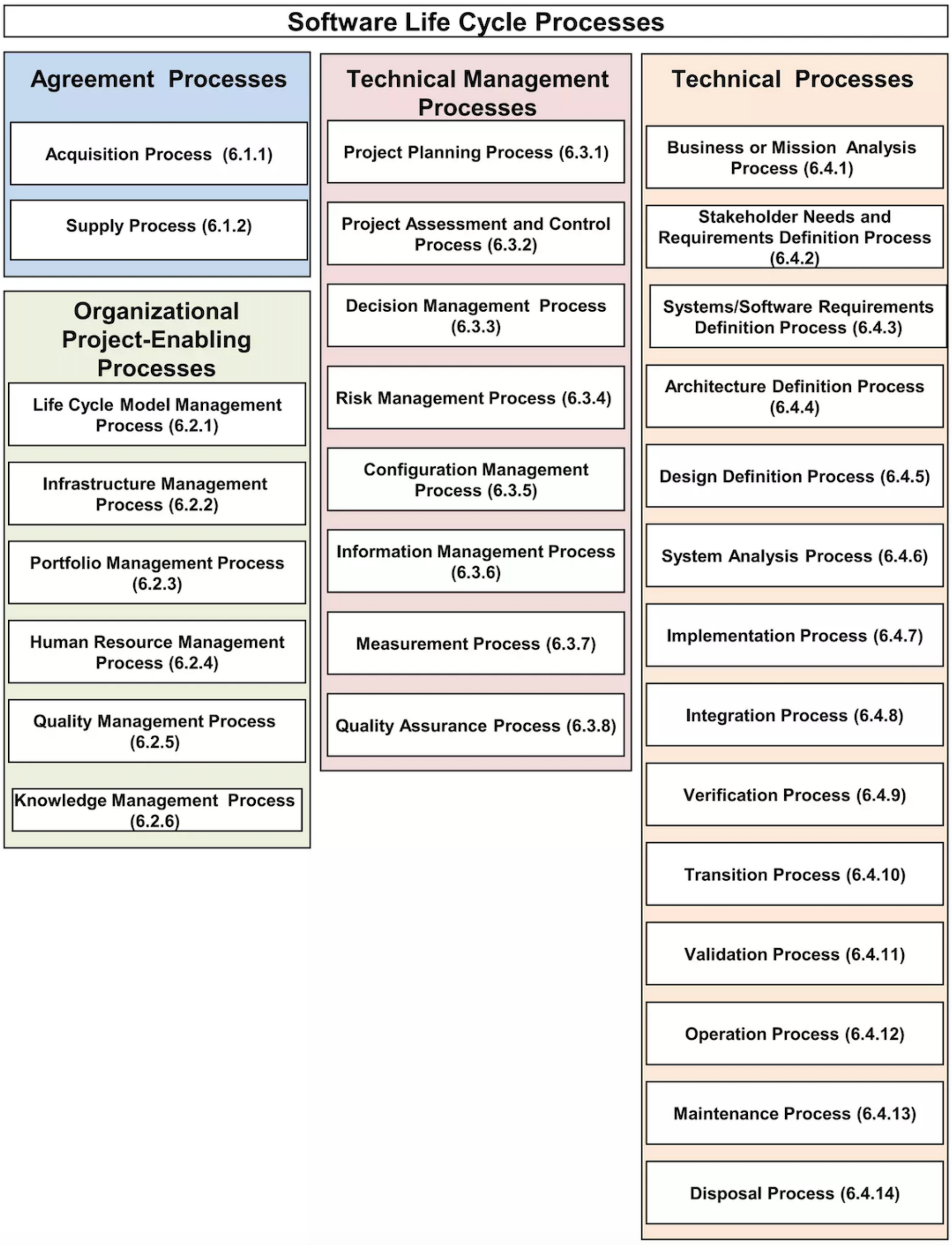

Software Life Cycle Processes
Agreement Processes
Acquisition Process (6.1.1)
Supply Process (6.1.2)
Organizational Project-Enabling Processes
Life Cycle Model Management Process (6.2.1)
Infrastructure Management Process (6.2.2)
Portfolio Management Process (6.2.3)
Human Resource Management Process (6.2.4)
Quality Management Process (6.2.5)
Knowledge Management Process (6.2.6)
Technical Management Processes
Project Planning Process (6.3.1)
Project Assessment and Control Process (6.3.2)
Decision Management Process (6.3.3)
Risk Management Process (6.3.4)
Configuration Management Process (6.3.5)
Information Management Process (6.3.6)
Measurement Process (6.3.7)
Quality Assurance Process (6.3.8)
Technical Processes
Business or Mission Analysis Process (6.4.1)
Stakeholder Needs and Requirements Definition Process (6.4.2)
Systems/Software Requirements Definition Process (6.4.3)
Architecture Definition Process (6.4.4)
Design Definition Process (6.4.5)
System Analysis Process (6.4.6)
Implementation Process (6.4.7)
Integration Process (6.4.8)
Verification Process (6.4.9)
Transition Process (6.4.10)
Validation Process (6.4.11)
Operation Process (6.4.12)
Maintenance Process (6.4.13)
Disposal Process (6.4.14)

Figure 1.2: Categories of processes in ISO/IEC 12207:2017, *Source:* [1]

1.2.1 Agreement processes

The **agreement processes** define the *activities necessary to establish an agreement between two organisations.*
This category includes

- Acquisition process
- Supply process

The acquisition and the supply processes will be discussed a lot in this thesis.
Organisations use the **acquisition process** to *acquire products or services.*
In a bespoke software development project, the acquisition process is generally *carried out by the acquirer.*
The **supply process** is used by organisations to *provide products or services.* In a custom software development project, the supply process is generally *performed by the supplier.*

Name	**Purpose**
`Acquisition process`	Obtain a product or service in accordance with the acquirer's requirements
`Supply process`	Provide an acquirer with a product or service that meets agreed requirements

Table 1.1: Acquisition processes and their purpose

1.2.2 Organisational project-enabling processes

The **organisational project-enabling processes** help *ensure the organisations the capability to acquire and supply products or services* through the initiation, support, and control of projects. This category includes

- Life cycle model management process
- Infrastructure management process
- Portfolio management process
- Human resource management process
- Quality management process
- Knowledge management process

It is important to underline how these processes should be carried on by both parties signing a contract. In fact, both acquirers and suppliers ensure aspects such as quality standards, availability of resources (human, material, time, and economic), and governance on the project.
For the objectives of this thesis, the **life cycle model management** and the **quality management processes** will be cited. In bespoke software development, the former

is usually performed by the supplier, but the characteristics and the will of the acquirer tend to have an impact on the outcomes of this process.

Name	**Purpose**
`Life cycle model management` process	Maintain and assure availability of policies, life cycle processes, life cycle models, and procedures for use by the organization
`Quality management` process	Assure that products, services and implementations of the quality management process meet organizational and project quality objectives and achieve customer satisfaction

Table 1.2: Noteworthy organisational project-enabling processes for this thesis

1.2.3 Technical management processes

The **technical management processes** are used to e*stablish and evolve plans, to execute the plans, to assess actual achievement and progress against the plans, and to control execution through to fulfillment.* This category includes

- Project planning process
- Project assessment and control process
- Decision management process
- Risk management process
- Configuration management process
- Information management process
- Measurement process
- Quality assurance process

For the scope of this thesis this processes will be taken into consideration.

Name	**Purpose**
`Project planning process`	Create and manage efficient and practical plans. This will determine the range of project management and technical activities, outputs, tasks, and deliverables, set up task completion timetables, including success criteria, and the resources needed to complete tasks.

`Project assessment and control process`	Assess whether the plans are aligned and feasible. It determines the status of the project, the technical and process performance and directs the execution to help ensure that performance is in accordance with plans and schedules, within projected budgets, to satisfy technical objectives
`Risk management process`	Recognize, examine, address and keep track of risks constantly. It can be applied to risks related to the acquisition, development, maintenance or operation of a system
`Information management process`	Generate, obtain, confirm, transform, retain, retrieve, disseminate and dispose of information to the designated stakeholders
`Measurement process`	Gather, examine, and present objective data and information to facilitate effective management and illustrate the excellence of the products, services, and processes

Table 1.3: Noteworthy technical management processes for this thesis

1.2.4 Technical processes

The **technical processes** are used to *define the requirements of a software system, to transform the requirements into an effective product, to permit consistent reproduction of the product when necessary, to use the product to provide the required services, to sustain the provision of those services, and to dispose of the product when it is retired from service.*

All these processes are generally carried out by the supplier, except for the business or mission analysis process and operation process, which are more domain of the acquirer, with partial involvement of the supplier.

This category includes

- Business or mission analysis process
- Stakeholder needs and requirements definition process
- System/software requirements definition process
- Architecture definition process
- Design definition process
- System analysis process
- Implementation process
- Integration process
- Verification process
- Transition process

- Validation process
- Operation process
- Maintenance process
- Disposal process

Some of these processes require the active involvement of the acquirer, and these will be object of analysis in this thesis. In particular, this work will refer to

Name	Purpose
Business or mission analysis process	Identify the business or mission problem or opportunity that can be addressed. Analyse the potential solutions that could be used to resolve the problem or take advantage of the opportunity. Establish the solution classes that could be employed to address the issue or exploit the opportunity
Stakeholder needs and requirements definition process	Define the stakeholder requirements for a system that can provide the capabilities needed by users and other stakeholders in a defined environment
System/software requirements definition process	Convert the stakeholder's user-focused perspective of desired capabilities into a technical view of a solution that meets the operational needs of the user. This will generate a set of quantifiable system requirements that define, from the supplier's point of view, the characteristics, attributes, and functional and performance requirements the system must have in order to meet stakeholder requirements
Verification process	Provide objective evidence that a system or system element fulfils its specified requirements and characteristics
Validation process	Provide proof that the system, when used, meets its business or mission objectives and stakeholder requirements, and is successful in its intended use in its designated operational environment
Operation process	Establishes requirements for and assigns personnel to operate the system, and monitors the services and operator system performance. To keep services running, this process helps to recognise and examine discrepancies compared to contracts, stakeholder requirements, and organisational limitations

Table 1.4: Noteworthy technical processes for this thesis

All these processes will be often cited in this thesis.

1.2.5 SDLC models in bespoke software development

Another important concept that will be often recalled in this thesis is Software Development Life Cycle Software Development Life Cycle (SDLC). A **software development life cycle** is a *model that covers all the stages of software development*, which are:

1. Planning
2. Requirement analysis
3. Design and prototyping
4. Implementation
5. Testing and integration
6. Maintenance and retirement

The model concept should not be mistaken with methodology. In fact, the former is descriptive and describes what to do, whereas the latter also describes how to do it.
There exist many types of SDLC model, each with different scopes, approaches, and principles [41]. In custom software development, two main elements gained importance over time: the **sequential model** and **agile methodologies**.
The former, also known as the **waterfall model**, is based on the notion of repeatable processes. It provides *strictly sequential phases*. The access to the next phase is governed by the satisfaction of specific pre-conditions; therefore, phases are not overlapping over time. Some extraordinary events may imply to go back from the start. Each phase defines planned activities and expected incoming and outgoing products, contents and structure of documents, responsibilities and roles involved, and product delivery deadlines. The main criticisms leveled against this model involve an *excessively rigid approach towards changes and lack of parallelism.*
As a response, two main corrections were proposed. The former promotes the usage of disposable prototypes that facilitate the comprehension of requirements or the design of solutions. The latter defines returns in the waterfall flow. Each return cycle groups subsequences of phases.

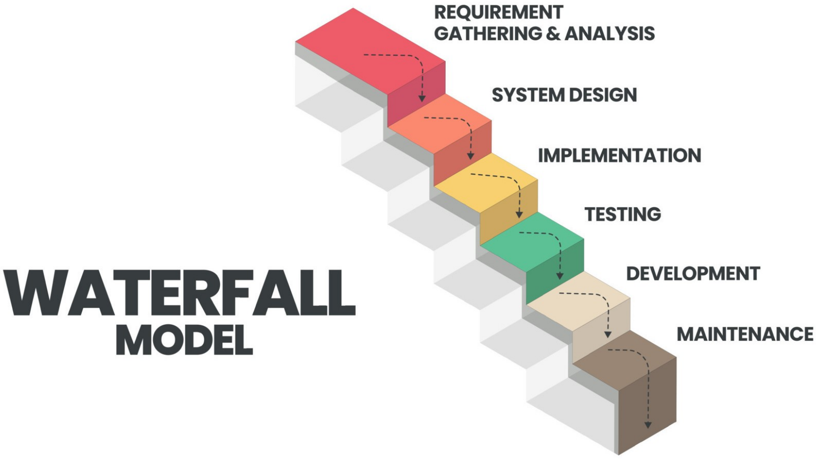


Figure 1.3: The waterfall model, *Source:*[39]

The *waterfall model* is still highly appreciated and requested by acquirers. In fact, it is a *suitable solution for clients who prefer not to be involved in the development process* and simply want to be involved in the beginning and then receive a complete product at the end [51].
This type of contract, called turnkey model, is usually well-known to acquirers. However, turnkey contracts do not include changes during the realisation of the product. Therefore, in these situations, suppliers often ask for additional charges to implement **change requests** [50]. Nevertheless, in software development projects, changes are quite common and unpredictable. Consequently, **agile methodologies** gained crescent popularity. They are driven by the following four guiding principles [31]

- *Individuals and interactions over processes and tools*, an excessive rigidity obstacles the production of value
- *Working software over comprehensive documentation*, maintained documentation does not always coincide with great software
- *Customer collaboration over contract negotiation*, stakeholders must collaborate towards a certain goal, there is no point on setting excessively bureaucratic structures
- *Responding to change over following a plan*, adaptability over sticking to a plan

Agile methodologies are usually the preferred solutions by suppliers, thanks to their adaptability to unforeseen events or new implications.
However, they might be unappreciated by acquirers, especially if they have no previous familiarity with it.

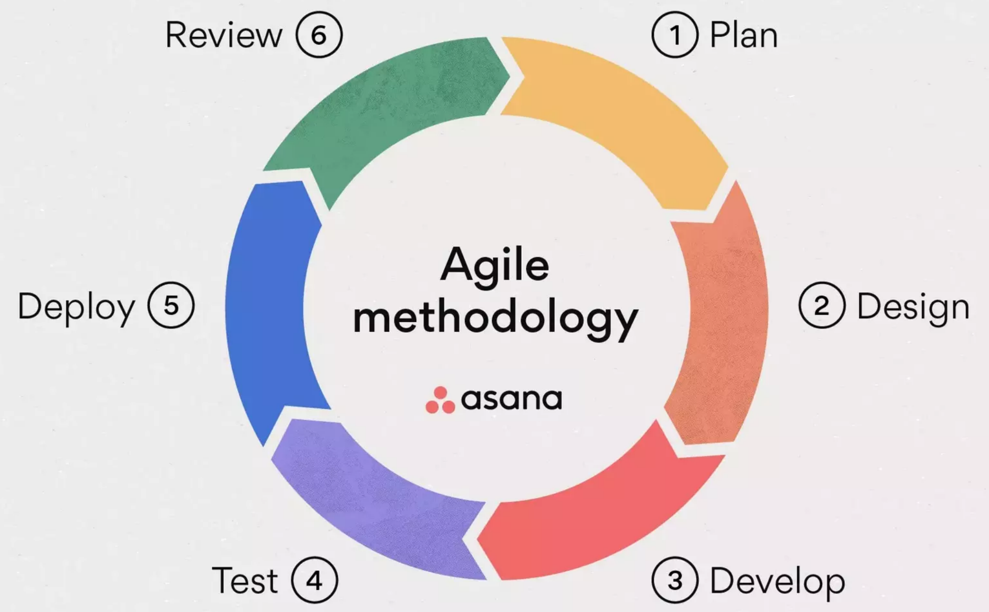


Figure 1.4: Characteristics of the agile methodologies, *Source:*[31]

The choice of the proper SDLC has important implications over the project and it is a noteworthy aspect in the analysis of customer-supplier issues.

1.3 Problem definition and objectives

Custom software development is by nature highly dependent on the interaction between the various stakeholders involved. But this aspect can turn out to be a double-edged sword for many reasons, as it is necessary to converge the interests and the objectives of two (or more) organisations.
Suppliers in fact often report recurring problems that impact the efficiency of the development of the project, and in some cases they may even lead to project failures if not adequately managed. On the contrary, customers might be exposed to uncertainty about the completion of the project within the established times and costs.
Analysing this scenario is not trivial, as many heterogeneous factors are involved. Before arriving at the definition of the problem, it is crucial to understand the risks that customers and suppliers may face.

1.3.1 Risks for the customer

Every software development project can be described in terms of three objectives: what the project needs to deliver, when it needs to be delivered, and within what budget. These **objectives** are generally represented by an *established budget*, a *set of defined requirements*, and *delivery deadlines*. These constraints are linked to each other, which means that it is not possible to change one without impacting the others.
Therefore, the acquirer is exposed to **three main risks**, which are *budget overrun, delays in the delivery of the product, and incomplete requirements development.* [3]
A **requirement** is defined by *an expression*, in the content of a document, *that conveys objectively verifiable criteria to be fulfilled and from which no deviation is permitted if conformance with the document is to be claimed.*
The following picture provides a complete classification of the types of requirements that may be expressed in a software project [15].

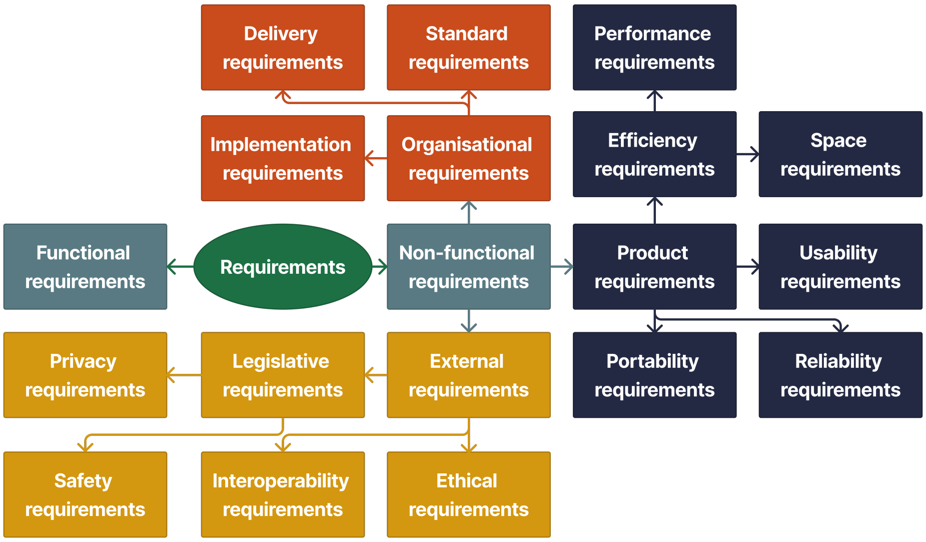


Figure 1.5: Requirements classification

The iron triangle model, presented by Martin Barnes in 1969, represents two possible scenarios in terms of Software Development Life Cycle (SDLC) adopted, resources, time, and scope (requirements).[3]
On the one hand, in a project that is developed through the **waterfall model**, *the scope (completion of the requirements) of the project remains fixed, while economic resources and deadlines may vary over time.*
On the other hand, **agile project**s depict the opposite scenario; that is to say, *costs and deadlines remain strict while the variable part is related to the requirements completion.*

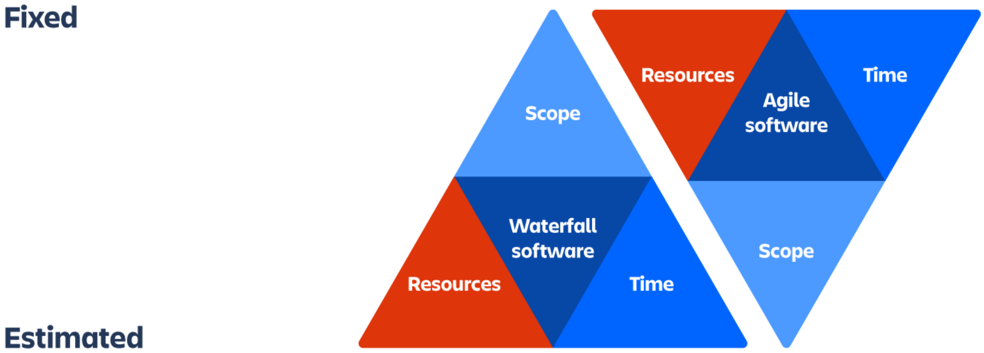


Figure 1.6: Iron triangle model, *Source:*[3]

The possibility of granting all the three aspects cited above with certainty is unrealistic; thus, the acquirer, in the selection process, should be aware of the aspects that characterise each SDLC model cited above.
Finally, acquirers may not have any previous experience in software development project, thus they usually need to take a leap of faith in selecting the supplier. As a result, they tend to trust the opinions of acquaintances who have already developed software projects with a specific suppliers.

1.3.2 Risks for the supplier

On the supplier's side, risks are instead less evident and highly dependent on the interaction and agreements with the supplier and the state of the art technology involved in the development of the project.
It is a common opinion to think that the contract proposal, which is usually drafted by the supplier, tends to propose favourable conditions for the supplier.
Instead, in many cases, *suppliers try to acquire clients by proposing more favourable prices or less stringent conditions* (especially for small and medium-sized organisations), or simply do not consider certain aspects that are usually added as clauses after previous bad experiences.
Furthermore, *suppliers tend not to enforce many clauses to avoid appearing unfriendly or losing their clients*, since generally *most leads* in bespoke software development organisations *are usually generated by word of mouth*, often independently of aspects such as the size of the company.
Furthermore, *suppliers* usually *need to work for customers with different levels of expertise in software development*, and therefore the explanation of certain principles and ways of working is not immediately clear to inexperienced acquirers. This knowledge

gap between the two parties is usually translated into suspicious tendencies expressed by potential leads.

Given these premises, the main risks for the supplier's side involve *loss of income on the project, overwork, tension and stress on the assigned team, bad reputation and loss of potential lead generated by word of mouth, loss of projects, and consequent inefficiency of employees* due to work reorganisation and idle times.

The *necessity of making a great impression on the acquirer* usually implies the *proposal of discounts on the initial price*, which may become even bigger in cases of supplier's faults (sometimes even if the supplier does not have any responsibility), erasing by far the margins of a possible profit.

Furthermore, especially with inexperienced customers, suppliers tend to propose *contracts that bind the development of all the requirements to a fixed budget* (usually referred to as the turnkey model) *with additions of specific delivery deadlines*, taking on all the risks associated with incorrect initial estimates or uncertainty about the technologies or the feasibility of the project itself.

In some cases, the necessity of completing all the requirements in a given time and with a given final price may imply *overwork* or a diminishing value for the cost/hour ratio, situations that generally produce *tension and stress over the team.*

1.3.3 State of the art - Customer-supplier issues in custom software development

Customer supplier interactions in bespoke software development have been studied for over forty years and still represent a relevant theme to be discussed on.

The analysis of this type of relationship spans over multiple heterogeneous topics, spacing from software development life cycles, psychological factors, marketing aspects, and legal and economic agreements between the two parties.

Many previous studies (e.g. Reel, 1999 [2]; Hanssen & Fægri, 2006 [20]; Wiegers & Beatty, 2003[59]) pointed out the *importance of achieving an optimal level of engagement and satisfaction of the acquirer during the whole development of the project*, to the extent of reaching the project goals with the predefined resources.

Moreover, several authors (e.g. Wallace & Keil, 2004[49]; Boehm, 2002[7]) also agree on the fact that the *customer (or some aspects related to him/her) is* usually *one of the main factors that cause project failures* or consistent budget overruns (up to 34% of total development costs, according to Jorgensen and Molokken-Ostvold, 2006[27]).

Issues like *shallow definition of requirements, poor communication between stakeholders, scope creep and lack of prioritisation, unrealistic expectations, and poor commitment* are likely to have a negative impact on the project (Cerpa & Verner, 2009[8]; Lehtinen et al, 2014 [33]).

However, a shared misconception claims that good project managers are always able to address every problem that may arise from the interaction with the supplier. Keil et al, 1998 [28] confute this statement by affirming that *not all customer-related issues cannot be directly addressed by the project manager*, since they are related to the customer's behaviour. According to their work, **customer mandate** (risks related to the relationship with the customer and his/her expectations) and environment risks (external risks such as natural disasters or events that may influence the project) are the two classes of risks on which the project manager can only partially influence or just try to foresee. A tangible example of actions that cannot be directly controlled by the project manager is the completion of tasks that can only be carried on by the supplier, such as the provision of essential material or the sharing of the access to private information (e.g. credentials or business plans). Therefore, the project manager should be able to forecast customer-related issues to achieve success in project implementation. The adoption of known best practises might then be the key to addressing these issues.

The current state-of-the-art includes some qualitative analyses that were conducted in order to analyse limiting and enhancing factors that affect relationships among customers and suppliers. In particular, this thesis refers to the work of Pasi, Nikula, Smolander, 2008 [36]; Hanssen & Fægri, 2006[20]; Siddique & Hussein, 2016 [44]. The first work focuses on the specific customer-supplier analysis through the application of grounded theory to a set of interviews to different roles of software development organisations. The objective of the survey includes the identification of customer-related problems that, according to suppliers, negatively impact the completion of the project.

Hanssen & Fægri's study instead focuses on customer engagement in projects that adopted agile methods through a longitudinal case study, following a migration from the adoption of a waterfall model to an evolutionary project management process.

In conclusion, the last paper targets the main aspects that promote or hinder customer involvement in agile software projects, focusing on the Norwegian software industry. Also in this case, interviews and qualitative analysis through the adoption of grounded theory were the main tools used to conduct this study.

These three articles were the source of inspiration for this thesis, which converged to the analysis of customer-supplier interactions after a previous iteration aimed at formulating and confusing four hypotheses in potential customer-supplier issues in bespoke software development. These hypotheses are discussed in paragraph 1.3.4, while the selection of the case study will be discussed in chapter 2.

1.3.4 Scope of the study

This **thesis** aims to *analyse the main problems faced by bespoke software development organisations with their customers.*

This broad definition of the problem was established at the beginning of the project as a precise objective related to the job of a project manager.

The work carried out must be seen from the perspective of an iterative process, whose outcome is the selection of a specific scope of the set of problems that are associated with the customer-supplier relationship.

In order to obtain the **domain of the possible targets of this analysis**, I conducted a preliminary market investigation through a systematic review of the state of the art, some brainstorming sessions with peers in my company, and a set of ten interviews with different project managers of bespoke software development organisations.

The first version of the questionnaire was very basic and short, as it had to be completed within a time of a phone call with a prospect (approximately 10 minutes). In this phase, prospects were project managers of software houses that work on custom-made products.

The questionnaire aimed to let the prospects of the interview talk about the recurring problems they experience and the solutions they adopted to mitigate or overcome the reported issues. The next step in the call was the engagement of the prospect in order to convert him/her into an interviewee for the next phase of the process, which is described in chapter 2.

ID	Question
QP1	Does your company mainly work on bespoke software projects?
QP2	Which are the most recurring and annoying issues that your company experiences?
QP3	Did you try to address them? How?

Table 1.5: Questions of the first iteration of the questionnaire

The transcriptions of the interviews, in addition to various blog posts and dedicated pages were then used to perform a *Thematic Content Analysis* [4]. Its results were then used to select some categories of interest from the variety of problems experienced by software houses in bespoke software development. The results of this work defined **four main possible problems** that could become objects of analysis for this thesis.

1. **Selection of the appropriate software development life cycle**

   The **selection of the SDLC model** is a crucial aspect for a software development project. In fact, an incorrect choice can influence the success of the project itself.
   However, *suppliers usually embrace only one or two SDLC models* (usually waterfall or agile), considering only a limited subset of possible parameters that may influence the choice.
   Furthermore, suppliers prefer to adopt the Scrum [56] framework or their own methodology (which is usually an adaptation of Scrum obtained by removing certain events or figures of the framework itself) in every project. This approach is intrinsically not optimal.

2. **Analysis of customer-supplier interactions**

   Many prospects reported having issues of various nature with their customers. These aspects generally cause a sense of dissatisfaction in both parties, noticeable delays in the delivery of the project, or budget overruns.
   The problems that were reported were usually shared by different suppliers, making this scenario an interesting use case for an in-depth analysis.
   However, it is also crucial to understand the needs and reasons behind certain **behaviours manifested by customers**.
   Therefore, the objective of this study involves the identification of a relevant set of *recurring problems in customer-supplier relationships*, followed by the **highlighting the characteristics shown by both parties that can represent a risk to the project** from this perspective.

3. **Improvement of the estimation of the contingency factor in the quotations**

   Estimates of costs made by software houses always vary according to a value, called the **contingency factor**, which is the *translation of the estimated risk reported in the final price.*
   Providing an adequate estimate of the contingency factor is a crucial but not trivial task.
   On the one hand, a contingency factor that is too low may help in proposing competitive prices, but may also expose the company to earning losses (especially in turnkey contracts).
   On the other hand, contingency factors that are too high may grant a reduced exposition to earning losses, but may also propose prices that are perceived as excessive by the acquirers.
   However, *many software houses tend to propose the same contingency factor for every project*, or they try to compare the project to previously developed ones without adequately analysing all the possible risks that may arise in the development of the new one.

4. **Analysis of the main critical issues and weaknesses of contracts between the two parties**

   A subset of software houses, especially the smaller ones, reported **issues with the management and application of their contracts**. Project managers reported that their contracts often present issues such as u*nforeseen scenarios, invalid clauses, or long redaction times.* Some suppliers also reported adopting a more risky approach, regulating only the core aspects of the project and applying as little as possible any kind of clause present in the contract. This approach is necessary, as the contract management process would take too long to be completed.

In conclusion, this chapter aimed to provide the reader with some essential information to understand the concepts described in this thesis. Chapter 1 also described the preliminary work that was carried out to obtain the domain of possible noteworthy problems to be analysed. However, the resources available for this work led to the necessity of selecting one of these four problems.
Chapter 2 will be dedicated to the performance of this selection, in addition to the effective analysis of the problem and the description of the corresponding methodology that was applied.

Chapter 2

Analysis of the possible solutions

This chapter describes the transition from the hypotheses domain to the selection of the problem that was the object of interest in this thesis. In addition, a description of the tools, the methodology, and the steps used to select the object of interest and perform its analysis in comparison with the current state of the art will be provided to the reader.

2.1 Domain of the problems

The analysis of the market and the first iteration of the questionnaire presented in paragraph 1.3.4 highlighted the **four problems** that will be the object of interest in this chapter. Every problem will be referred to through its ID.

ID	**Problem**
P1	Selection of the appropriate software development life cycle
P2	Analysis of the interactions with the customers
P3	Improvement of the estimation of the contingency factor in quotes
P4	Analysis of the main critical issues and weaknesses of contracts between the two parties

Table 2.1: Set of problems considered in the domain

The following sections aim to describe in a detailed manner each problem spotted by the first set of interviews.

2.1.1 P1 - Selection of the appropriate software development life cycle

The selection of the proper SDLC model plays an essential role in all software development projects. However, performing the right choice in choosing the proper SDLC model to be adopted in a project is non-trivial.

Each SDLC model is *driven by a specific objective*, such as quality assurance, risk mitigation, team scaling, etc.
The survey produced by Ruparelia [41] provides a complete overview of the SDLC models and their application scenario.

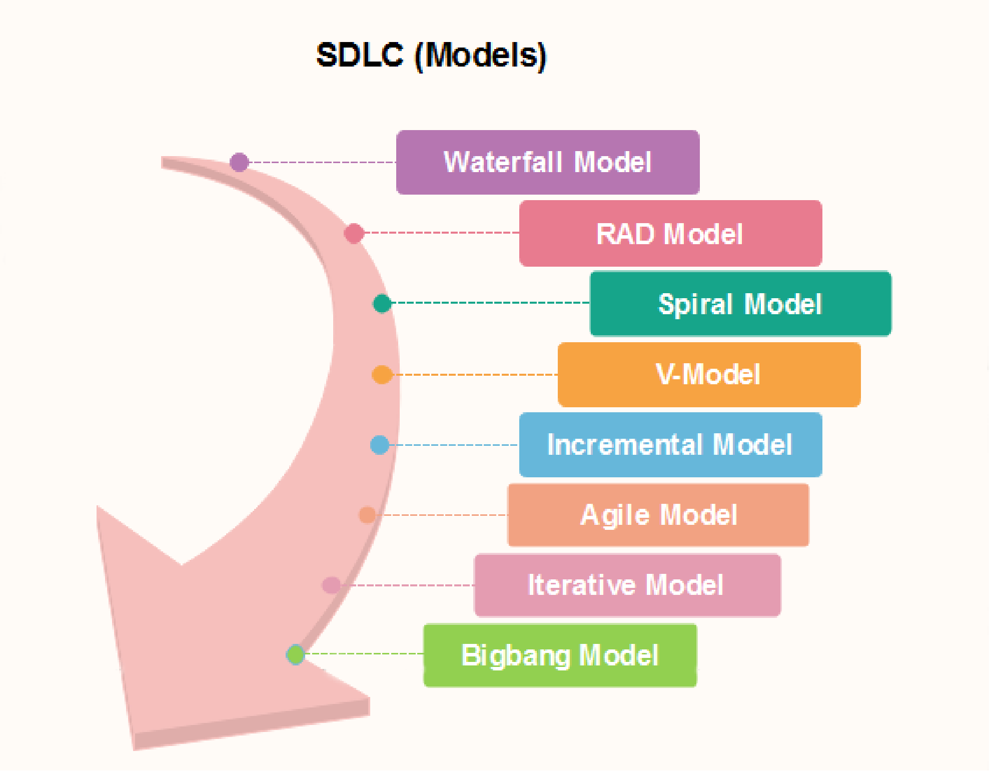


Figure 2.1: Kinds of SDLC models, *Source:* [42]

The selection of software development life cycles also has an *high impact on the way of working of the supplier, the costs of the project, the level of involvement of the customer in the projects* (in terms of role, responsibilities, control on the project, and risk sharing) *and project planning* (with reference to expected outputs and delivery times).
The adherence to an SDLC model must be properly supported by both parties, and with reference to the supplier it must be properly achieved to a proper application of the software life cycle model management process. In particular, the supplier must grant adequate capacities and capabilities, life cycle processes, policies, and procedures to ensure the proper adoption of the SDLC model.
A common problem experienced by suppliers is the *way to perform this choice*, as the selection is highly dependent on a long list of factors. An interesting objective for this analysis would involve the design of a process in which, given in input a series of predefined parameters, returns the best possible (or a ranking of) SDLC model to be applied.

2.1.2 P2 - Analysis of customer-supplier interactions

Bespoke software development is by definition highly dependent on **customer-supplier interactions**. Customers in need of purchasing tailor-made software form a vast and heterogeneous set. Each customer may differ in factors such as level of expertise in the IT sector, previous experience in developing software projects, type of required project, attitude towards it, etc.
Every step of the development of the project (which can be consulted in the image below) requires defining specific touch points and assignments for the customers, also according to the selected SDLC model.

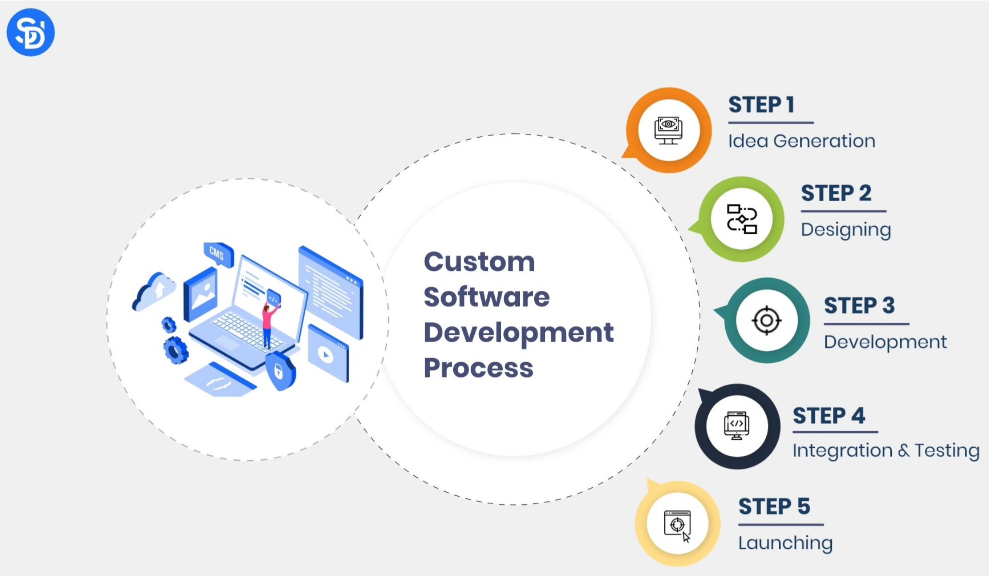


Figure 2.2: Steps in the process of custom software development, *Source:* [6]

Every phase of the project (and even the stages that precede its development, that is to say, when the customer is still a lead or prospect) is important as it allows suppliers to gather information about the acquirer's (potential or real) behaviour.
In particular, suppliers would be interested in understanding *which factors of a customer may cause potential issues in the development of the project*. This would allow them to properly assess and treat them or decide not to take the project.
However, there is no shared and clear set of factors that should be considered. Consequently, the correlation between some unknown factors and their effects on the project might be unclear or hidden to the supplier.
In conclusion, P2 can be formalised as the *need to identify a set of traits of acquirers' behaviour that can translate into potential issues during the project development,* in addition to their correlations with the effective problems that are likely to be verified.

2.1.3 P3 - Improvement of the estimation of the contingency factor on quotes

Estimates always represent a topic of discussion in custom software projects. Negotiations on prices, delivery deadlines, and included features can last for several

months. These situations can potentially result in significant losses for the supplier if the estimate drafting is not paid.
The following picture lists a set of factors that must be taken into consideration while drafting an estimate for a custom software project

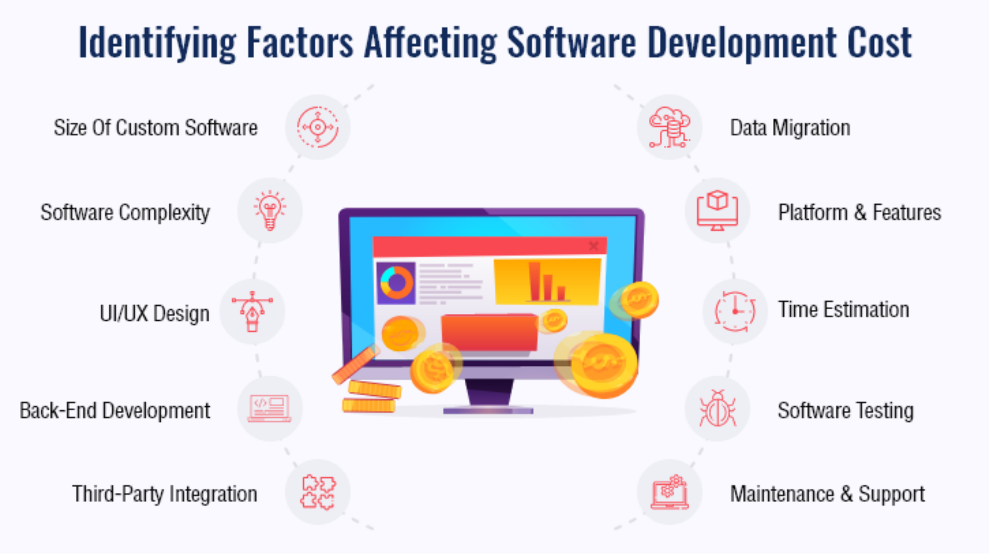
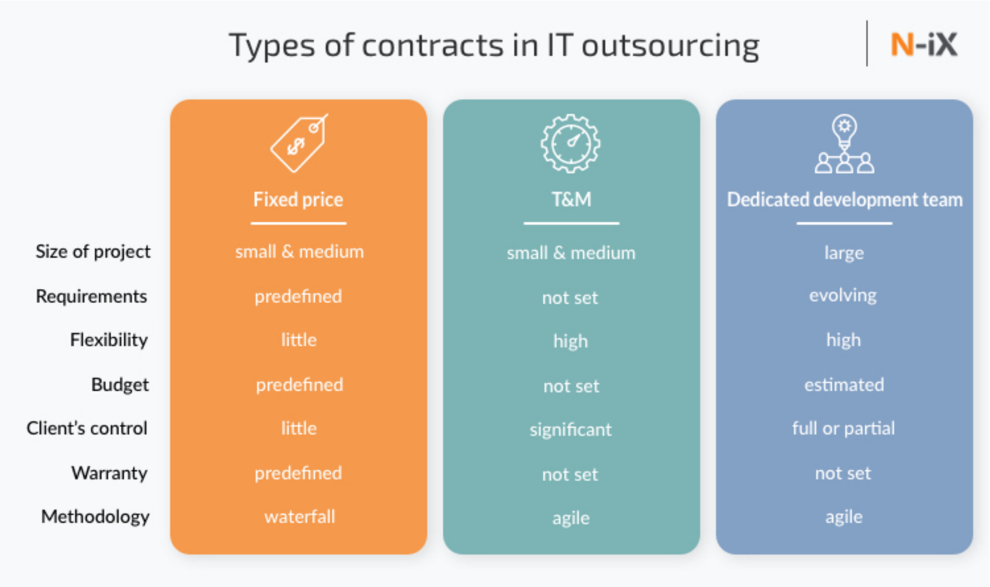


Figure 2.3: Factors that affect custom software estimates, *Source:* [5]

However, estimates must also take into *account risks and unforeseen events* that might have a relevant impact on the financial sustainability of the project development. This consideration is usually taken into account by estimating a **contingency factor**. Estimating this value is not a trivial task, as the contingency factor represents a double-edged sword for the market competitiveness of the supplier.
As reported in section 1.3.4 , on the one hand, a contingency factor that is too low may help to propose competitive prices, but may also expose the supplier to earning losses (especially in turnkey contracts). On the other hand, contingency factors that are too high may grant a reduced exposition to earning losses, but may also propose prices that are perceived as excessive by the acquirers.
Therefore, *optimisation of the estimation of the contingency factor* may represent a noteworthy object of interest for the suppliers.

2.1.4 P4 - Analysis of the main critical issues and weaknesses of contracts between the two parties

Contracts do represent another topic of discussion in agreements between customers and suppliers in custom software development. Also in this case, negotiations may tend to stack up over time and result in a loss for the supplier upon missing an agreement between the two parties.
In this sector, suppliers can generally offer three types of contracts, which are fixed price (or turnkey), Time & Materials (T& M) or dedicated team ones. The following image reports the characteristics of each of them



Types of contracts in IT outsourcing | N-iX

	Fixed price	T&M	Dedicated development team
Size of project	small & medium	small & medium	large
Requirements	predefined	not set	evolving
Flexibility	little	high	high
Budget	predefined	not set	estimated
Client's control	little	significant	full or partial
Warranty	predefined	not set	not set
Methodology	waterfall	agile	agile

Figure 2.4: Types of contracts in bespoke software development, *Source:*[21]

From the first set of interviews, the majority of the interviewees declared that their organisations adopted their own **templates** for all or almost all the classes of contracts reported above. Templates are updated once serious threats are discovered or negative events occur.

However, a subset of suppliers declared to experience *recurring problems* with their contracts. First of all, *they do not have the capacities to afford and structure proper templates*, thus it takes a lot of time for them to draft contracts. In addition to that, they tend not to enforce too many clauses for two main reasons. The former includes the effective *validity of the clauses*. In some cases, the interviewees had doubts about the effective proper applicability of the clauses, as in their opinions some clauses were open to different interpretations.

The latter instead involved the *will to maintain positive relationships with their customers*, even if they were not fully compliant with the contract.

Therefore, P4 can be formalised as the *necessity of reducing the time needed to draft a contract with a corresponding solid legal validity.*

2.2 Narrowing down to the problem of interest

Each problem has a relevant impact on the relationships between customers and suppliers in the development of bespoke software.

Nevertheless, their areas of interest are vast and heterogeneous. The **limited time and resources** available for the development of this project necessarily required making some choices to **focus only on a specific element** of this domain.

Therefore, the second phase of the project aimed to choose one of these topics.

2.2.1 Methodology for the selection of the topic

The state of the art suggests many methods to perform project selection. In this case, it is hard to establish quantitative metrics that could be relevant for this scope.

Traditional metrics, such as Return on Investment (ROI), Rate of Return (ROR), and earned value, could not be easily estimated.
The criteria for the selection of the object were mainly qualitative. They took into consideration

- The **relevance of the problem for the majority of the suppliers** who would have been interviewed
- The **feasibility of a noteworthy contribution in a period of six months**
- **My level of expertise in the specific sector** of each problem
- **My level of capability and capacity to** eventually **resolve the problem**

Therefore, given the various methods for the selection of a project [48], I decided to restrict the choice between the ones that were more suitable for a *qualitative-based selection*.
In particular, the ranking method, the scoring model and the analytic hierarchy process were the possible candidates.
The **ranking method** is quite simple and straightforward. It *arranges the projects on a scale* of one to four (in this case) *based on their importance*, and it allows identification of top priorities. It works effectively when there are limited criteria to evaluate, and it is easy to assess the factors involved.
The **scoring model** works when there are many selection criteria to consider and the projects compared differ considerably between each other. Instead of selecting one or two criteria as in the ranking method, the scoring model *considers groups of factors* (e.g. strategic alignment, benefits, risk). Every criterion receives a rating, and *every group is given a weight*. The *weighted average score is* then computed to arrive at *the final project score*.
At last, **analytic hierarchy process** combines subjective elements with mathematical models to provide a more holistic technique than the other cited methods. It suits particularly well to complex project evaluations. Similarly to the previous model, the analytic hierarchy process works with a long list of selection criteria. However, it does a *pairwise comparison, putting every two criteria against each other*. This technique reduces the possibility of errors and biases. After this pairwise comparison is completed, *values are normalised, and the weighted score is computed*.
Given the *restricted set of elements in the domain* and the *limited number of parameters* that could impact the choice, I decided **to opt for the ranking method**. In addition to my ranking, I also decided to **involve suppliers in this selection**.
In order to do so, I performed **the second iteration of the questionnaire**.
This second version was structured as follows

1. Interviewee profiling
2. Investigation of interest towards problem P1
3. Investigation of interest towards problem P2
4. Investigation of interest towards problem P3
5. Investigation of interest towards problem P4

Every section of this iteration of the questionnaire will be presented in the following paragraphs.

2.2.2 Interviewee profiling

This part of the interview aimed to *highlight the main characteristics of the interviewee's organisation*, to the extent of relating the characteristics of the supplier to their needs and expectations analysed in the remaining part of the questionnaire.
The interviewee profiling section contained the following questions

ID	**Question**
Q1	What kind of services does your organisation develop? Do you mainly work on internal products or on bespoke/commissioned software? In what indicative percentage do you deal with commissioned software? How many commissioned projects do you develop in a year?
Q2	How long do commissioned projects you develop usually last (in months)?
Q3	Which SDLC models do you follow/contract types that you offer? Does your organisation have any qualifications/certifications (e.g. ISO 9001/27001, Scrum certifications, Prince certifications)?
Q4	On average, how many people make up your work team in a bespoke project? How many employees/collaborators are there in the organisation?

Table 2.2: Questions included in the supplier profiling section

This section of the questionnaire aimed to frame some strategic aspects of the interviewee's organisation.
First of all, the first object of interest included the *importance of bespoke software development for the interviewee's organisation.* Q1 aimed, in fact, to highlight how relevant is bespoke software for the interviewee's organisation in terms of effort in a year.
Another important consideration about the interviewee's organisation was related to the *average duration of their project.* Suppliers with longer (and lots of) bespoke projects might in fact be heavily affected by the issues discussed in this thesis.
The third question aimed to *understand the offer and expertise of the supplier interviewed in terms of SDLC models.* The second objective of this question included *tracking of the most followed SDLC models* (or applied methodologies) by organisations that work on custom software.
Finally, the fourth notable aspect of the profiling took into consideration the *size of the organisation* and its teams.

2.2.3 Investigation of interest towards problem P1

This part of the interview was guided by **two main objectives**.
The former aimed to *analyse in a qualitative way the implementation of the life cycle model management process* provided by the interviewed supplier.
The latter instead wanted to *highlight the feelings and sensations of the interviewee towards the implementation of this process.*
It is essential to understand that the objective of these questions was qualitative. They

aimed to understand the main aspects of problem P1 without focusing too much on the details.
This section contained the following questions

ID	**Question**
Q5	How challenging is it for you to choose the right software development cycle? Are you generally satisfied with the choice afterwards?
Q6	Which factors do you usually consider when selecting a SDLC for the project?
Q7	Have there been cases where you were particularly dissatisfied with the development cycle of the software adopted / Did you encounter problems during the progress of the project? Are there such situations that repeat themselves with recurrence? If so, have you tried to direct them? If so, how?

Table 2.3: Questions included in the investigation of interest towards problem P1

Each question aimed to make different correlations between the problems and the stage of the process in which they might be verified.
Q1 wanted to *investigate whether the supplier had a method of selecting the SDLC model* and *how it worked.* Furthermore, Q1 wanted to capture eventual problems freely exposed by the interviewee.
Q2 instead wanted to *investigate the reasoning made by the interviewee's organisations behind the establishment of the life cycle model management process.*
In conclusion, Q3 presented two main objects of investigation.
The former wanted to *grasp eventual dissatisfaction towards the process*, in order to bring out possible undetected problems.
The latter instead targeted the *approach of the interviewed supplier towards these problems.* A proactive approach in attempting resolutions is usually synonym of major interest towards the problem and more need of facing/resolving it.

2.2.4 Investigation of interest towards problem P2

This part of the interview wanted to *grasp the sensations of the suppliers towards their customers, their behaviour, and the consequences the latter has on the project.*
The investigation was still qualitative as it was mainly concerned with understanding the cause-effect relationships rather than gathering quantitative information.
In custom software development projects, the sharing of information and materials is usually bidirectional. The flow that goes from the customer to the supplier assumed a major relevance for this part of the interview. Materials and information that pass through this way include images/texts/videos, credentials for third-party services accounts, devices and sensors (e.g. in IoT projects), crucial information to define (and refine) use cases and feedback on the delivered versions of the product.

This section contained the following questions

ID	Question
Q8	Do customers generally meet your expectations/constraints related to the commitment required of them in the development of a project (e.g. meeting deadlines, clear definition of project constraints and requirements, active participation in project phases)?. What initiatives could be useful to solve the problem?
Q9	Are you generally able to maintain consistency between the activities envisaged by the software development methodology and those actually carried out during construction (in terms of times, estimated costs, and project objectives)? With which development model(s) does this occur?

Table 2.4: Questions included in the investigation of interest towards problem P2

The first question aimed to point out the *expectations that the interviewee had towards his/her customers.* This question let the interviewee express his/her point of view, relating it to the problems that are frequently verified in his/her organisation. As stated before for Q3, the second part of the question wanted to analyse the *approach of the interviewee towards the resolution of the problem.*
Q9 instead wanted to *grasp the cause/effect relationships between the perceived behaviour of the customer and the repercussions it has on the project* in terms of additional costs, relevant delays, or downsizing of the goals. The last part of the question wanted to explore the relationship between the selected SDLC model, the presence of problems in the problems in the project, and the impact that the customer's behaviour had on these situation.

2.2.5 Investigation of interest towards problem P3

The investigation of interest towards problem P3 targeted *the approach of the interviewee's organisation towards effort estimations.* In particular, the supply process, project planning, life cycle model management, risk management, and stakeholder needs and requirements definition process implementations were targeted in this analysis. The investigation was still qualitative, as it is mainly concerned with understanding the cause-effect relationships rather than collecting quantitative information.
This section wanted to gather information about the *way of working of the interviewed supplier towards the drafting of quotations.* Estimates are generally a noteworthy argument of discussion and negotiation between the two parties in bespoke software development. Therefore, it would have been interesting to analyse the behaviour and sensations of the suppliers towards their quotations. In this way, problems related to estimates could emerge from the interview.
An important role in quotations is played by the **contingency factor**, which is a *provision that the cost estimator makes to cover unforeseeable expenses the project may incur* [32]. The contingency factor estimation is a non-trivial operation. On the one hand, a high contingency factor may ensure the supplier coverage from possible risks (including changes or gaps in the analysis), but may also prevent the supplier from

offering competitive prices. On the other hand, a low contingency factor may ensure more competitive prices, but may not cover unforeseen events.

ID	Question
Q10	How much does the choice of the software development cycle influence the contingency factor of the quotation?
Q11	Generally how do you estimate project costs a priori? How important is the choice of software development cycle in your estimation? How long does it usually take you to estimate the costs of a project?

Table 2.5: Questions included in the investigation of interest towards problem P3

The first question aimed to *analyse how impactful is the selection of the SDLC model in economic terms*, that is to say how much the contingency factor is influenced by this choice. On the one hand, turnkey contracts shift the risks to the supplier as this figure is forced to ensure all the requirements stated in the contract. Therefore, the final price must take into account all the risks associated to the completion of the requirements (therefore the contingency factor should me much higher). On the other hand, agile models allow the sharing of risk between the two parties. Hence, the contingency factor can be reduced[10].
Q11 instead let the interviewee *describe his/her way of working in cost estimations*, describing the parameters that are taking into account, their relevance, and the average time required to make a quotation. Q11 also wanted to investigate whether the selection of SDLC model (and consequently the application of the methodology that the organisation developed in order to effectively apply the concepts of the model) brings other considerations into the quotation. An example might be the necessity of carrying out different activities depending on the choice made.

2.2.6 Investigation of interest towards problem P4

The last part of the interview aimed to extract significant information related to the *perceptions of the interviewees about their contracts*, their way of drafting and managing them, and their eventual previous experiences with cloud-based third party services that manage legal contracts.
Contract negotiations are another recurring and important theme in customer-supplier relationships. Negotiations, in fact, can last for a long time, and the application of the clauses can pose a threat to the maintenance of good customer relationships.

ID	Question
Q12	How long does it usually take you to draft a contract?
Q13	How many changes are made on average to contracts in a year?
Q14	How often do you consult contracts during a project?

Q15	How do you manage and keep your contracts? (e.g. paper archive, signed and scanned copies on Drive, cloud software for drafting contracts)
Q16	Have you ever used cloud services for contract automation? If so, how was your experience?
Q17	Do you make use of legal advice for the drafting of contracts?
Q18	Have there been cases where you were particularly dissatisfied with the contracts you use to regulate the development of commissioned projects? Are there such situations that repeat themselves with recurrence? If so, have you tried to direct them? If so, how?

Table 2.6: Questions included in the investigation of interest towards problem P4

This set of questions was conceived with three main objectives as guidance. Q12, Q15, and Q16 aimed to *detect potential interest in contract automation* by gathering eventual previous experience with cloud-based contract management tools, the time required to redact a contract, and the actual tools used to store and draft this kind of documentation.
Q13, Q14, and Q17 aimed to explore the *way of working in drafting the contracts and their effective usage* (consultation/application and updates) during the project development.
At last, Q18 wanted to grasp the *eventual dissatisfaction related to the contract management process*. As for the other similar questions, the last part of Q18 explored the interviewee's attitude of the interviewee towards the exposed problems.
At the end of the interview, I also **asked the interviewees to rank each problem according to the perceived impact on their organisations**. This classification was an expression of the relevance of the problem for the majority of the suppliers, and it will be discussed in paragraph 2.2.8.

2.2.7 Results of the interviews

The questionnaire was submitted to a set of 10 project managers/CEOs of different suppliers located in the Veneto region. The set of interviewees presented the following characteristics

- Size of the organisations included between 3 and 250 employees/collaborators (even though the majority counted between 20 and 50 total employees)
- Mainly working on bespoke software, at least 60% of the total time (most of the organisations declared to work on bespoke software for over 85% of their time).

P1 - Selection of the appropriate software development life cycle

The data set obtained from the interviews revealed noteworthy details about each problem. For what concerned P1, *suppliers declared to follow two main models*: **agile and waterfall**, independently of the size of the organisation. However, this parameter

had repercussions on the actual kind of agile methodology that was applied by the respective organisation.
On the one hand, bigger suppliers (>30 employees) declared to adopt the Scrum framework [56] more frequently, while the smaller ones preferred to stick to their own methodology, which was usually obtained by removing some events and figures presented by Scrum. Smaller suppliers frequently declared that they did not have sufficient resources and expertise to properly apply Scrum. Some interviewees also declared to adhere to the extreme programming methodology [52].
Other models, such as the V-model [41] or the Spiral one[41], were usually less or not known to the interviewees.
The *suppliers also declared that they could easily choose the SDLC model that suits the project best* (among waterfall and agile), since the choice was usually immediate. In fact, suppliers tended not to follow more than two models, and they were able to choose according to the acquirers' will and some parameters (e.g. available budget, project type, size, B2C/B2B target).
However, most of the suppliers have declared that *they experience recurring bad situations, especially in cases where methodologies based on the waterfall model are adopted.* Interviewees stated that in these cases, **scope creep** was likely to be verified. In particular, on many occasions, stakeholders (on the acquirer's side) that should have been involved during the planning and analysis phases of the project suddenly take decisions that usually affect the objectives of the project. *Suppliers tended not to like this approach* as they were *continuously forced to repeat some previous stages that were already completed.* Therefore, they needed to ask for a budget increase for change requests, which the client usually did not want to accept.

P2 - Analysis of customer-supplier interactions

P2 was the selected object of interest. We will see in paragraph 2.2.8 the reason that led to the selection of this specific problem.
Q8 and Q9 clearly highlighted many notable factors.
First of all, *almost every supplier declared experiencing various well-known problems with at least 40% of their customers.* Interviewees also declared that their *expectations* towards their customers were *frequently unattended.* In particular, in the Q8 responses, every interviewee reported having a considerable subset of *clients that represent the main cause of delays in the delivery of the project.* Tasks like materials (e.g. services accounts, documentation, media and texts, and hardware) provisioning and requirements definitions tended to be delivered outside the deadlines agreed by the two parties. These situations caused noticeable delays, which can stack up to several months. These issues might grow up to the point of representing a threat to the correct realisation of the project.
Suppliers reported to have tried *different solutions*, especially related to the *insertion of dedicated clauses in contracts* or about *client training* (referred to as the set of activities aimed to explain and accustom the client to work on software development projects). However, in most cases these solutions did not work. In fact, *contractors tended not to enforce these clauses*, as they would have *compromised the health of the relationships between the two parties.* These behaviours led to a *forced adaptation of the way of working and the expectations of the supplier*, causing frustration, deadlocks, and losses for the interviewee's organisation. These aspects were predominant in the set of responses to Q9.
Another recurring situation that emerged from Q8 pointed out the *progressive disinter-*

est shown by the acquirer towards his project. As clues to this trend, the interviewees stated that their customers usually participated in an active, but often shallow way to the definition of requirements. Furthermore, acquirers *did not frequently test functionalities* in staging environments. A noteworthy aspect that emerged from the responses to Q9 highlighted how these problems were more *likely to verify if the waterfall model was adopted.*

P3 - Improvement of the estimation of the contingency factor in quotes

The investigation of interest towards problem P3 also highlighted interesting aspects in customer-supplier issues. This problem gained a lot of interest but its importance was perceived as minor with respect to P2. On the one hand, most of the interviewees belonging to larger organisations declared that *the choice of the followed SDLC model had a considerable impact on the contingency factor of the quotation.* They confirmed that they used a *smaller contingency factor in agile-based projects* and a *larger one in waterfall-based ones.* In particular, the average contingency factor for an agile-based project was included between 15% and 20% of the total cost of the project, while its corresponding value in waterfall-based projects was included between 25% and 40% (with peaks of 50% in scenarios with considerable risks). On the other hand, *smaller organisations* partially confirmed what stated above, but they also *reported having harder times in making their acquirers understand this reasoning.*
The other factors that emerged with relevance to the estimation of the contingency factor can be classified into three main categories

Category	**Factor**
`Acquirer related factors`	• Level of previous experience with the customer • Perceptions about the attitude of the customer towards the project • Level of expertise of the customer in the IT sector and in the core business subject • Time that the acquirer can dedicated to the project • Degree of variability of the budget allocated for the project • Requirements analysis already been done • Time and cost constraints • Performance requirements • Vision of the project (long or short term)

Supplier related factors	• Level of expertise on the subject • Similarities to previous projects and degree of replicability of code from other projects • Selected SDLC model • Degree of expertise related to the involved technologies • Perceived degree of complexity • Maturity of the technology
Other external factors	• Legal requirements • Additional privacy and security aspects (over the minimum required level) • Chance of catastrophic events

Table 2.7: Reported factors that can affect the contingency factor estimation

Many interviewees declared noteworthy times for the drafting of an estimation, with an average time of three weeks.

P4 - Analysis of the main critical issues and weaknesses of contracts between the two parties

In conclusion, the responses related to the P4 questions marked some very specific scenarios attributable to the characteristics of the interviewees' organisations.
In particular, issues experienced by the suppliers were related to the number of employees and the philosophy towards their customers.
Smaller companies (< 10 employees) usually tended to maintain a lighter contractual approach towards their customers for three main reasons.
First of all, one of the *strengths leveraged and sponsored by smaller suppliers* was the *ability to better adapt to the needs of a customer.* Hence, forcing the signing of complex contracts would be synonymous with incoherence.
Furthermore, these suppliers declared that *they did not have the resources to maintain more structured and complex contracts.*
In the end, this specific subset of interviewees claimed not to have interest in redacting more detailed contracts for small projects, as that task would have taken a *considerable amount of time in relation to the total time dedicated to the project.*
On the one hand, this approach made contract management sustainable for smaller organisations, but on the other hand, it exposed them to a lot of threats. Interviewees who were in this category reported experiencing recurrent problems. The high level of *adaptability to customers' needs* represented a double-edged sword, as it *could lead to no separation of concerns* and an *excessive yielding to the claims of a defaulting*

customer.
All these aspects were usually *less likely to be verified with more structured contracts.* Conversely, *organisations with more employees* (usually >20) were able to propose more structured and complex contracts. Generally, these agreements were standardised as *templates*, which were the results of previous experience and the assistance of lawyers. Templates were usually *updated only when critical cases were discovered or a bad situation occurs.*
The *interviewees* who declared to adopt this approach reported being *satisfied with their templates.* However, they would have been *interested in automating the contract signing and management processes.*

2.2.8 Selection of the problem

The criteria for the selection of the object of interest took into consideration

- relevance for the majority of the suppliers that were interviewed. Suppliers must work (at least in part) on bespoke software projects
- feasibility of a noteworthy contribution in a period of six months

As stated in paragraph 2.2.7, **the choice fell on problem P2**. At the beginning of the selection process, I considered each problem equally worthy of analysis.
The choice was mainly driven by the feedback that I received from the various interviewees and by my personal experience with the topics related to each problem. At the end of the interview, I asked every interviewee to rank every problem according to the perceived impact on his/her organisation. *80% of the total responses put in first place P2 as the most relevant problem* for their organisations.

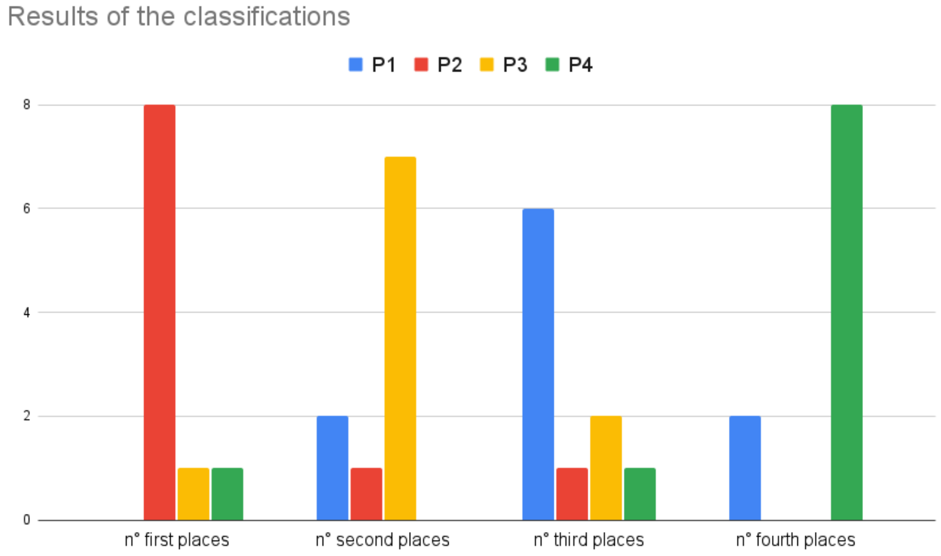


Figure 2.5: Results of the classifications of the problems made by the interviewees

I also **asked the interviewees to discuss each choice** to obtain a valid proof of the rationality behind their decision making.

P1 - Selection of the appropriate software development life cycle

For what concerned P1, interviewees proposed two main arguments. The former pointed out that suppliers would be interested in financing training activities about SDLC models, but the required *effort to properly train their employees would be unsustainable* in terms of money and time and would require many years of practise. The latter instead argued about the actual advantages linked to the conforming to different models outside the already adopted ones. In fact, most of the interviewees claimed that *there would not be a significant return on investment* (other than personal knowledge and perhaps some little improvements in the way of working) in putting effort in this objective. According to them, in most cases, *customers would not be able to perceive this feature as a valuable factor* in supplier selection. Consequently, customers would not be inclined to make larger economic investments in order to avail of this service.

P3 - Improvement of the estimation of the contingency factor in quotes

The choice not to undertake the analysis of P3 matured because of both interviewees' feedback and personal considerations. This problem was the second best choice among the available ones, since in 70% of the cases it was placed second in the ranking made by the interviewees. They mainly argued about the fact that *this analysis should be specific for every project.* Furthermore, many *parameters* that might be essential for this scope *might not be known or immediately available.* These parameters might also be highly influenced by poor requirement analysis or sudden changes emerging from external factors. I also decided to discard this problem for additional considerations. First of all, performing a proper analysis of the problem would have *required access to other organisations' private and sensitive data.* Therefore, the analysis would have been inevitably influenced by professional secrecy, non-disclosure agreements, and bias. Furthermore, my personal knowledge of economic and marketing studies that could have been involved in the analysis of this problem was not solid enough to perform such an analysis. Hence, I would have needed the *involvement of external collaborators.* This aspect would have required *more resources than the available ones.*

P4 - Analysis of the main critical issues and weaknesses of contracts between the two parties

P4 was perceived as the problem of less relevance to the interviewees. The main reason underneath this perception was leveraged on the fact that most of the interviewed suppliers already relied on existing templates. The interviewees stated that they felt *sufficiently protected by the agreements between acquirers and suppliers during the development of the project.* Furthermore, on average, they declared to consult contracts only a few times a year and make modifications on even fewer occasions.

Conversely, interviewees belonging to smaller organisations showed a greater interest in this topic. However, also in this situation, I would have needed the *involvement of external collaborators* and *access to other organisations' private and sensitive data.*

All these aspects, in addition to my major degree of expertise in the involved subjects (as a bespoke software development project manager), made P2 clearly stand out as the best object of interest to be undertaken for further analysis.

2.2.9 State of the art for problem P2

This paragraph describes in a more detailed way the results obtained by the other researchers and explains why they are relevant for this case study.
The current *state of the art* related to the analysis of customer-supplier issues and related topics provides many different studies. The analysis of customer-supplier interactions has been widely studied and still represents an interesting research topic. Among the large set of works related to this topic (or related ones), this research finds its foundations on four main studies, introduced in section 1.3.3.

A framework for identifying software project risks

The first relevant job for this case study, *"A framework for identifying software project risks"*, was presented by Mark Keil, Paul E. Cule, Kalle Lyytinen, and Roy C. Schmidt [28].
In this paper, the authors provided a framework for classifying software risks according to the perceived level of control exercisable by the project manager (low/high) in certain activities and the perceived relative importance of risk(low/high).
Specifically, the authors classify risks in a 2x2 matrix, placing on the two axes the perceived relative importance of risk and the perceived level of control exercisable by the project manager.

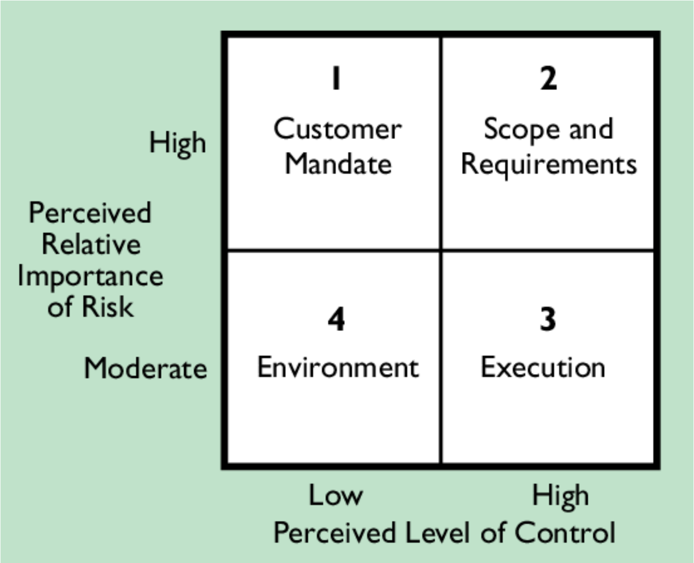


Figure 2.6: Risk classification matrix, *Source:* [28]

The following table provides an overview of the classes and the types of risks that are presented in this work

Category	**Description**
`Customer mandate`	*Risks that cannot be controlled (but can be influenced) by the project manager.* Project managers must take reasonable steps to ensure they have the support and commitment needed to properly deliver the project. The risk mitigation for Quadrant 1 involves relationship management, trust building, and political skills. Project managers must have these skills to effectively address these risks
`Scope and requirements`	*Risks related to the ambiguities and uncertainties that arise in establishing the scope and requirements of the project.* Strategies to reduce risk in Quadrant 2 involve managing ambiguity and change
`Execution`	The *risks* in Quadrant 3 concern the *actual execution of the project.* To effectively address risks in Quadrant 3, project managers must adhere to a well-defined development methodology and be proactive in predicting and responding to events that could jeopardise the development process
`Environment`	The potential dangers associated with this area can be attributed to the *environment of the project, both within and outside the organisation.* Risks in quadrant 4 are those in which the project manager has limited or no influence. They are unlikely to happen, so they are not seen as particularly important. However, if they do occur, they can be serious and hazardous

Table 2.8: Risk categorisation framework proposed in "A framework for identifying software project risks" [28]

Agile customer engagement: A longitudinal qualitative case study

The second relevant paper for this case study is "Agile customer engagement: A longitudinal qualitative case study" by Geir Kjetil Hanssen Tor Erlend Fægri [20]. This article discusses th*e benefit in the transition of a small organisation from waterfall-based methodologies to agile methodologies* (evolutionary project management). In this study, data collection was achieved through a series of interviews with both internals and customers of the organisation in a period of two years. This study showed how different aspects of a software project can be affected by the adoption of different software development methodologies. Among the various aspects that are discussed in it, this paper points out interesting aspects related to *customer engagement.* In particular, it highlights how there is no big difference in involving just one stakeholder on the acquirer side, as multiple ones will get involved anyway. Furthermore, *acquirers sometimes do not perceive any form of payment or compensation, and they are also not given any formal power either.* The sole reason to get involved as seen from the *acquirers' perspective* is the opportunity to *affect the course of development.* Therefore, they might perceive the *sensation of being outsiders* rather than playing an active role in the project.

The last two works considered for this thesis performed an analysis much similar to the one presented in this thesis. Both presented a qualitative research approach (grounded theory) to analyse customer-supplier issues in software development.

Customer-Supplier Issues in Software Development

"Customer-Supplier Issues in Software Development", analysed the perceptions expressed by different roles of three organisations that worked as custom software suppliers. This work identified three main categories of factors that can represent a threat to a software project [36].

Category	**Description**
`Knowledge transfer difficulties`	Includes aspects such as *insufficient domain knowledge* (situations where the customer is not able or willing to provide knowledge about the application domain to suppliers), *incompatible terminology* (incompatibility between the different vocabularies of software developers and experts in the application domain, such as users and customers), and *language barrier* (problems that are caused by limited knowledge in the used language)
`Change Management during the Project`	*Changes* may be difficult to be implemented due to deadlines, resource problems, and complex development environments. They may ultimately result in *poor quality and delayed projects whether their enforcement is unplanned* and therefore needs *extra time and resources* that are not given by the customer
`Customer dominance`	Includes aspects such as *customer dominance in the process* (a situation where the customer forces the alignment of development processes in the supplier team and the software development process becomes disturbed) and *business-induced power asymmetry* with which we refer to the supplier's fear of losing business (situations in which project teams accepted customer demands as such because they were afraid of losing the customer)

Table 2.9: Main critical customers' issues in software development according to Pasi, Smolander and Nikula [36]

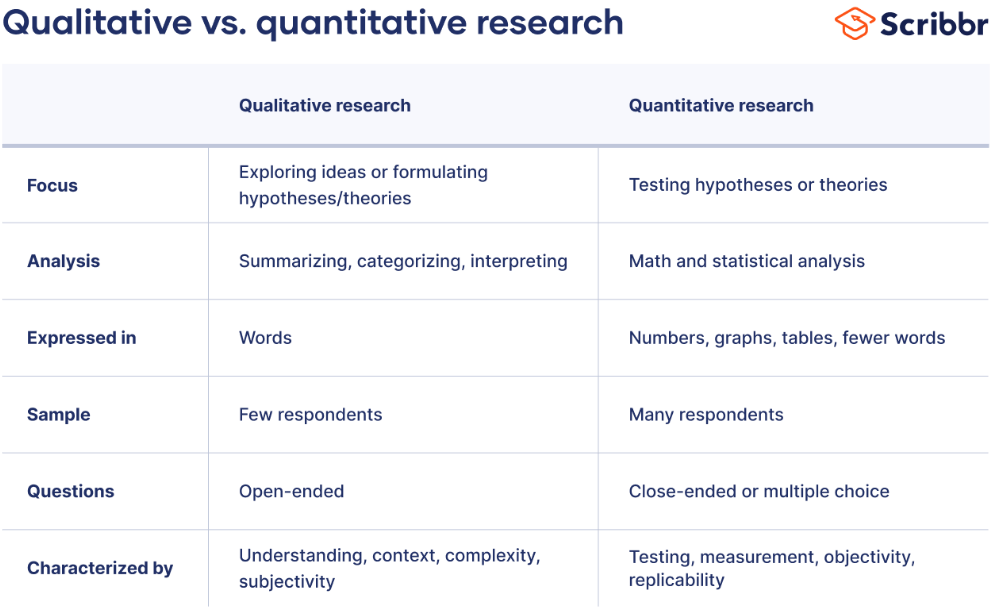
Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers' Perspective

"Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers' Perspective" provides a similar approach to the previous paper, contextualising the research to the Norwegian market.
This paper reports most of the issues identified in the previously discussed work, increas-

ing the set and reported additional threats induced by customer-supplier relationships towards the project.
In particular, the study extends the set of **threats** by adding *excessive organisational hierarchy in public organisations, contracting issues, scope aspects, personal egos, financial issues, and the selection of an improper team.*
According to the author, the **consequences** that follow include *decreased productivity, wastage of time and resources, diverted attention from project objectives, loss of motivation, poor decision making, and loss of communication between stakeholders* [44].

2.3 Selection of the research process methods

The choice of problem P2 concluded the *selection of the topic to be investigated.* The next step involved the selection of the type of research method to be adopted to investigate the problem.
Research methods can be classified into two main categories: **qualitative and quantitative research** methods.
The former is expressed in words, and it is used to study concepts, experiences, or thoughts. It allows the *gathering of in-depth insights on topics that are not yet well understood.*
The latter is expressed in numbers and graphs, and it is used to confirm or confute theories. Quantitative research methods can be used to establish *generalisable facts about a topic* [26]. The following picture highlights the main characteristics of the two categories



Qualitative vs. quantitative research

Scribbr

	Qualitative research	Quantitative research
Focus	Exploring ideas or formulating hypotheses/theories	Testing hypotheses or theories
Analysis	Summarizing, categorizing, interpreting	Math and statistical analysis
Expressed in	Words	Numbers, graphs, tables, fewer words
Sample	Few respondents	Many respondents
Questions	Open-ended	Close-ended or multiple choice
Characterized by	Understanding, context, complexity, subjectivity	Testing, measurement, objectivity, replicability

Figure 2.7: Qualitative vs quantitative research methods, *Source:* [26]

The study of P2 can be conducted with both the presented categories. But many reasons led me to **opt for the qualitative research methods**.
First of all, one of the objectives of this thesis involved the *formalisation of unnoticed or underrated recurring problems in customer-supplier relationships.* Therefore, starting from the baseline given by the state of the art, I decided to try and investigate deeper into these problems. Hence, a qualitative analysis was more suitable for this purpose.
Moreover, during the first two phases of the interviews, I noticed that the *interviewees were unable to provide precise data* on some parameters that might be necessary to perform any kind of analysis. Extracting and elaborating basic data (e.g. the average duration and costs of the projects) are non-trivial operations and their communication to people that do not work in the organisation could be also biased. In addition to these considerations, the number of parameters that might be taken into consideration in this type of analysis might be quite large. Therefore, their correlations might not be easily identified.
Quantitative research also needs large data sets in order to provide meaningful results[23]. The time required to gather large data sets from various organisations was not compatible with my current resources.
In conclusion, many studies that analysed similar problems adopted a qualitative research method. This recurring path led me to assume that other researchers could have applied the same reasonings in selecting the type of analysis to be conducted.
In contrast to the other studies, *what this study wants to bring is an approach that also takes into consideration the acquirer's point of view.* Therefore, I decided to observe the problem from both the perspective of acquirer and supplier in order to enrich my theories with the expression of the two parties.
As reported by Amsterdam UMC [12], Frambach et al (2013) offers a useful overview of different quality criteria and strategies to be adopted in a qualitative analysis [16]

Criteria	**Strategies**
`Credibility / Internal validity`	• Use multiple data-analysis methods (methodological triangulation) • Ask participants' feedback on the data or interpretation of the data (member checking)
`Transferability / External validity`	• Make the findings meaningful to others by describing them and their context in detail (thick description) • Discuss the resonance of the findings with the existing literature from different settings

`Dependability / Reliability`	• Continuously analyse the data to inform further data collection (iterative data collection) • Continuously re-examine the data using insights that emerge during analysis (iterative data analysis)
`Confirmability / Objectivity`	• Search the data and/or literature for evidence that disconfirms the findings • Discuss the findings with peers/experts (peer debriefing) • Keep a diary to reflect on the data analysis (reflexivity) • Document the steps and decisions taken in the data analysis (audit trail)

Table 2.10: Criteria and strategies for a good qualitative analysis according to Frambach et al (2013) [16]

The current state of the art related to the qualitative research includes six main methods [53]. This thesis only reports the three most structured ones, which could be useful to analyse P2.

2.3.1 Phenomenological Method

Phenomenology is described as the *study of structures of consciousness as experienced from the first-person point of view.*
The central structure of an experience is its intentionality, its being directed toward something, as it is an experience of or about some object. An experience is directed toward an object by virtue of its content or meaning (which represents the object) together with appropriate enabling conditions [45].
Phenomenology uses methods such as *observation and interview to try to find out why people make the decisions they make outside of purely logical reasons.* Phenomenological research tries to investigate actions that might not be easily perceived as rational. It aims to *investigate other contributing factors that influence their subconscious decision-making.* Furthermore, this method is designed to analyse the feelings of the participants towards the object of interest. This approach requires researchers to set aside their prejudices and a priori assumptions and focus primarily on the immediate experience. Phenomenology studies the feelings and instincts expressed towards the objects of interest of the study (this methodology is often applied to brand and campaigns).
Phenomenological research is therefore not always straightforward. Yet, this can be one of the most important types of qualitative research in understanding how consumers and users actually make decisions.
The phenomenological research design is descriptive. The researcher aims to describe the structure of a phenomenon as accurately as possible. It requires the researcher to first describe the lived experiences objectively and then reflect on the description with

reference to the existing theories about the phenomenon.

Researchers can use a variety of methods in phenomenological designs. Some of the most common methods include participant observation, interviews, conversations with participants, analysis of personal text, action research, and focus meetings.

Regardless of the method involved, the researcher must focus on the research issues and avoid influencing the participants. In addition, he/she should show empathy and establish a good level of relationship to gain deep insight into the experiences of the participants.

There is *no standard way to carry out a phenomenological inquiry.* However, research involving phenomenological design often follows this pattern [37]

1. Identification of the phenomenon
2. Development of a detailed description of the phenomenon
3. Bracketing personal prejudices and a priori assumptions
4. Collection of data from the participants: the form of data collection can be interview, observation and surveys among other forms. This data capture will often have to be in the moment, capturing how people feel at the point of sale or action. In fact, if they are asked later, the reasons given might change into more logic-based ones
5. Data analysis: this phase usually involves reading the data, demarcating the data, eliminating irrelevancies, grouping and naming the data into constituents and organising the data into themes that accurately and fully describe the participants' lived experiences.
6. Development of a composite description of the phenomenon
7. Presentation of the description

2.3.2 Ethnographic Model

Ethnographic research is a qualitative research method that involves the *systematic study of people in their natural environment.* The objective of this type of study includes understanding their *way of life with respect to how they see and interact with the world around them.* Another intention of an ethnographic study involves *the production of a rich and comprehensive account of a social setting from the point of view of participants.*

In applying this method, *researchers spend time with the participants.* They gain insights into their social interactions, and collect data through participant observation, face-to-face interviews, etc. Ethnography in qualitative research is also known as thick description, as it involves a close observation of the participants and a detailed description of their cultures, behaviour, mutual differences, and practises.

Researchers using the ethnographic research model look to observe how people act, or how they use a product rather than how they might self-report they use it.

If phenomenology is studying why people make decisions, ethnographic research will provide information on how they then use the product or service.

Often, people might say they use something one way, but observation shows a different story, especially when people are observed over extended periods of time.

In ethnographic research, the researcher must act as a fly on the wall. This might be achieved via a viewing booth, the skill of the agency making participants feel at ease,

creating an environment in which they provide genuine reactions.
Ethnographic methods are the techniques used to collect data for an ethnographic study. The three main **ethnographic methods** are *participant observation, interviews, and archival research.* Typically, researchers use all or a combination of all of these ethnographic methods to collect data for their study.

Ethnographic method	**Description**
`Ethnographic observation`	There exist two types of ethnographic observation: *active and passive participant observation.* T*he former involves becoming a member of the study group*, participating in their daily activities, and working directly with them to gain hands-on experience and understand their perspective. *The latter involves only observing and taking notes.* You do not participate in the activities of the group
`Interviews`	Interviews in ethnographic fieldwork involve observing participants in their natural environment and asking them questions to gain more insights. Researchers interact with the study group to understand the actions and thought processes of the participants. In this way, the researcher gets to ask questions directly related to the experiences of the participants. These insights might not be revealed if the interview occurred outside of the natural environment of the subject
`Archival research`	Archival research is an ethnographic method in which you *collect and analyse existing research data from different sources* (e.g. websites, statistical research, reports, etc.). Archival research helps the researcher to *gather more preliminary information on the participants*' demographics, economic status, educational levels, cultures, interests, etc., before conducting ethnographic fieldwork

Table 2.11: Ethnographic methods

An ethnographic research follows these steps [14]

1. Set the objectives of interest that will be targeted by the research and identify the questions to be asked during ethnography study
2. Choose the best ethnographic method (or a mix of them) to collect the needed data
3. Identify and contact the potential study subjects
4. Perform the ethnographic fieldwork
5. Analyse the obtained data. Coding ethnographic data involves identifying patterns and themes in the collected data.

6. List all the insights uncovered, including answers to the research questions.
7. Redact the ethnography. An ethnography is a detailed and comprehensive description of the research subjects, the insights discovered and how the study was conducted.

2.3.3 Grounded Theory

Grounded theory is a qualitative approach that aims to *discover new theories that are based on the collection and analysis of real-world data.* It is an *inductive approach* in which new theories are derived from the data. This approach is opposite to hypothesis-deductive ones, which are aimed to come up with a hypothesis to be proved/disproved. In grounded theory, the process of data collection, data analysis, and theory development takes place in an *iterative process.* Iterative data collection and analysis occurs until theoretical saturation is reached. **Theoretical saturation** is the turning point at which *additional data do not add additional insight into the new theory.*

The *usage* of grounded theory is generally suggested in projects whose scope is related to discovery of such things as *social relationships and behaviours of groups.* The research may start with either a question or just a quantity of research data. This would then be studied, and theories would be allowed to emerge. Emerging theories could be grouped, and then further analysis can be developed. Grounded theory is generally used as a research method that builds on previous work, demonstrating the joined-up nature of qualitative research. Data are collected through *interviews, observation, surveys, and questionnaires*, *regardless of the mode of communication* used (in person, by phone, or online).

The inventors of grounded theory, Glaser and Strauss [18], emphasised two main concepts: **theoretical sampling** and **theoretical sensitivity**.

The former is used to produce more data to endorse or refute the categories that have been identified in the previous analysis. It is the *process of making conscious decisions about what further detail they feel needs exploring as the new theory develops.* It usually occurs after some initial key concepts or categories have been identified.

The latter refers to the insight of the researcher. It concerns the *researcher in being able to give meaning to the data, understand what the data say, and be able to separate out what is relevant and what is not.* By being theoretically sensitive and using insight, the researcher can develop a theory that is grounded, theoretically dense, and cohesive. Sensitivity comes from several sources including literature, where in-depth reading offers a rich understanding of the phenomena being studied; professional and personal experience offers an understanding of the events and topics being explored; and the analytical process allows insight and understanding of the phenomena.

It is important to note how *data analysis happens at the same time as data collection.* In grounded-theory analysis, this is also known as **constant comparative analysis**. Data analysis is achieved through three main steps [38]

Step	**Description**
`Open coding`	A **code** is defined as a *tag/label that is assigned to excerpts of texts.* In particular, codes are used to encapsulate excerpts of data that represent and group the same central idea or concept. Open coding is the phase of the analysis in which codes are generated

`Axial coding`	Axial coding is used to *compare codes with other codes.* In particular, the aim of this phase is to *create categories.* A **category** is defined as a *connection between multiple codes*
`Selective coding`	Once theoretical saturation is reached in codes and categories, selective coding is performed. With selective coding, you *connect all your codes and categories together under one core category.* This core category represents the *central thesis of the research* and is the core idea behind the theory. This core category can be an existing category that was derived earlier, or it can be a new category derived from all other existing categories so far

Table 2.12: Stages of the analysis in grounded theory

The following table reports the pros and cons of using grounded theory [60]

Pros	**Cons**
No need for hypothesis Researchers do not need to know the details about the topic they want to investigate in advance, as the grounded theory methodology will bring up the information	*Theoretical sensitivity is influenced by knowledge base* If a researcher does not know enough about the topic being investigated, then their theoretical sensitivity about the meaning of the data may be lower and the information may be missed if it is not properly coded
Flexibility Researchers can take the topic according to their perceptions, based on what the data tells them. This means that exploration avenues that may be off-limits in traditional experimental research can be included.	*Large topics take time* There is a significant time resource request to properly conduct research, evaluate the results, and compare and analyse each excerpt. If the research process finds more avenues for investigation, the time needed to explore them increases.
Data-first The grounded theory is based on data analysis in the first place, so the conclusion is based on information that has strong data behind it. This could be seen as having more validity.	*Qualitative research is harder to analyse than quantitative data* Unlike numerical factual data from quantitative sources, qualitative data are harder to analyse, as researchers will need to look at the words used, the sentiment and what is being said.

Multiple stages improve the conclusion Having a series of coding stages helps refine the data into clear and strong concepts or themes	*Bias in interpreting qualitative data* As the researcher is responsible for interpreting the results of the qualitative data analysis and putting his/her own observations in text, there can be bias of the researcher that could skew the data and possibly affect the final grounded theory.
	Not repeatable Although grounded theory can present a fact-based hypothesis, the actual data analysis from the research process cannot be easily repeated because opinions, beliefs, and people may change over time. This may affect the validity of the grounded-theory result.

Table 2.13: Pros and cons of grounded theory

2.4 Selection of the qualitative research method

The **qualitative research method** used in this project was **grounded theory**. Beyond this choice there were three main reasons. First of all, this project aimed to *identify a set of traits of acquirers' behaviour that could cause issues in the development of the project*. This was not the first research conducted on this topic. Therefore, this work could confirm/confute some existing theories or provide contributions to the state-of-the-art. Grounded theory was the most suitable method for this scope. Furthermore, the *phenomenological and ethnographic method* posed two preconditions that could not be easily satisfied with the available resources. The former, in fact, *required to gather data at the point of action*, while the latter required to *spend time with their participants* and observing them. These two aspects would have required to be in the actual location of the interviewees (in a determined instant). Moreover, these conditions would have limited the data set to only organisations willing to participate in the research in a nearby location. Therefore, the information could have been easily biased by the cultural factors of the territory.

Finally, grounded theory provided a more structured approach and was deemed valid and suitable by other researchers in order to perform similar analyses.

As stated above, this work took into account both the perspectives of acquirers and suppliers. To perform data sampling, I iterated again on the supplier's questionnaire and designed another specific version for the acquirer.

2.4.1 Supplier's side

For this specific round of supplier interviews, I also set a **list of parameters** that should have been *taken into consideration during the selection of prospects.*

- age of the interviewee
- site location
- organisation dimensions (in terms of employees/stable collaborators)
- presence of multiple sites
- achievement (and eventual type) of certifications
- type of customers (start-ups, small organisations, large organisations, IT organisations)
- technical expertise (mobile, web, IoT, etc)
- sectors of expertise (healthcare, commercial, etc)
- type of services offered (landing pages, e commerce, web apps etc)
- location of their customers
- interviewee's level of experience in developing projects
- adopted frameworks, methodologies and tools
- ways of working with the customers (e.g. dedicated team vs part-time team, work in separate locations vs working directly on the customer's headquarter)
- average time a team can dedicate to project (or to a customer, according to the stipulated contract)

In order to schedule these interviews, I adopted *two different ways of engaging prospects*, the former including *just a presentation mail* while the latter *provided a direct contact through telephone calls.*
In general, *the second approach performed better*, as direct contact allowed me to provide major useful data for the selection of the correct person to be interviewed.
For this phase I contacted 121 European suppliers (approximately divided into 55% Italian suppliers and 45% coming from other European countries). At the end of this phase, the set of suppliers who were effectively interviewed counted 28 suppliers, so the conversion rate interviewees/prospects was approximately 23%. Some interviews were held in presence, but the majority was performed in video calls.
For this specific data sampling, *I also considered as relevant data gathered from the second set of interviews* (just the part referred to profiling and P2). Then that part was used as a *baseline* to make *conscious decisions about the further details to be investigated.*
There have been different iterations of these decisions (not every iteration will be reported), often updated after interviews or discussions with peers.

In the end, the results of this iterated decision-making process translated into the following of the questionnaire.

- Interviewee profiling (unchanged)
- Investigation of interest towards problem P2 (unchanged)
- Client relationship management

The questions reported in sections 2.2.3, 2.2.5 and 2.2.6 were excluded from the new version of the questionnaire for reasons related to the duration of the interview.

2.4.2 Client relationship management

This part of the questionnaire focused on the various customer-related issues that were progressively emerging from the previous interviews.
The latest version of the newly introduced section contained the following set of questions

ID	**Question**
Q19	How do you usually acquire new customers?
Q20	Do you usually give deadlines and assignments to your clients? When and how do you present them?
Q21	How are phases and activities that are not purely about implementing functionalities (e.g. analysis, testing, documentation) perceived by your customers? Are they willing to pay for them?
Q22	How important is documentation for your organisation? Which project phases require the most time to write related documentation?
Q23	How do you manage changes during construction (especially on budget projects)? Do you make change requests?
Q24	Do you generally specify which stakeholders should be involved in the project? What aspects do you take into account?
Q25	How is the quality control of the application managed? How is the acceptance of requirements handled? What are the activities of the testing phase that require more time for your organisations (definition, documentation, implementation)?
Q26	Do you generally track project costs and relate them to the respective progress of the project? How? Is the customer aware of the trend of these metrics?

Table 2.14: Questions included in the analysis of client relationship management

The questions take into account a *subset of crucial processes* described by the ISO, in which the involvement or will of the customer is relevant. In particular, this analysis aims to understand which are the main problems that suppliers report to experience in trying to apply these processes in bespoke software development

- Supply process
- Life cycle management process
- Quality management, verification and validation processes
- Project planning process
- Information management process
- Measurement process
- Stakeholder needs and requirements definition process
- System/software requirements definition process

Q19 aimed to explore the possibility of *relating the level of expertise in the IT sector owned by a customer to the marketing channel* through which he/she contacts (or perhaps is contacted by) the supplier. The set of channels may include websites, social networks, word of mouth, outsourcing of projects made by larger organisations, fairies, and events, etc.
Q20 and Q24 were intended to analyse two aspects of the suppliers' communications with their customers. In particular, the object of the analysis was the *way of working of the suppliers in the assignment of tasks and concerns to their acquirers.*
Q21 was the central focus of the third set of interviews. This question aimed to grasp the *phases and activities of the project that are perceived as less valuable by their customers. Q22, Q25, and Q26 were added by applying theoretical sampling* on the results of the second set of interviews and the responses to Q21. They focus each aspect that was reported by the interviewees to be shallowly or not considered by their clients. In particular, most of the interviewees reported experiencing *severe issues in explaining the need to perform tests and write project documentation.* Furthermore, some of their *customers did not track costs or were too optimistic about the status of the work*, overestimating the percentage of completion of the project. In conclusion, Q23 wanted to explore the perception related to *change requests* when waterfall-based methodologies are applied.

2.4.3 Acquirer's side

The previous studies considered in section 2.2.9 mainly focused on the supplier's perspective.
This work instead also tried to consider the point of view of an acquirer as a proof and enhancement of the observations reported by the suppliers.
It is important to underline that this activity was much more complex and hindered by different factors that will be analysed in section 4.2.
In order to conduct this study, another questionnaire was designed to be submitted directly to acquirers.

The ideal prospects for this set of interviews should have shown *at least one* of the following criteria.

- They **commissioned a software project for the first time**
- They generally showed **little commitment towards the project**
- They **did not understand the value of all the activities** that were carried out (e.g. test, documentation, analysis)
- They had **difficulty using tools/following project steps**

For this specific analysis, *I asked the suppliers I interviewed to get in touch with their customers.* I also contacted some of the customers of my organisation to obtain a richer data set. I did not contact the customers of other suppliers directly, as this approach would have not ensured compliance with the previous criteria.
At the end of the project, the set of interviewed customers counted 9 interviewees. Also this questionnaire went through a series of iterations that produced two main versions.

1. Acquirer's profiling
2. Analysis of customer-supplier interactions during the project
3. Analysis of the complexity perceived by the customer in performing the assigned activities

Every section of this questionnaire will be presented in the following paragraphs.

2.4.4 Acquirer's profiling

This part of the interview aimed to highlight the main characteristics of the acquirer that was undertaking the interview. The main interest related to this set of questions involved the analysis of the *correlation between the characteristics of the acquirers and their needs and expectations.*
The profiling section contained the following questions

ID	**Question**
QA1	Have you ever commissioned software development projects from any supplier? What kind of projects have been developed?
QA2	How long did the project last? Is it still in development or maintenance? Are you satisfied with the development of the project?
QA3	Which software development model/contract did you use to develop the project? Are you satisfied with the model/contract that was adopted? Did the supplier provide training on how to work for the project? In your opinion, was that enough or would you have preferred to be followed more during the work?
QA4	Can you provide an indicative range for the budget related to the project? If so, was it breached or maintained?

QA5	Which parameters did you consider when selecting the supplier's organisation? What skills were you looking for?
QA6	Which results were the most satisfying for you and what skills did you appreciate?
QA7	Which results were the most unsatisfactory to you? Did you face critical issues?

Table 2.15: Questions included in the acquirer profiling section

QA1 and QA2 aimed to *assess the previous experience of the interviewee in bespoke software development* and the complexity of the commissioned work.
QA5 responses represented a *proof of the interviewees' expertise*, as it is reasonable to assume that more experienced customers would have provided more *technical criteria used for the choice.*
The next step of the investigation focused on the offered SDLC models. The intention behind QA3 involved the *analysis of the choice of SDLC and the awareness behind it.* Additionally, QA3 evaluated the *presence of training related to the topic provided by the supplier* and the level of satisfaction related to it.
QA4 aimed to confirm / refute the impressions provided by the previously interviewed suppliers about the tracking of costs during the project.
In conclusion, QA6 and QA7 were deliberately open-ended. The former aimed to *spot possible techniques or skills shown by suppliers and appreciated by customers.* The latter instead focused on identifying eventual problems not declared by the interviewed suppliers.

2.4.5 Analysis of customer-supplier interactions during the project

This section of the questionnaire explored the context of the *involvement of the customer in his/her own project.* This research was carried out through a set of questions that refer to customer-supplier interactions during project development.

ID	**Question**
QA8	Was the project an essential need for you or was it an idea you wanted to carry out as a parallel project to your main activities/an opportunity for non-essential improvements?
QA9	How many hours a week were you able to dedicate to the project? How many hours on average did the supplier require you?
QA10	How often did you communicate with the supplier during development? Did the meetings take place on a regular basis?

QA11	Were the activities and requests made by the supplier clear? Were the deadlines feasible or too strict for you?
QA12	Compared to what was established at the beginning, have you made any changes to the requirements? If so, how often? Were the changes particularly large or rapid?
QA13	How were the tests handled? Were they automatic? Did the supplier ask you to test the increments? With what timing? Did you generally find errors/non-compliance? How often?
QA14	Was the commitment required by the supplier constant throughout the project or did it vary according to the phases of the project?

Table 2.16: Questions included in the analysis of customer-supplier interactions during the project

QA8 contextualised the project in terms of the *interviewee's perceived need.* A project that represented the core business or critical improvement should be perceived by the acquirer as important and worthy of interest.

QA9 focused on two important aspects of the project. The former included the *perceived amount of time per week required to participate in the project.* The latter instead aimed to *explore whether the interviewee was aware of the time effort required by the supplier* (if specified).

QA10 and Q14 focused on the *overall effort required during the project phases*, to the extent of identifying the eventual phases in which the customer was less involved (and linking them to the corresponding adopted SDLC model).

QA11 analysed the perceptions that the interviewee felt about the activities and requests made by the supplier. In particular, it aimed to *identify the presence of eventual known problems*, such as language barriers or unrealistic previsions.

QA12 investigated the perceptions of the interviewees about *what was considered a change request for them and what was not.* Customers (especially those with low expertise in software project development) may in fact not perceive all the requests they make as changes. Additionally, they *may not be aware of the impact of changes* on the project.

In conclusion, QA13 investigates the type of tests performed on the product, *the awareness of the interviewee related to automatic tests, and his/her attitude towards the performance of acceptance tests.* In fact, suppliers reported that part of their customers did not perform any tests in a staging environment. Suppliers also had to cut automatic tests due to budget constraints.

2.4.6 Analysis of the complexity perceived by the customer in performing the assigned activities

This section investigated the interviewee's perceived complexity towards assigned activities.

ID	Question
QA15	What were the project activities that you found most difficult to follow?
QA16	Were you generally able to follow all project activities and deadlines?
QA17	Do you have an IT department within your organisation? Who made the decisions about the project? Were there intermediaries or does the person actively work directly on the project?

Table 2.17: Questions included in the analysis of the complexity perceived by the customer in performing the assigned activities

This last set of questions wanted to highlight *factors such as difficulties in completing the assigned tasks* (e.g.account managing and materials provisioning) within the established deadlines.
In conclusion, Q17 aimed to investigate the *correlation between project failures and the absence of stakeholders that could facilitate task completion* on the acquirer's side. Additionally, the last part of the question was intended to understand the *relationship between project failure and the absence of direct involvement of the main contractor* (decision maker) in every phase of the project.

In conclusion, the first part of this chapter aimed to explain the steps, the methodologies, and the reasons behind the need to select one of the four problems identified in section 1.3.4.
The rest of the chapter is dedicated to the review of the state-of-the-art related to problem P2 (Analysis of customer-supplier interactions), the methodologies and the tools used to perform this specific analysis.
Chapter 3 will be dedicated to the application of grounded theory (the qualitative method selected for the analysis of problem P2) and the analysis of the achieved results.

Chapter 3

Results of the research

This chapter describes the results achieved in applying the principles of grounded theory to analyse P2. Every step prescribed by the method is described. Additionally, the second section provides a comprehensive description of the results that were obtained. In conclusion, the last section discusses the results obtained.

3.1 Application of the grounded theory

The previous chapter described the steps taken to perform a selection between the described problems, the tools, and the methodologies used to make the two main choices that led to the selection of the final problem to be investigated.
The comparison of the research methods led to the selection of **grounded theory** as the appropriate choice to be made. This inductive method allows for derivation theories based on the collected data through constant comparative analysis upon reaching theoretical saturation.

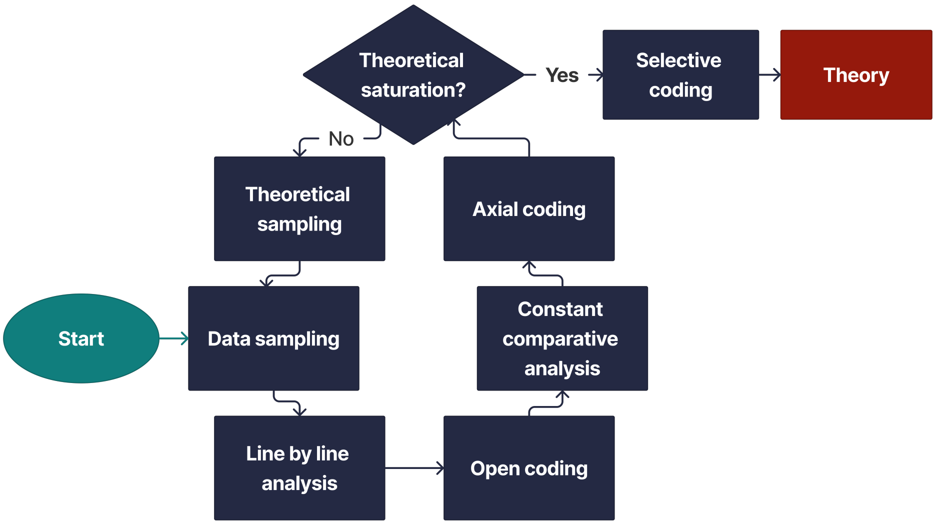


Figure 3.1: Phases of the grounded theory research method

The following paragraphs describe the results of each step provided by the methodology.

3.1.1 Open coding

Open coding is the first step of the data analysis process provided by grounded theory. This step requires the researcher to *identify codes*. **Codes** are defined as *tags/labels that can be assigned to excerpts of texts*. In this case, the objects of study are represented by the transcripts of the second and third sets of interviews (including customers' ones). During open coding, data are interrogated by line, sentence, or paragraph, or as a document in its entirety, using comparisons and asking the following questions (Holton [22]; Strauss & Corbin [47])

- What is this observation (event), sentence (idea), or paragraph (process) a study of?
- What is actually happening in the data?
- What is the main concern?

In order to extract codes, I compared the various transcripts by reporting the responses provided by the interviewees to the questions in a table. For the second set, only the questions related to interviewee profiling and investigation of interest towards problem P2 were considered, to the extent of consistency between the two sets. The following table reports some examples of codes identified during open coding

Code	**Excerpts**
`Word of mouth`	• *Some acquaintances spoke well of the organisation*, and I kept it in mind in order to perform the selection • *We mostly rely on word of mouth* to acquire new customers • *Customers generated by word of mouth* have quite different levels of expertise in the IT sector • A good part of our *leads is generated by word of mouth* • The majority of our new clients come because our *current or previous ones introduced us in a positive way* • I got in touch with the organisation because *a friend of mine*, who commissioned a project to the organisation before, was satisfied of the obtained results and he *suggested me to work with them* • We organise events in which our customers and some prospect are invited. In this way, we enhance the chances of *being introduced in a positive way to potential new customers.*

Late materials delivery	• 95% of the time *the customer is late on the delivery of the material* • *Contents (e.g. images and texts) are never delivered within the defined deadlines* • It often happens that the *customer does not deliver the materials in time* • More than sometimes *I had to wait more than a month to get the credentials* of some accounts • We experience delays when we need to receive materials from other stakeholders (customers or other suppliers involved in the project)
Turnkey (fixed price) contracts	• Customers that require *turnkey contracts* are generally more likely to be dissatisfied at the end of the project • We have had projects at a loss or on an equal footing. This in particular with *turnkey contracts* • I wish I had more flexibility but the *fixed price contract* did not allow so • We overran the budget because of the changes, *we had a fixed price contract*
Waterfall-based methodologies	• We tend not to apply *waterfall-based methodologies* anymore • In some cases we apply *waterfall-based methodologies* but we tend not to • In general I always suggest using agile and never *waterfall* because of my previous experiences • I always try to persuade customers to not using *waterfall-methodologies* • There were a few cases in which we successfully applied *waterfall-based methodologies* • We apply *waterfall-based methodologies* in cases where the situation of the customer related to the budget requires so

Table 3.1: Examples of codes identified by performing open coding

Open coding helps in *breaking data down into discrete parts* that will form the basis of the analysis. Performing open coding allows for a progressive focus on the relevant aspects of every excerpt, but this phase alone is *not sufficient to derive a theory.*
The next step, axial coding, helps the researcher move in this direction.

3.1.2 Axial coding

Axial coding is the second step in coding in grounded theory. Unlike open coding, the objective of axial coding is to draw *connections between multiple codes.* These connections are referred to as **categories**. With axial coding in qualitative research, codes and their underlying data are analysed to find how they can be grouped and abstracted into categories. Categories could be created by abstracting out an existing code or developing new concepts that encompass several different codes. It is also possible to define a hierarchy of categories[25].
Corbin and Strauss provided a way to identify subcategories that will be used to build categories[47]. Their paradigm is based on three main features used to describe subcategories that build up a category: conditions, actions-interactions, and consequences or outcomes.

Conditions

Conditions are *sets of events or happenings that create situations, issues, and problems related to a phenomenon and, to some extent, explain why and how persons or groups respond in certain ways.* Conditions might arise from time, place, culture, rules, regulations, beliefs, economics, power, or gender factors, as well as the social worlds, organisations, and institutions in which we find ourselves, along with our personal motivations and biographies.

Actions/interactions

The second feature, **actions/interactions**, includes *routine or strategic responses made by individuals or groups to issues, problems, happenings, or events that arise under those conditions.* Actions/interactions are represented by the questions by whom and how. They denote what goes on among individuals, groups, organisations, and typical scenarios that depict the situation. In addition, they include aspects such as discussions and negotiations and other types of talk that occur in group situations.

Consequences

In conclusion, **consequences** are *outcomes of actions/interactions.* Consequences are represented by questions as to what happens as a result of those actions/interactions or the failure of persons or groups to respond to situations by actions/interactions, which constitutes an important finding in and of itself.

The *second set of interviews* led to a clear definition of a group of subcategories related to customer-supplier issues in bespoke software development: *customers delays in delivering materials, customers delays in performing tests, changes introduced by the customer, insufficient training on the methodologies applied by suppliers, poor or no knowledge in software development shown by the customer, related lack of definitions of precise deadlines, insufficient resource allocation, lack of contract clauses application,*

shallow requirement definition, late involvement of key stakeholders, little attention to contracts shown by the customer, excessive tolerance towards change requests with turnkey contracts, failure to apply the terms of the contract.
The application of **theoretical sampling** brought some changes and *additions in the categories set* and allowed further details to be explored. As the analysis proceeded, the questionnaire started to focus on these issues. As a result, the list of questions in client relationship management was obtained. Another result of theoretical sampling was the *exploration of the acquirer's side* through the dedicated questionnaire presented in sections 2.4.4, 2.4.5, and 2.4.6. The resulting set of categories with related subcategories is reported in the following table

Category	**Related subcategories**
`Customer dominance`	*Conditions:* business induced power asymmetry, customer dominance on process, perceived risk of involvement in legal case, need to make a good impression on the customer due to word of mouth, market competitiveness, inappropriate communication channels abuse *Actions (supplier):* slow down the development process when the acquirer is late, provide discounts, use inadequate channels to contact the acquirer, insert specific clauses that prevent them to respect deadlines if the customer is late, refuse to enforce clauses, reallocate developers *Consequences:* wastage of resources, reduced earnings, potential loss of relevant information, frustration
`Improper change management`	*Conditions:* continuous changes to graphical aspects for no functional reason, core product scope changes intercepted too late, changes related to the dissatisfaction of uninvolved stakeholders, unspoken needs *Actions (supplier):* require to pay (only) for major change requests, provide discounts *Consequences:* exceeding of budget and deadlines, dissatisfaction of customers, reduction of the profit margin, scope creep

`Knowledge transfer barriers`	*Conditions:* acquirer's lack of experience in software development, prohibitive learning curves for the acquirers, insufficient domain knowledge, inability of defining requirements, lack of understanding of the value provided by certain activities, lack of communication among stakeholders on the acquirer's side *Actions (supplier):* provide training on the various topics, hide the estimate for for activities not valued by the acquirers, passively accept the situations *Consequences (supplier):* loss of information, misunderstandings, acquirer's unawareness of the quality of the product, estrangement of some stakeholders
`Negative factors of customers' attitude`	*Conditions:* acquirer's progressive loss of interest towards the project, waterfall model adoption, defensive attitude over cooperation, passive attitude and contribution, urgency driven requests, lack of interest in the state of project, lack of time-effort definition for the acquirer, not way for the acquirer to monitor project progress and costs *Actions (supplier):* constantly ask the acquirer to test the application, incorporate tacit acceptance clauses in the contracts, negotiate discounts upon being threatened with lawsuits, evaluate the profile of the customer before accepting the work, provide an internal tool to promote visibility of project progress and costs *Consequences (supplier):* legal protection for untested conformance of requirements, reduced earnings (with reduced legal risks)
`Budget dominance`	*Conditions:* tight budget as predominant the factor in contract negotiation, long contract negotiation, turnkey contract dominance on low budget projects, removal of crucial processes or project phases, insufficient preliminary analysis *Actions (supplier):* try as hard as possible to reduce the length of unpaid negotiations on requirements, focus solely on product implementation, work only with a fixed price contract *Consequences (supplier):* reduced margins of profit or even losses if the work is not taken,low product quality, stressful situations

`Poor definition of concerns`	*Conditions:* presence of intermediaries without full control delegation, excessive bureaucracy in the organisations, inadequate supporting figures on the acquirer's side, roles misconception or bypassing, organisational silos, unclear definition of concerns after "go live" *Actions (supplier):* passively accept the situation (no chances of controlling the situation), explain multiple times roles and responsibilities to the acquirer *Consequences:* inconsistency between the information communicated by the delegator and the intermediary, irrelevant information broadcasting
`Low experience in suppliers' selection`	*Conditions:* absence of technical factors in the evaluations of suppliers made by the acquirers, absence of inadequate business plans, unmatching vision or missions, absence of sufficient lead generation to discard eventual prospects on the supplier's side *Actions (supplier):* accept part of these jobs *Consequences (supplier):* possibility of resulting inadequate for the requests of the acquirer

Table 3.2: Categories and subcategories identified in axial coding

Each category will be discussed in section 3.2. **Axial coding** provides a further step towards the definition of theory by identifying connections between codes, subcategories, and categories. However, it is *not sufficient to generate a theory.* The cycle shown in figure 3.1 continues upon reaching theoretical saturation. **Theoretical saturation** is the *point where the analysis of additional data does not bring any additional value to the investigation*. The next and last step to be undertaken is selective coding.

3.1.3 Selective coding

Selective coding is the last step in grounded theory. It is a process that *links all the categories of your research together around a single core category.*

This helps to create a unified theory that is based on the qualitative data collected in earlier coding cycles.

The core category can be derived from one of the categories identified in the axial coding stage, or it can be a new category created from the other categories.

This core category is the main focus of the research and is the central thesis of the study. Selective coding is the final step of the grounded theory approach. Its goal is to either create a new theory or modify an existing one based on research carried out [58].

In this research, the core category is represented by **customer-supplier relationship issues in bespoke software development**. All the categories previously cited are linked to this concept. This category will be the basis for the explanation of the theory described in the next section.

3.2 Results of the work

The previous section presented the application of grounded theory to this case study, to the extent of analysing the recurring issues related to P2. This section describes the theory in terms of the categories presented in table 3.2.
Every concept introduced in the set of subcategories will be explained to the lecturer to provide further details related to the achieved results.
In conclusion, a comparison with the state of the art is provided.

3.2.1 Customer dominance

This study found evidence of aspects related to customer dominance, a category that was also identified by the work of Pasi, Smolander, and Nikula [36]. **Customer dominance** represents the *power relationship that exists between a supplier and a customer.*
According to various interviews, this issue is particularly relevant for small and medium-sized organisations (<50 collaborators and employees).

Category	**Related subcategories**
`Customer dominance`	*Conditions:* business induced power asymmetry, customer dominance on process, perceived risk of involvement in legal case, need to make a good impression on the customer due to word of mouth, market competitiveness, inappropriate communication channels abuse *Actions (supplier):* slow down the development process when the acquirer is late, provide discounts, use inadequate channels to contact the acquirer, insert specific clauses that prevent them to respect deadlines if the customer is late, refuse to enforce clauses, reallocate developers *Consequences:* wastage of resources, reduced earnings, potential loss of relevant information, frustration

Table 3.3: Customer dominance and related subcategories

In this study, three main conditions that might cause situations of customer dominance were identified.
The first condition related to customer dominance is referred to as **business induced power asymmetry**. This subcategory is described as a *condition in which the supplier is afraid of losing the customer and, therefore, acquiesces to customer demands*, even in scenarios where these would represent a threat to earnings.
The development of custom software is a sector that is characterised by great demand but also by great competitiveness in the market. According to BoldData, in Italy there are currently almost 40000 software development organisations, mainly concentrated in the largest cities in the northern part of the country [46]. Therefore, **market competitiveness** is an aspect that every supplier must take into account.

Business induced power asymmetry is often translated into **discounts**. In some cases, suppliers stated that they preferred to lose the earnings margin associated with the project in order to retain the customer. Other suppliers also reported the **need to make a good impression on the customer due to word of mouth**. Almost all of the suppliers declared that the main channel for the acquisition of new projects is word of mouth, regardless of the size of their organisation. Therefore, suppliers felt the need to make a great impression on their customers *to retain them and generate new leads.* Hence, suppliers were *more prone to comply with acquirer's requests*, highlighting a clear threat of customer dominance manifestation.

«Generally we are already satisfied if we manage to break even on costs. It is essential for us to retain our customers and generate new leads through their referral.» - Supplier

«We always give something to the customer, but by controlling them. We are willing to keep a margin of 0 to retain the customer.» - Supplier

Another condition that underlies the dominance of the customer is described as the **perceived risk of getting involved in legal cases**. According to the interviewees, *legal cases* are automatically translated into *adverse situations for them.* In particular, they reported problems such as the necessity of sustaining legal fees, long terms for the eventual recognition of the missing payments, and uncertainties about the judge's sentences. Therefore, *suppliers felt unharmed against their clients*, and **in cases where the clauses of their contracts could be applied, they were likely not going to do so**.

«Legal actions are a constant loss for us. There is no way we are going to benefit from them, even if we win. In most of the cases we feel like we can win the lawsuit, but the refunds issued by the judge's sentence would arrive too late and we would not be able to pay our employees in the meantime. In addition to that, we would certainly also lose the customer.» - Supplier

«It is not worth taking legal action against customers, we always have to impose ourselves or find halfway agreements in these situations.» - Supplier

The last condition identified is the **dominance of the customer in the process**. It highlights a situation where the *customer is able to dominate the software development process selection* and creates situations where the supplier has to depend on the customer's will.

«We try as hard as possible to avoid positions in which we are vulnerable. Customers who are not into software development do not understand the value of some processes, as they ask why they need to pay for their application, without considering the benefits these activities ensure.
We make recommendations to our customers before accepting their terms. If they insist, we will usually accept their conditions, but we flag it as a vision divergence, so we will try to push away the client later.» - Supplier

«We have to impose ourselves on new customers, there are those who trust because they understand that they are not competent in the matter, there are those who do not trust us and there is an important job of building loyalty and increasing trust. If I gave

freedom of choice, the customer would not value them.» - Supplier

Most of the suppliers reported that their *productivity was generally highly influenced by these aspects.* These conditions usually forced the supplier to **adjust their development processes.**
Furthermore, a group of suppliers reported that they experienced **inappropriate communication channels abuse** from the supplier. In these situations, *customers tended to use inadequate communication channels, which were not the ones prescribed by the suppliers* (e.g. phone calls, which in certain cases were not useful to keep track of requests and eventual details). In this case, customer dominance relied on the fact that *suppliers were often forced to use inadequate channels to contact acquirers* (e.g. in cases where they need materials as soon as possible); otherwise, it was impossible for them to reach their clients even for long times. As a consequence, *some relevant information was lost* in the communication process, and there was no written and detailed proof of customers' requests.

«It is frustrating for me to have to continuously call my clients for hours to receive the needed materials.» - Supplier

«Our policies require us to have written proofs of customer requests, but I am often forced to report by email the requests that customers made during phone calls. This makes me lose a lot of time.» - Supplier

In order to respond to these situations, *most suppliers inserted specific clauses in contracts that prevent the supplier from meeting the agreed deadlines if the customer was late* with the delivery of materials. This solution *mitigated the possibility of getting involved in legal causes*, but did not completely resolve it, as customers might still have tried to take legal actions.
However, the forced adaptation to customer's rhythms also showed another hidden criticality related to the productivity of the teams of the supplier's organisation. In fact, delays in delivering the necessary information or materials often causes *forced interruption of work.* In these situations, *team members have to be temporarily reallocated.* Larger suppliers reported that they resolved this issue using time and material contracts. By using this type of contract, customers were obliged to pay for the days in which the team is inoperative due to customer's delays not reported in time. Conversely, *smaller suppliers tended not to apply similar clauses, as they would have hindered the relationships between the two parties.* As a consequence, they often declared to find themselves in *weakened positions* with respect to their customers.

3.2.2 Improper change management

Improper change management was another category similar to "Changes brought during the project" presented in Customer-Supplier Issues in Software Development [36]. This study added other subcategories and observations related to change management.

`Improper change management`	*Conditions:* continuous changes to graphical aspects for no functional reason, core product scope changes intercepted too late, changes related to the dissatisfaction of uninvolved stakeholders, unspoken needs *Actions (supplier):* require to pay (only) for major change requests, provide discounts *Consequences:* exceeding of budget and deadlines, dissatisfaction of customers, reduction of the profit margin, scope creep

Table 3.4: Improper change management and related subcategories

The first noteworthy condition that was observed is the **continuous set of changes required to graphical aspects**, even for no specific reason. According to a consistent subset of interviewees, *customers were excessively focused on the graphical aspects of the interfaces*, rather than focusing on flows and functionalities. The interviewees declared that their customers spend too much time reviewing wireframes and mockups, often trying to emulate the patterns of big competitors (which in some cases might even hinder the usability of the product) rather than focusing on the scope of the functionality.

«We had an internal department that provided graphics and mockups for the design of certain sections and functionalities of the application. Content analysis and creation very often ended up being long because various figures were involved and many reviews and approvals were required.» - Customer

«Customers perceive the design of interfaces as one of the fundamental part of the project. They often provide some guidelines and feedback at the beginning, but then they want to emulate existing similar platforms.» - Supplier

The second condition that emerged from the data analysis, **"Core product scope changes intercepted too late"**, is a major cause of project failures.
In product development, the core product is represented by the fundamental utility or benefit that a consumer receives from owning an item [55]. In software development, the **core product** is therefore represented by the *essential set of functionalities required by its users.* The core product identifies a set of characteristics that make the product unique. Therefore, the core product should not be changed.
However, in software development, *changes may occur for a variety of reasons* (e.g. market needs, trends, state of the art technology not sufficiently advanced, legal concerns, etc.). It is essential to intercept notable changes to core functionalities before they are implemented. To avoid these issues, some suppliers declared to propose solutions as preliminary market and competitors analysis, interviews to prospects, feasibility studies and Proof of Concept (POC) implementations. However, these solutions are costly and might get skipped for budget reasons, delaying the facing of possible issues in the development phase. The risks of doing so imply aspects such as *scope creep, higher budget overruns, and delays in project delivery,* as described in paragraph 1.3.1.

Both acquirers and suppliers recognised the importance of intercepting changes as soon as possible. But some perceptions and conditions might delay the communication of changes.
In particular, two main subcategories were identified as related to this aspect: **changes related to the dissatisfaction of uninvolved stakeholders** and **unspoken needs**.
The former was reported by both customers and suppliers. It depicted a scenario in which *one or more stakeholders that were not involved* (e.g. they delegated the responsibility of following the project to another stakeholder, they came in touch with the project during its development, or they were not dedicating enough time to the project) d*emanded relevant changes* during (or even after) the implementation of the required functionalities.
The main concerns with these situations arose when these stakeholders had the *right to influence or even interrupt the project* and, therefore, to exercise a position of customer dominance.

«There was a serious shortage on our side because the needs expressed by another department that had never emerged before arose, leading to a late discussion of fundamental requirements.» - Customer

«On our side there were relevant communication problems with an intermediary that should have helped us to understand the technical implications of some requirements. As a result, we had to change the core product, but luckily we were only slightly over budget.» - Customer

«In a two year project we experienced several issues. We are generally interested in finishing the job regardless of the difficulty. In this case, the client did not know what he wanted. A milestone-driven approach seemed the best option. However, a directive was imposed by a stakeholder who was not involved from the beginning. This directive obliged us to present a quote with mandatory deadlines, while the project was already begun.» - Supplier

The last condition observed in this category, denoted **"Unspoken needs"**, refers to a characteristic often *shown by customers who are approaching software development for the first time.* Some interviewees on both sides reported experiencing a budget overrun due to *requirements that were taken for granted by the customer but not by the supplier.*

«I assumed some things were easy to change when they were not. As a result, I had to allocate some additional budget, as these functionalities were not included in the contract.» - Supplier

«Sometimes our customers take for granted the presence of some functionalities because they are familiar with them. Their idea is often based on competitors which are much bigger than they are, so they assume the presence of certain functionalities shown by similar platforms in their product, too. However, we cannot take for granted the presence of these functionalities when making estimates.» - Customer

Change management is usually a slightly *more contained problem when adopting agile approaches* (and, in general, dedicated team or time and materials contracts), since the methodologies themselves are usually change-driven.

However, this *problem* becomes *much more evident when the waterfall model or fixed price contracts are adopted.* In fact, supplier estimates may include a small margin for changes in the contingency factor. However, when major requests arise, they need to present a new estimate for the specific change. This estimate is usually denoted as **change request**.
However, *customers are generally dissatisfied in having to pay for change requests*, therefore, most of the suppliers adopting fixed price contracts declared that they tended to *gift part of the changes* if they did not heavily impact the budget. This aspect could also be related to the need to make a good impression on the customer described in Customer dominance.

«We make change requests only if we deem it necessary, otherwise we give away (up to 10% of the project cost).» - Supplier

«We only make change requests for larger demands, as there is a lot of work to make our customers accept a new budget allocation.» - Supplier

3.2.3 Knowledge transfer barriers

Knowledge transfer barriers is another category emerged from this analysis that was already identified by Customer-Supplier Issues in Software Development [36]. It refers to *situations that may hinder the successful completion of the information exchange process between the organisations* involved in the project development.

`Knowledge transfer barriers`	*Conditions:* acquirer's lack of experience in software development, prohibitive learning curves for the acquirers, insufficient domain knowledge, inability of defining requirements, lack of understanding of the value provided by certain activities, lack of communication among stakeholders on the acquirer's side *Actions (supplier):* provide training on the various topics, hide the estimate for for activities not valued by the acquirers, passively accept the situations *Consequences (supplier):* loss of information, misunderstandings, acquirer's unawareness of the quality of the product, estrangement of some stakeholders

Table 3.5: Knowledge transfer barriers and related subcategories

The first condition related to this category is verified with customers who have no previous experience in the development of software projects.
The acquirers of custom software represent a heterogeneous set from the perspective of previous experience related to software development.
The **lack of previous experience in the IT sector** may represent a limiting factor in knowledge exchange between this type of acquirers and their suppliers. In fact, the development of software applications spans over many topics (e.g. technologies involved, architecture of the system, software development life cycle, etc.) and the *learning curve* would be quite *prohibitive* for them.

However, *customers are required to learn some technical aspects* for various reasons (e.g. acting as a data controller according to General Data Protection Regulation (GDPR) requirements).

Customers may become acquainted with a subset of these concepts during the development of the project, but generally may have difficulties understanding the relevant aspects. For example, some customers were unable to tell which software development life cycle model was followed in the project, which part of the architecture was involved in certain requests, or which third-party services were implied for some specific scopes. This may also be related to insufficient customer training or lack of a shared glossary definition. However, these last two aspects should have been properly assessed in a more structured way and would have required too much time during the interviews.

Another condition that emerged from this study was **insufficient domain knowledge**. It refers to the situation where the *customer's knowledge of the application domain is not available to the supplier*, as the customer is not able or willing to provide the knowledge of the application domain to the suppliers [36]. This appeared quite frequently in the interviews, especially in relation to delays in materials delivery.

«It may happen that our customers are unable to provide the business information needed for the project, for example, in situations where there are many steps or different entities involved.» - Supplier

«Even from the point of view of defining user stories, there are often problems, what is missing emerges in the development phase.» - Supplier

However, another similar condition is represented by the **inability of acquirers to define the requirements.** In particular, some customers reported that they were unable to grasp which aspects could help them to improve the definition of requirements, especially for what concerned non-functional ones.

«It is a particularly new area for me, the supplier should improve the ability to think like a person who does not know anything about the IT sector, I did not understand how I could improve in defining the requirements.» - Customer

«At the beginning I did not get some concepts that should have been assessed before, as, for example, the implications that some choices have on the requirements.» - Customer

One of the main conditions that is likely to affect the quality of the final product is the *lack of understanding of the value provided by certain activities* shown by the customers.

This subcategory refers to a cultural mismatch often shown by acquirers who have little or no experience in software projects. In this case, customers show a lack of understanding of the value brought by certain project activities and phases. In the worst cases, the acquirers were not even aware of the existence of these factors.

Almost every interviewee on the supplier side stated that a subset of their customers stated phrases like "Why should I pay for tests? I thought you had already tried this version and everything worked fine!" or "I only need the user's guide documentation".

As a response to this condition, some *suppliers reported not declaring the presence of these activities in estimates*, as they would bring arguments of discussion with their customers that they did not want to face. They reported that this workaround seems to work efficiently, even though in this way the customer is unaware of the level of quality that the commissioned product presents.

«I do not understand what do you mean by automated testing.» - Customer

«I estimate the activities that are generally the result of discussions within the development costs, without separating them from the rest. If I formalised them explicitly, many would make me remove them because they would not understand the added value they bring.» - Supplier

In conclusion, the last condition related to knowledge transfer barriers that resulted from this investigation was the **lack of communication between stakeholders** on the acquirer's side. Specifically, this scenario refers to the case where not all stakeholders on the acquirer's organisation are regularly updated on the project status. The typical scenario in which this situation is likely to be verified involves the nomination (formal or informal) of a *representative who should be in charge of reporting all the relevant information coming from one side to the other.*
However, there may occur situations in which not all stakeholders are up-to-date. In particular, this issue is perceived in face-to-face events where all stakeholders are present. In this specific scenario, the supplier has no control over the situation, and influencing the selection of intermediaries may lead to personal disputes that are not beneficial to the progression of the project.
The *suppliers* stated that they usually have to *passively accept these situations.* However, they try to promote the visibility of information by using indirect channels (e.g. emails, instant messaging groups), where they report the essential information to all the intended stakeholders.
However, these actions do not seem to work every time, as some stakeholders do not regularly consult these channels.
Consequently, some *stakeholders* end up being *estranged from the project and may cause miscommunication and misalignment* during moments of confrontation.

«Sometimes during sprint reviews I do not have the impression that all stakeholders have been informed about certain choices and aspects that the representative and I agreed on.» - Supplier

«We had to remove an intermediary and take over from him, as there have been occasions in which we were not informed about some relevant changes discussed with the supplier.» - Customer

3.2.4 Negative factors of customers' attitude

This category refers to all the *issues in the customer's attitude and behaviour that may represent a threat* to the development of a custom-made software product.
This category assumes significant relevance in contexts where the waterfall model is applied.

Negative factors of customers' attitude	*Conditions:* acquirer's progressive loss of interest towards the project, waterfall model adoption, defensive attitude over cooperation, passive attitude and contribution, urgency driven requests, lack of interest in the state of project, lack of time-effort definition for the acquirer, not way for the acquirer to monitor project progress and costs *Actions (supplier):* constantly ask the acquirer to test the application, incorporate tacit acceptance clauses in the contracts, negotiate discounts upon being threatened with lawsuits, evaluate the profile of the customer before accepting the work, provide an internal tool to promote visibility of project progress and costs *Consequences (supplier):* legal protection for untested conformance of requirements, reduced earnings (with reduced legal risks)

Table 3.6: Negative factors of customers' attitude and related subcategories

Many suppliers reported how their *acquirers tended to gradually lose interest in their project.* According to their point of view, this condition was likely to manifest more frequently while following the waterfall model and fixed price contracts. In these cases, the peak of customer's attention was reached during the analysis and design stages and progressively decreases during development and testing stages. When adopting the waterfall model, it is reasonable to assume that the acquirer might assume a passive attitude towards the project. In fact, customers who prefer not to participate in the development process and who want to be involved at the start and then receive a finished product prefer to opt for this model [51]. But this adaptive behaviour becomes an issue in the testing phase, where *the acquirer is generally asked to try the functionalities and provide feedback on the application.* According to suppliers, customers often tried to delay this activity as long as possible or did not perform any test at all.

As a response to this condition, most of the *suppliers incorporated tacit acceptance clauses* if the tests are not carried out within a predefined number of days from the announcement of the release of the new version in the staging environment.

Consequently, suppliers are legally protected, but the attitude of the passive customer usually remains unchanged.

«I have to contact my customers frequently to remind them to try the application.» - Supplier

«Some customers expect everything to be tested and working. They do not immediately understand that, as we need them to approve the results of the job by contract, they need to try the application and report any possible problems. They usually still do not test the new version. Therefore, I added a clause to the contract that grants tacit acceptance of the implemented functionalities if the client does not provide any feedback for more than 30 days.» - Supplier

Another frequent condition often related to the adoption of fixed price contracts is represented by a **defensive tendency over cooperation** manifested by the customer. This is particularly evident in cases where some issues occurred during the development stage. Some suppliers reported experiencing situations in which *acquirers required heavy discounts* or even *threatened to take legal actions* upon deadline overruns or alleged non-conformities on some requirements. In these cases, acquirers may exercise a position of dominance towards the suppliers, which from contract assume the risks of plan through a fixed price model. Depending on the gravity of the situation, suppliers stated that usually try to **negotiate some discounts** or the addition of features. This usually brings losses to the supplier's organisation and perhaps might still lead to legal causes later on, causing tension between the two parties. Additionally, some suppliers reported a more preventive approach. Specifically, they *asked* their potential new customers *to describe their previous experiences with other software suppliers (if any)*, and they took into account the eventual threats that arose in the response. However, this does not completely cancel the risk of having disputes if the project is taken.

«A former client sued us because we missed a very important deadline for him due to some requirements that had not been explored enough in the analysis phase.» - Supplier

«We try as hard as possible situations in which we can be involved in legal causes, granting some additional discounts in case something goes wrong. However, before starting a new project, we always ask the acquirer to tell us about his previous experiences with other companies in the sector, in order to identify any red flags in the behaviour shown towards any former supplier.»

The last relevant condition reported by suppliers is the **passive attitude** with which some of their acquirers face the project. Theoretical sampling led to the definition and further analysis of two scenarios in which this issue is likely to be displayed. Specifically, the analysis focused on customers that frequently tended to make **urgency-driven requests** and projects where the acquirer did not follow the monitoring processes and costs.

In this category, two recurring conditions appeared to be strictly related to these actions.

The former is related to the *lack of a clear definition of the effort required to the acquirer* during the project. In particular, this condition refers to a situation where the supplier only communicates the tasks and activities in which the acquirer would be involved, without associating them to an estimate of the time needed to complete it. As a result, an acquirer that has no experience in software development may underestimate the time needed to perform some tasks, and therefore procrastinate them to periods with more availability of time. In response to these situations, suppliers tended to accommodate requests when possible, but pointed out to customers that these situations were very risky and stressful.

«We do not give deadlines, we highlight the need to complete the required tasks. We do not give an estimate of the time required per week for the project, as it may vary according to the various activities in which the customer is involved. We try to reduce the things customers have to do as much as possible.»

«We give deadlines just to have a complete list of requirements before starting the

project and to have the availability of the environments in which we need to install the application. We do not give an estimate of the time required to complete these tasks.»

The latter referred to the *lack of an immediately available way to track progress and sustained costs*. In fact, not all suppliers reported communicating directly metrics such as project costs and the respective level of completion of the same. In these scenarios, the supplier only communicates which requirements have been completed. The customer may trace the costs incurred through invoices.

Not showing these metrics may prevent the customer from knowing whether some unexpected events have occurred. Consequently, the acquirer assumes that everything is proceeding well and that there is no need to re-plan any aspects of the project.

However, these conditions did not seem to describe the complete scenario, as some suppliers declared that even though they communicated these metrics to their customers, the latter did not seem particularly engaged in the project. Unfortunately, it was not possible to fully investigate the matter with these specific customers. This aspect will then be discussed in section 4.2.

«We keep track of the costs incurred and the number of days spent by each team member on the project. We establish checkpoints a priori with the client according to which we should be at a given percentage, and we relate it to the percentage of budget used. We generally set 5 to 7 checkpoints per project. The customer does not directly know the costs incurred, but can obtain them from the invoices and the amount established in the estimate. »

«We use our own internal software to track these metrics. We track the hours worked by each employee on a project, at the end of the month we create a report that compares the cost spent and the expected cost based on the hours spent and expected. We don't always show them to the customer. It depends on the type of contract (shown in T& M contracts, hidden in fixed price contracts). »

«Not all clients follow the project costs, however, they are available to them. We have built a specific tool for this scope. Not all customers understand the correlation with the project progress, but they would like to (e.g. they count the tasks and calculate the percentage of completed requirements but do not understand that not all tasks have equal weight in the completion of the project).»

3.2.5 Budget dominance

Budget dominance is a category that reflects *situations in which the budget becomes the predominant constraint of the project.* Suppliers stated that this problem becomes particularly evident in projects such as Content Management System (CMS) based e-commerce, websites (landing pages) and Minimum Viable Product (MVP) of applications (web or mobile) where the business analysis has not been carried out.

The budget for this type of project might be really tight (sometimes it is not even declared at all).

This problem was strongly perceived by many suppliers. In fact, many of them declared that they were not willing to take on these projects, as there would be many risks and stressful situations while working on them.

Budget dominance	*Conditions:* tight budget as predominant the factor in contract negotiation, long contract negotiation, turnkey contract dominance on low budget projects, removal of crucial processes or project phases, insufficient preliminary analysis *Actions (supplier):* try as hard as possible to reduce the length of unpaid negotiations on requirements, focus solely on product implementation, work only with a fixed price contract *Consequences (supplier):* reduced margins of profit or even losses if the work is not taken,low product quality, stressful situations

Table 3.7: Budget dominance and related subcategories

The first criticality observed in this category is a **long preliminary contract negotiation**. The standard flow which characterises this provides a first presentation of the project and its use cases carried out by the acquirer. After this preliminary step, the acquirer asks the supplier to provide an estimate for project development. The quote is then generally translated into a fixed budget contract. Therefore, the delivery of all the requirements becomes mandatory.

However, in many cases, the estimate provided by the supplier is much higher than the budget that the customer expected to allocate, leading to the first condition, denoted as tight budget as the predominant factor in contract negotiation.

From this situation, a long negotiation between the two parties begins. The focus of the talks is shifted towards the requirements that will be included in the contract and those that will be delayed to further phases. This settlement generally takes a long time (even months) to complete and is *generally offered by the supplier, which sustains some relevant losses before the project begins.*

Consequently, the supplier tries to *reduce the analysis* needed to provide the estimate to the minimum. This need to rush may inevitably present some repercussions during the development phase.

At the end of the negotiations, what mainly drove the selection of the requirements and the delivery deadlines is the client's budget, which is likely not to be varied from the initial phase.

At this point, the supplier is placed in front of a fork in the road: accept the plan in order to recover a part of the expenses incurred or lose the plan without receiving some compensation for the preliminary job carried out.

As a consequence, the supplier willing to accept the work needs to *focus completely on the product implementation.* Therefore, some activities and services that should be included in a software project are removed to maintain lower prices. Specifically, suppliers tend to *remove* most of the needed *documentation* and *automated testing* from the activities carried out during the project. These removals allow them to generate a small profit on the project, although they might severely affect several processes and aspects of the product (e.g. quality management, project planning, risk management, measurement process, verification and validation processes).

Another forcing that the supplier must face is the need to adopt a **fixed price contract**. As most of the suppliers reported, there would not be enough room for changes and flexibility within the allocated budget. Therefore, the adoption of the waterfall model becomes a necessity. However, this choice usually becomes quite stressful for the supplier, as acquirers will still try to require some changes along the way and are not willing to allocate some additional budget allocation to implement change requests.

«We decided to stop working on projects such as websites and e-commerce. In our previous experiences, we never made relevant profits on them, and we were forced to work superficially. In some cases, we suffered considerable losses related to the loss of work after the drafting of the estimate, which is usually offered to the customer.» - Supplier

« We do not deal with landing pages and e-commerce, as income is not worth the stress and problems that these projects bring.» - Supplier

3.2.6 Poor definition of concerns

Poor definition of concerns refers to situations in which the *definition of roles and responsibilities is not clear to at least one of the stakeholders involved.*
On the one hand, organisations that mainly work with smaller customers declared that these situations are likely to be verified with acquirers commissioning a software project for the first time. According to them, the presence of these situations is mainly due to customer inexperience and knowledge transfer barriers.
On the other hand, suppliers working with larger organisations reported identifying these types of issues when many different stakeholders are involved. Specifically, this problem is verified when it is difficult to establish responsibilities in tasks that span across different sectors of expertise or when there is an excessive bureaucracy involved.

`Poor definition of concerns`	*Conditions:* presence of intermediaries without full control delegation, excessive bureaucracy in the organisations, inadequate supporting figures on the acquirer's side, organisational silos, roles misconception or bypassing, unclear definition of concerns after "go live" *Actions (supplier):* passively accept the situation (no chances of controlling the situation), explain multiple times roles and responsibilities to the acquirer *Consequences:* inconsistency between the information communicated by the delegator and the intermediary, irrelevant information broadcasting

Table 3.8: Poor definition of concerns and related subcategories

According to suppliers, one of the main conditions that leads to a poor separation of concerns is represented by **delegation of communication with the supplier to an intermediary**. The latter can be internal (another employee of the customer's organisation) or external (a freelancer involved for this specific purpose). The introduction of this figure adds a layer of complexity, as usually the delegator is not willing to be involved personally in the development to the project. However, the *delegator still* wants to be informed about the status of the project and *wants to be able to exercise a good level of control* over it. There may occur situations in which the delegator intervenes and provides information other than that provided by the intermediary, or vice versa. At this point, it becomes *unclear to the supplier what direction to take and who should be informed.* In the worst cases, the intermediary is not qualified enough to sustain this role, and he/she ends up being an obstacle for the project development rather than bringing a valuable contribution. In these cases, the suppliers stated that they did not have a high level of control over the situation. *They usually just accepted the figures offered by the acquirer.* In fact, only organisations working with customers that already have an IT department are able to specify the characteristics of the stakeholders that should be involved from the customer's side.

«Since it was my first software project, I asked a friend of mine who was more into the IT sector to help me. However, it became evident that at a certain point he did not fully understand my needs and he talked directly to the supplier rather than consulting me first. As a result, some changes that were implemented were different from what I asked.» - Customer

«The customer directly gives me the reference people, the most I can do is specify the characteristics that these stakeholders should have.» - Supplier

Another scenario reported by suppliers involved customers in their first software development project. Specifically, they referred to some conditions where customers were unable to discern what was inside the business domain and what was inside the technical domain. Customers therefore manifested wrong expectations towards the tasks of the supplier. The suppliers declared that these situations were usually solvable once the supplier provided further explanations to the customers.

In conclusion, for what concerned poor definition of responsibilities, the suppliers reported two other problematic conditions: **irrelevant information broadcasting** from the acquirer's side and **roles bypassing or misconception**. The former refers to situations where the customer uses indirect communication channels to broadcast information that is irrelevant or misleading to other members of the team. In these cases, the suppliers do not have direct control. Therefore, all they can do is to *explain again the roles and responsibilities of the stakeholders on their side.*

The latter is usually a consequence of the former and *depicts a scenario in which customers (voluntarily or not) communicate directly to the team members without passing through the project manager.* These actions may cause inconsistent situations between the directions provided by the project manager and the ones provided by the customer. For what concerned larger organisations, both suppliers and customers reported that roles bypassing or misconception is verified together with the presence of **organisational silos** [30]. This term refers to situations where *teams of people are isolated from other parts of your business due to a minimised flow of information.* Organisational silos might be the cause of limited collaboration and complicated management of responsibilities and roles within the organisation.

«In a project we experienced some communication issues, as the customer kept communicating every change request directly to the developer team, bypassing my role (project manager). As a result, developers had to continually come to me to ask whether they should have implemented the change or not.» - Supplier

«One of the biggest problems we face as an organisation involves organisational silos. It may happen that some tasks that involve many different teams from different departments take a long time to complete compared. This is generally caused by problems related to communicating the right information to the correct person. Sometimes we bypass roles as we perceive that some figures are acting as bottleneck (e.g. because they have a lot of work to do and cannot follow everything) but as a consequence their knowledge is not up to date with the latest decisions. » - Supplier

In conclusion, a subset of interviewees on both sides reported experiencing situations where some responsibilities remained unassigned after the development of the project. Specifically, interviewees referred to scenarios where they did not agree in short times about maintenance contracts, therefore some roles and responsibilities remained vacant upon reaching a new agreement.

«At the end of the development of the first version of the product, we did not immediately agree on a new maintenance contract. However, the relations remain excellent, but we could have lived situations in which we would have needed the help of the supplier to resolve some problems. » - Customer

3.2.7 Low experience in suppliers' selection

In conclusion, this category highlights all the issues related to the **lack of experience in the selection of suppliers shown by the acquirer**.
This category finds particular relevance with customers that are approaching software development projects for the first time.

`Low experience in suppliers' selection`	*Conditions:* absence of technical factors in the evaluations of suppliers made by the acquirers, absence of inadequate business plans, unmatching vision or missions, absence of sufficient lead generation to discard eventual prospects on the supplier's side *Actions (supplier):* accept part of these jobs *Consequences (supplier):* possibility of resulting inadequate for the requests of the acquirer

Table 3.9: Low experience in suppliers' selection and related subcategories

Successful custom-made projects are usually distinguished by a common vision, mission, and objectives shared among all stakeholders.
However, *the selection of the proper supplier* may be a non-trivial task. Customers who do not have experience in software development may face several difficulties in selecting the appropriate supplier.
This type of interviewee on the acquirer side shared a common trait. *Only a few of them included technical and vision criteria* in selecting the supplier. The remaining part often reported that the *choice* was driven mainly by *word of mouth and similarities between the two organisations.*
In particular, the most reported criteria included: geographical proximity, prices compared to competitors, advice from acquaintances, proximity in age between the

supplier's and customer's contact persons, and positive feelings following the first interview.
These factors might be partially good for seeking social affinities between the two parties. However, other criteria should be taken into account. Some examples might target the capacity and capability of the supplier, the expertise on the topic, quality standards and certifications, ways of working, etc.
Smaller suppliers usually showed interest in the topic. However, they declared that they `did not have enough lead generation` to be able to cooperate with clients that met all these requirements. Therefore, sometimes issues related to cultural or business differences between the two parties occurred.

«An acquaintance of ours introduced us to the company. We had good feedback from the first interview. They helped us to build the needs, we appreciated problem-solving skills quickly. Geographical proximity and close proximity to age were two very convenient factors to build a good relationship.» - Customer

3.3 Comparison with the state of the art for problem P2

The objective of this thesis involved the identification of acquirer behaviour traits that can be translated into potential problems during the development of a custom software product. In particular, this problem was analysed through the application of grounded theory, a qualitative research methodology used to derive a theory based on the collected data.
Other previous studies were conducted by applying qualitative research methods to their respective case studies. In particular, this work considered the four articles described in paragraph 2.2.9.

A framework for identifying software project risks

The first article that was analysed, "A framework for identifying software project risks" by Mark Keil, Paul E. Cule, Kalle Lyytinen, and Roy C. Schmid, highlighted how project managers could not exercise control over every type of risk present in software development projects. For this study, the main class of risks considered was the "customer-mandate risks". The authors referred to these risks as those related to the lack of customer support and commitment during the development of a software project, reporting that they can be influenced, but not controlled, by the project manager [28]. This research led to similar conclusions, as *the results described in section 3.2 highlighted several situations in which the project manager can just partially intervene.*

Agile customer engagement: A longitudinal qualitative case study

The second study taken into account was "Agile customer engagement: A longitudinal qualitative case study" by Geir Kjetil Hanssen Tor Erlend Fægri [20]. It described the benefit in the transition of a small organisation from waterfall-based methodologies to agile methodologies (evolutionary project management). This article pointed out interesting analogies related to customer engagement. In particular, it highlighted various drawbacks to the latter when the waterfall model is applied. The study emphasised how acquirers usually do not receive any form of gratification/compensation, or formal power either, when the waterfall model is used. Consequently, the customer sees his involvement only as an opportunity to affect the course of development. According to the authors, acquirers might perceive the sensation of being outsiders rather than playing an active role in the project. *This result was analogous to the one reached in analysing negative factors of customers' attitude.*

The last two articles, "Customer-Supplier Issues in Software Development" by Aparna Pasi, Kari Smolander, and Uolevi Nikula [36] and "Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers' Perspective" by Lubna Siddique and Bassam Hussein [44], conducted studies to investigate the reasons behind the conflicts and issues between suppliers and acquirers in software development.
In both cases, the researchers adopted grounded theory as the qualitative method designated to conduct the research. Furthermore, both performed data sampling by interviewing stakeholders on the supplier's side.
This research instead also took into account the point of view of the customer.

Customer-Supplier Issues in Software Development

"Customer-Supplier Issues in Software Development" identified three main categories of aspects that may hinder the development of the project.
The first category, knowledge transfer difficulties, described situations where there is a lack of successful communication between acquirers and suppliers. In particular, the three subcategories related to knowledge transfer difficulties target aspects such as insufficient domain knowledge, incompatible terminology, and language barrier.
Insufficient domain knowledge is defined by the authors as the situation in which the customer is unable or unwilling to provide the supplier with knowledge of their application domain.
The second subcategory, incompatible terminology, is defined as the incompatibility between the different terminology of software developers and experts in the application domain (e.g. acquirers and users).
Finally, the authors defined language barrier as the subcategory that includes issues that are caused by the limited knowledge in the language used. This issue of transferring and conveying knowledge can lead to an incomplete understanding of the requirements, and may even result in incorrect requirements implementation if the development team does not comprehend them.
The second main category identified in this work is change management during the project. In particular, this category refers to changes that customers require during the project. The authors did not provide any subcategory to describe this problem, and this work was able to analyse this type of issue in a more detailed way.
Finally, the last category identified by the authors paper was customer dominance.
Customer dominance was described through two main subcategories: customer dominance on process and business induced power asymmetry.
The former revealed a condition where the customer dictates the software development process. The customer does not work along with the development team, and thus a discrepancy between the customer and the development team schedules occurs.
The latter highlighted a situation where the development team does not refuse customer requests due to fear of losing the customer.

The analysis described in this thesis presented a lot of similarities with the work of Pasi, Smolander, and Nikula, and extended its results by defining new categories and enhancing the descriptions of the others already identified.

First of all, *this study confirms* the existence of **knowledge transfer barriers** between acquirers and suppliers. An additional condition that refers to this category was identified as the *lack of experience in software development shown by the acquirer.* This condition evidenced how acquirers that are not acquainted with software development experience hard times in understanding several concepts (e.g. SDLC models, architectural concepts of the application, roles, and responsibilities), as the learning curve is prohibitive. However, for some reasons, they are required to get in touch frequently with some concepts (e.g. for legal requirements). Another additional condition that was introduced in this category highlighted the *inability of defining requirements*, defined as the inability of grasping which concepts could be useful in defining requirements (especially for what concerned non-functional ones). The last two aspects that enriched the concept of knowledge transfer barriers were the *lack of understanding of the value provided by certain activities* and the *lack of communication among stakeholders on the acquirer's side.* The former underlined how some customers were unable to perceive the value of certain activities as presented by the suppliers. The latter instead revealed possible knowledge barriers among the various stakeholders involved on the acquirer's side. Specifically, it referred to situations where a representative of the customer fails to report all relevant information to the other stakeholders.

Additional contributions were also associated with **change management during the project**.

This research focused on cases of inadequate change management, reporting how *customers often requested too many modifications to the visual elements of the interfaces.* Furthermore, it described situations where *changes in the scope of the core product were intercepted too late.* These late modifications, in particular, depended on two additional aspects: *changes related to the dissatisfaction of uninvolved stakeholders and unspoken needs.* The former reported how the presence of delegates who entrusted the project to intermediaries, but who still maintained a high level of control over the project, could affect the core product scope. The latter instead underlined how unspoken needs of customers could turn require changes to the core product scope.

Finally, other consequences that derived from improper change management were discovered. In particular, the most relevant ones were identified in *customer dissatisfaction related to the payment of change requests* proposed by suppliers and the consequent *reduction of the profit margin on the supplier's side* due to the unpaid compliance with part of the acquirer's requests.

In conclusion, other major contribution were brought to the **customer dominance** category. Also this research found evidence related to the concepts of *customer dominance on process* and *business induced power asymmetry.* However, it also explained how *the fears of the supplier* must also take into account the *necessity of making a good impression on the customer* to try to gain new leads through word of mouth, which was stated to be the primary source for lead generation for many suppliers.

Another condition identified was termed the *risk of involvement in legal cases feared by suppliers.* Suppliers asserted that legal proceedings often led to unfavourable outcomes, such as having to pay for legal costs, waiting a long time for overdue payments to be acknowledged, and not knowing what the court's ruling will be. As a result, suppliers declared to feel insecure against their customers and, in cases where the terms of their contracts could be enforced, they were unlikely to do so.

The last noteworthy condition related to customer dominance that was identified was referred to as *inappropriate communication channels abuse.* This definition denoted situations where customers continuously used communication channels that were not the ones agreed upon. This condition generally caused frustration for the supplier as the latter was forced to use inadequate tools that could cause loss of information or the absence of written proof.

Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers' Perspective

In conclusion, the research conducted by Lubna Siddique and Bassam Hussein reported several categories that could cause conflicts between customers and suppliers. According to the authors, the reasons beyond conflicts between the two parties should be researched in: the role of the product owner, an inexperienced project manager, the customer´s lack of knowledge about methodology, organisational hierarchy in public companies, contracting, scope aspect, personal egos, financial issues and not getting the right team.
This research agreed with a subset of these categories and was enriched by providing some additional aspects.
The concept of the **customer´s lack of knowledge about methodology** was explained as the risk of putting too much responsibility on a customer that is not mature enough and lacks knowledge of the way agile methods work. This category finds some similarities with *knowledge transfer barriers* and *customer dominance*, with refer to conditions such as lack of experience in software development and inappropriate communication channels abuse.
The o*rganisational hierarchy in public companies*, described as the complex structure of bureaucracies implemented by public administrations to gain more control over the project, is somehow related to a **poor definition of concerns**. In particular, the authors referred to situations where the product owner would not be able to work without talking to someone else in other departments in order to get the necessary approvals to make a decision. Consequently, the flow of information takes a long time to be distributed and may suffer some bottlenecks. This situation is much similar to the condition described as organisational silos, which depicts situations where teams of people are isolated from other parts of your business due to a minimised flow of information.
Finally, other analogies were identified between the categories *financial issues and improper change management.* In particular, both studies reported diffused *dissatisfaction shown by customers who had to pay for change requests.*
In conclusion, this chapter aimed to provide the lecturer with information related to the application of grounded theory to the case study, the main findings achieved by this research and the comparison of the obtained results with the state of the art that was previously discussed in paragraph 2.2.9. The most relevant aspects discussed in this chapter regard the eight main categories identified during axial coding and their discussion data-based theories in section 3.2.

Chapter 4

Conclusions and outlook

This conclusive chapter provides a summary of the project, including the various steps and choices undertaken during the development of the project. In addition, a retrospective on the results and development of the project will be presented. In conclusion, the last section will present the aspects related to the extensibility of the project.

4.1 Synthesis

This thesis describes the conception and development of a project aimed to **analyse the main issues in customer-supplier interactions in the context of custom software development**. The path towards this analysis was not straightforward, as it required performing some choices in the course of work.
The **first objective** to be achieved was the *definition of what could be considered a customer-supplier issue in this context.* To this end, I conducted a preliminary market analysis that included a systematic review of the state of the art, some brainstorming sessions with peers in my organisation, and a set of ten interviews with different project managers from bespoke software development organisations. The results of the interviews were then analysed by performing a Thematic Content Analysis on their transcriptions.
The latter led to the identification of four main problems.

ID	**Problem**
P1	Selection of the appropriate software development life cycle
P2	Analysis of the interactions with the customers
P3	Improvement of the estimation of the contingency factor in quotes
P4	Analysis of the main critical issues and weaknesses of contracts between the two parties

Table 4.1: Set of problems considered in the domain

Nevertheless, resource constraints led to the need to perform a selection on the problem that would have been the object of analysis.

Therefore, the **next objective** to be pursued became the *selection of one of these four problems.*

The first step provided the *selection of the methodology with which to address the selection.* However, in these scenarios, traditional quantitative metrics could not be easily estimated and the assessable criteria for selection were mainly qualitative.

Therefore, the domain of possible methodologies from which to choose took into account only those that took into account this type of factors in their processes. In particular, the set of possible candidates included the ranking method, the scoring model, and the analytic hierarchy process.

The **criteria for the selection of the problem** took into account *the relevance of the problem for the majority of suppliers who would have been interviewed, the feasibility of a noteworthy contribution in a period of six months, my level of expertise in the specific sector of each problem, and my level of capability and capacity to eventually resolve the problem.* Given the restricted set of elements in the domain and the limited number of parameters that could impact the choice, the ranking method was evaluated as the most suitable methodology for this use case.

As stated above, **the participation of the supplier** would have been necessary in order to focus on the main threats perceived by the latter.

Therefore, I designed **a questionnaire** that aimed to *analyse more in detail the interest and perceptions of the suppliers towards the four problems.*

This version of the questionnaire provided five main sections

Section	**Description**
`Interviewee profiling`	Highlight the main characteristics of the interviewee's organisation, to the extent of relating the characteristics of the supplier to their needs and expectations
`Investigation of interest towards problem P1`	Collect information about the implementation of the life cycle model management process provided by the interviewee's organisation and highlight the feelings and sensations of the interviewee towards this process implementation.
`Investigation of interest towards problem P2`	Collect supplier's perceptions towards their customers, their behaviour, and the consequences that the latter has on the project
`Investigation of interest towards problem P3`	Gather information on the way of working of the interviewee's organisation towards the drafting of quotations, with a particular focus on the contingency factor

`Investigation of interest towards problem P4`	Extract information related to the perceptions of the interviewees toward their contracts, their way of drafting and managing them, and eventual previous experiences with cloud-based third party services that manage legal contracts

Table 4.2: Sections of the version of the supplier's questionnaire for this phase

The next step involved *scheduling a series of interviews with various local suppliers.* At the end of the questionnaire, I asked the interviewees to rank each problem in relation to the perceived impact that it had on the supplier's organisation. The results of the latter question were used as a criterion to assess the relevance of the problem for the majority of suppliers.
Most of the suppliers showed interest in P2, claiming to *experience recurring situations related to this problem with many of their customers.* Specifically, they described *scenarios in which they were in weak positions* or in which *their way of working was hindered by the behaviour of the acquirer.*
The results obtained from the previous step, in addition to the evaluation of the other criteria in the list, led to the selection of P2 as the most relevant problem to studied in this thesis.
The **last objective** involved the *analysis of the problem P2.*
The path to achieve it required a series of four steps.

Step 1 - Systematic review of the state of the art related to P2

For this specific problem, I selected four main articles that discussed the same or closely related themes.

The first study, "A framework for identifying software project risks", was conducted by Mark Keil, Paul E. Cule, Kalle Lyytinen, and Roy C. Schmidt [28]. It describes *a framework for classifying software risks according to the perceived level of control exercisable by the project manager in certain activities and the perceived relative importance of risk.* For this case study, the most relevant category is customer mandate risks, over which the project manager has little or no control.

The second work that was selected was "Agile customer engagement: A longitudinal qualitative case study" by Geir Kjetil Hanssen Tor Erlend Fægri [20], described the *benefit in the transition of a small organisation from waterfall-based methodologies to agile methodologies* (evolutionary project management). In particular, it highlighted various drawbacks in customer engagement when applying the waterfall model.

The third and most similar research, "Customer-Supplier Issues in Software Development" by Aparna Pasi, Kari Smolander, and Uolevi Nikula [36], presented a qualitative analysis to identify the main causes of issues between acquirers and suppliers in software development. Their research was carried out through the usage of grounded theory as well. This work identified **three main categories** that could be the cause of issues between customers and suppliers in software development: *knowledge transfer difficulties, changes brought during the projects, and customer dominance.*

In conclusion, the fourth work considered was "Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers' Perspective" by Lubna Siddique and Bassam Hussein [44]. This last article described nine main categories that could be identified as the causes of conflicts between stakeholders in the context of agile software development in Norway. In particular, according to the authors, **the main causes of conflicts** in this scenario should have been investigated in: *the role of the product owner, an inexperienced project manager, the customer lack of knowledge about methodology, organisational hierarchy in public organisations, contracting, scope aspect personal egos, financial issues, and not getting the right team.*

Step 2 - Selection of the methodology

The *next step* to be undertaken implied the *selection of the methodology that would have been used to analyse the problem.* I decided to opt for a qualitative approach, as it was more suitable to investigate the cause/effect relationships offered by this scenario and because quantitative approaches would have been highly impacted by biased or incorrect data. The suitable choices for the qualitative method to be applied included the phenomenological method, the ethnographic model, and the grounded theory. I decided to opt for grounded theory, a research method used to identify cause/effect observations through iterations and constant comparative analysis. Therefore, grounded theory allows one to derive a theory directly from the gathered data.

Step 3 - Design of questionnaires for both acquirers and suppliers

The *third step* provided the creation of a *revised supplier's questionnaire and the development of an acquirer's questionnaire.*
Both would have been used to perform data sampling on both sides.
The new iterations on the supplier's questionnaire provided the following structure.

Section	**Description**
`Interviewee profiling (unchanged)`	Highlight the main characteristics of the interviewee's organisation, to the extent of relating the characteristics of the supplier to their needs and expectations
`Investigation of interest towards problem P2 (unchanged)`	Collect supplier's perceptions towards their customers, their behaviour, and the consequences that the latter has on the project
`Client relationship management`	Gather data about customer-related issues that were progressively emerging from previous interviews

Table 4.3: Sections of the revised version of the supplier's questionnaire

The first version of the questionnaire on the acquirer's side was structured as follows

Section	**Description**
`Acquirer's profiling`	Analysis of the correlation between the characteristics of the acquirers and their needs and expectations
`Analysis of customer-supplier interactions during the project`	Analysis of aspects such as perceived need to develop the project, effort required to complete project tasks, perceptions towards change requests and towards project phases
`Analysis of the complexity perceived by the customer in performing assigned activities`	Analysis of the aspects such as perceived difficulty to complete assigned tasks and presence of intermediaries or IT figures in the project

Table 4.4: Sections of the first version of the acquirer's questionnaire

Step 4 - Application of grounded theory

The *final step* of this thesis involved the *application of the grounded theory* method to this case study. In particular, I conducted a series of *interviews with both acquirers and suppliers* to gather data according to what the methodology prescribes. Through the application of the principles and iteration of the three main steps (open coding, axial coding, and selective coding) provided by grounded theory, I was able to identify *seven main categories that could cause issues between customers and suppliers in custom software development.*

Category	**Description**
`Customer dominance`	Power relationship that exists between a supplier and a customer, it describes the situations where the customer can influence the supplier's behaviour
`Improper change management`	Situations in which changes may obstruct or disrupt the development of the project
`Knowledge transfer barriers`	Situations that can hinder the successful completion of the information exchange process between the organisations involved in the project development

Negative factors on customers’attitude	Issues in the customer’s attitude and behaviour that may represent a threat in the development of a custom-made software product
Budget dominance	Situations in which the budget becomes the predominant constraint of the project
Poor definition of concerns	Situations in which the definition of roles and responsibilities is not clear to at least one of the stakeholders involved
Low experience in suppliers’ selection	Issues related to the lack of experience in the selection of suppliers shown by the acquirer

Table 4.5: Categories identified by this study

4.2 Limitations of the study

The development of this project involved the use of a qualitative research method to study the behaviour of customers and suppliers in the development of custom software. In particular, grounded theory was used to the extent of analysing the conditions that could cause known issues in software development, the measures undertaken by the suppliers to respond to these problems, and the consequences of these actions. Consequently, the results might be subject to the limitations shown by the methodology.

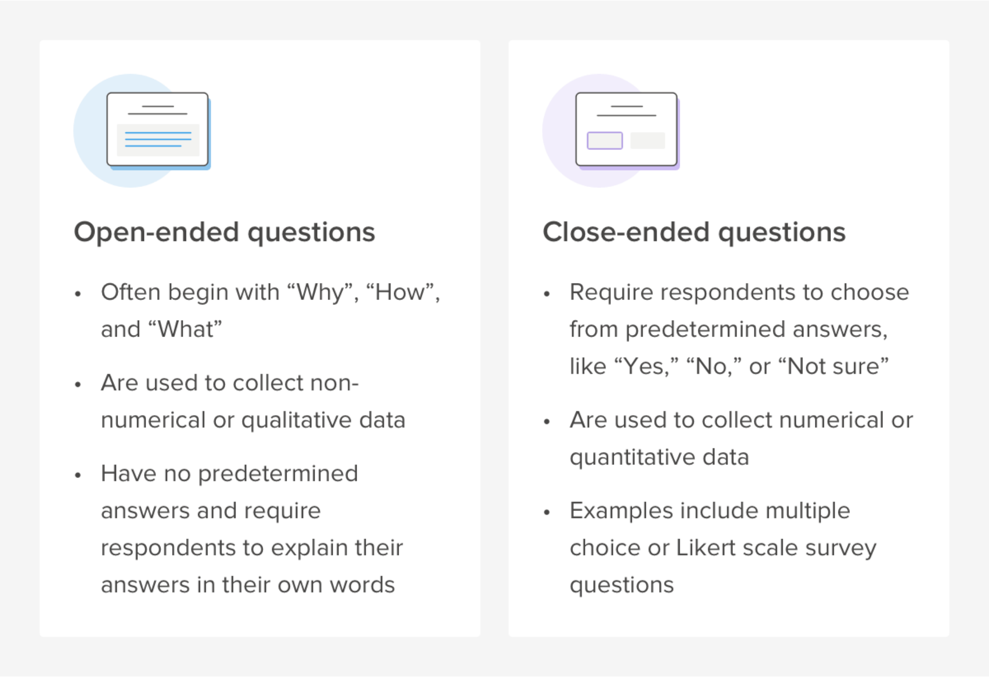
In this case study, the results of a study conducted using this methodology can be influenced by bias arising from data sampling and interpretation, lack of experience in applying grounded theory, influence of an insufficient knowledge base on theoretical sensitivity, and lack of time resources for data sampling and data analysis. The latter condition may also be referred to as the inability to reach theoretical saturation on some aspects of the research. [60]

Taking into account all these factors, I tried to mitigate as much as possible the risks of incorporating one or more limiting factors in this analysis.

Bias in the results

Bias is classified into two main categories: *respondent bias* and *researcher bias*. The former refers to any situation in which the interviewee’s responses are not an accurate expression of their thoughts or feelings. This type of bias can manifest itself in three main ways: social desirability bias (the desire to align answers regarding sensitive questions in a socially acceptable way), agreement bias (the desire to please the interviewer or moderator, leading to answers that they think the researcher is looking for), and sponsor bias (answers are influenced by opinionated views about the research sponsor). In order to limit these factors, the researcher should ask indirect and open-ended questions and should not give away sponsor details [40].

To avoid this form of bias, I tried as much as possible to insert open questions where necessary. However, in some cases a close-ended question was necessary to introduce the topic, understand whether the interviewee understood the topic, and if the interviewee’s



Open-ended questions

- Often begin with "Why", "How", and "What"
- Are used to collect non-numerical or qualitative data
- Have no predetermined answers and require respondents to explain their answers in their own words

Close-ended questions

- Require respondents to choose from predetermined answers, like "Yes," "No," or "Not sure"
- Are used to collect numerical or quantitative data
- Examples include multiple choice or Likert scale survey questions

Figure 4.1: Open vs close ended questions, *Source:* [43]

situation was suitable for the topic. As categories were defined, it was easier to enter almost exclusively open-ended questions, as shown in paragraphs 2.4.2, 2.4.4, 2.4.5 and 2.4.6.

The latter includes confirmation bias (data interpretation mislead by hypothesis support and removal of unfavourable data), question order bias (question order that impacts the way the succeeding questions are answered), cultural bias (conclusion based only on the researcher's cultural lens), halo/horn effect (assumptions about a respondent due to a positive or negative attribute), and question wording bias (asking questions that only lead respondents in one particular direction).

In order to contain these factors, the researcher should use multiple collaborators to interpret the data, conduct an external review of the work, consider his/her role in the study (intended as the need to remain impartial), use data triangulation (intended as the analysis of secondary sources to verify if your primary data are valid and reliable), and ask participants to evaluate the findings [40].

To reduce this form of bias, I conducted a periodic review with some peers from my company and the university and with some previously interviewed participants.

I was unable to involve multiple collaborators to analyse data due to resource constraints. However, many blog posts, articles and dedicated pages on other suppliers' websites allowed me to perform a solid data triangulation.

Lack of experience in applying grounded theory

This factor may represent the most limiting aspect of this thesis, as this was my first qualitative research on a large scale. Furthermore, this was my first research that applied the grounded theory method. Therefore, inexperience might represent a limit for this thesis.

Influence of insufficient knowledge base on theoretical sensitivity

This type of issue could be referred to as lack of experience in the sector of the problem. This aspect was already considered and minimised during the problem selection described in paragraph 2.2.8. As a project manager, in the past, I had to deal with some situations related to P2. However, not all these problems were immediately clear to me and I had to investigate more on each of them to expand my knowledge base.

Lack of time resources for data sampling and data analysis

This issue was verified in trying to analyse in depth the category "negative factors of customers' attitude", especially for what concerned the subcategory lack of an immediately available way to track progress and sustained costs. In particular, I was unable to schedule more interviews with other acquirers, both due to the unavailability of potential interviewees and due to time constraints. Consequently, theoretical saturation was not reached on this aspect.

Limitations of the data set

In conclusion, some characteristics of the data set might be considered as limiting factors for this thesis.

In particular, the final data set of interviewees was unable to reflect all desirable features described in paragraph 2.4.1 and paragraph 2.4.3. In particular, the suppliers' data set mostly contained small and medium-sized Italian organisations, while the acquirers' data set contained quite little data for a more structured analysis. However, I preferred not to contact directly customers bypassing the role of intermediary of the supplier, as I wanted to make sure (as much as possible) that the interviewed acquirers would reflect the desirable features of the data set.

4.3 Outlook

One of the limitations shown by grounded theory is the impossibility to replicate the study, as opinions, beliefs, and people may change over time (influencing data analysis).

However, this study predicted some limitations related to the category "negative factors of customers' attitude". Furthermore, I wanted to explore more in depth some additional topics. In particular, they included

- The activities and the phases of the project that were perceived as more or less valuable by the acquirers
- Expectations and concerns of acquirers about the project

- The way acquirers can express their requirements and how a project manager can help them
- Perceptions and way of working of acquirers in monitoring the progress of the project and its costs

To do so, I iterated on the questionnaire producing a new version that I was able to submit to only one interviewee.
This questionnaire also wanted to study other aspects related to acquirer's decision-making, trying to make more interactive activities rather than simple questions.
In this version, five sections were included

Activities and phases perceived as more or less valuable by acquirers

ID	**Question**
QA18	From the list of the following aspects, you can choose only 5: 1. Requirements analysis 2. Interface design 3. Project management 4. Feasibility study 5. Software architecture design 6. Automatic tests 7. Project documentation 8. Project progress monitoring 9. Quality check 10. Training on the product
QA19	You are given a budget of 100 thousand euros, allocate it for the 5 aspects you listed above
QA20	Describe what needs to be done for each for these aspects and the reasons of your choices

Table 4.6: Questions related to the perception of acquirers towards certain aspects of the project

The objective of this set of questions included the understanding of which are the aspects that customers evaluates more, an estimate of the gap in value between these aspects and why did they make this choices. Furthermore, it is interesting to observe whether customers understand all the activities and tasks beyond a certain aspect of the project.

Expectations and concerns of acquirers about the project

ID	**Question**
QA21	1- What are your expectations towards a software project? Have you already had to deal with project examples carried out by other acquaintances of yours (only for customers at their first project)?
QA22	What are your main concerns regarding the project? In which phases (planning, analysis, design, implementation, testing, maintenance) would you like to be followed the most?

Table 4.7: Questions related to the expectations and concerns of acquirers about the project

This set of questions aimed to analyse which are the main expectations of customers towards their project and if other acquaintances' experiences might influence their opinions. Furthermore, it is interesting to explore their main fears and in which phases they would like to be assisted more.

The way acquirers can express their requirements and how a project manager can help them

ID	**Question**
QA23	How do you define the requirements? Do you generally get help in doing this? From who?
QA24	How do you decide what will be in the project and what will not?
QA25	How do you plot your requirements (in mind, on a sheet of paper, on a spreadsheet)
QA26	How do you establish the importance each requirement has for you?

Table 4.8: Questions related to the way acquirers can express their requirements and how a project manager can help them

This set of questions aimed to analyse the way of working of the customers in defining requirements, their parameters to decide what to include and what not in a project and their of way of reporting requirements. Moreover, the last question aimed to explore how the customers assign priorities and importance to each requirement.

Perceptions and way of working of acquirers in monitoring the progress of the project and its costs

ID	Question
QA27	How do you understand at what stage the project is?
QA28	How do you know how much you are spending?
QA29	How do you plot your requirements (in your mind, on a sheet of paper, on a spreadsheet)
QA30	How do you track the progress of the project? Does the supplier provide you with a dedicated tool? Which one? How do you find them?

Table 4.9: Questions related to the perceptions and way of working of acquirers in monitoring the progress of the project and its costs

This conclusive section aimed to analyse the perceived importance in monitoring costs and project progress shown by the acquirers, their way of working and their perceptions towards the tools that are used.

4.4 Final notes

This thesis provided a qualitative study aimed at investigating the main issues that occur in the relationships between acquirers and suppliers in the development of custom software. This problem and its implications remain quite relevant for the current market situation, as suppliers might be influenced by factors that affect both economic and software development.

On the one hand, dealing with intended or unintended adverse factors shown by customers can be very stressful and dangerous to suppliers, as they may face losses, overwork, and unpleasant situations.

On the other hand, customers need to deal with the risks arising from developing complex software products and the need to trust an external organisation for this scope.

Nevertheless, providing possible solutions to these problems is certainly not trivial. In fact, the design of best practises for these situations would require years of study and a noticeable experience in the sector of project and client management.

However, valuable solutions should respond to well-defined problems. Even defining these issues is non-trivial, as data must be gathered from the unbiased opinions of the stakeholders involved in the development of a project.Stakeholders may perceive the need to conceal their behaviour, as other people could consider it strange or inappropriate.

However, collecting biased data only deflects the proper definition of the problem, delaying by consequence the definition of solutions and best practises to address them, and this is why contributing to the proper definition of these issues was exactly the objective of this thesis.

Acronyms

CMS Content Management System. 68, 89

GDPR General Data Protection Regulation. 64, 89

MVP Minimum Viable Product. 68, 89

POC Proof of Concept. 61, 90

ROI Return on Investment. 22, 90

ROR Rate of Return. 22, 90

SDLC Software Development Life Cycle. 90

T& M Time & Materials. 20, 90

Glossary

CMS A content management system (CMS) is software that helps users create, manage, and modify content on a website without the need for technical knowledge. In other words, a CMS lets you build a website without needing to write code from scratch (or even know how to code at all).[54]. 88

Dedicated team With this type of contract the client gets a team of software development professionals fully focused on the project, works closely with them to define the project roadmap, either participates in managing the team or delegates management to the vendor, and has full control over the software product.This model is flexible and fully adaptable to any change [9]. 20, 89

Earned Value Earned value is a way to measure and monitor the level of work completed on a project against the plan. It is a quick way to tell if the project behind schedule or over budget on your project. The earned value in a project is calculated by multiplying the percentage complete by the total project budget.[24]. 22, 89

Fixed price In a fixed-price contract, both parties must agree from the outset on the detailed project scope, requirements, budget, and timeline. In addition, this is a rather expensive model of cooperation, since the vendor assumes all risks and bears full responsibility for the project's success. This is the least flexible model, since any deviation from the plan requires a complex change request approval process[9]. 20, 89

GDPR The General Data Protection Regulation (GDPR) is a legal framework that sets guidelines for the collection and processing of personal information from individuals living and outside of the European Union (EU). Its aim is to give consumers control over their own personal data by holding companies responsible for the way they handle and treat this information. The regulation applies regardless of where websites are based, which means it must be heeded by all sites that attract European visitors, even if they do not specifically market goods or services to EU residents. [17]. 88

MVP A minimum viable product (MVP) is the release of a new product (or a major new feature) that is used to validate customer needs and demands prior to developing a more fully featured product. To reduce development time and effort, an MVP includes only the minimum capabilities required to be a viable customer solution.[35]. 88

POC Proof of concept is evidence obtained from a pilot project, which is executed to demonstrate that a product idea, a business plan or a project plan is feasible. A proof of concept collects user feedback and insights from your team members, including those who might not have otherwise contributed, thereby mitigating unforeseen risks. [34]. 88

Return on Investment Return on investment is a ratio that divides the net profit (or loss) from an investment by its cost. Because it is expressed as a percentage, you can compare the effectiveness or profitability of different investment choices. [13]. 88

Rate of Return A rate of return is the net gain or loss of an investment over a specified time period, expressed as a percentage of the investment's initial cost. [29]. 88

SDLC The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond. The software development lifecycle methodology provides a systematic management framework with specific deliverables at every stage of the software development process. As a result, all stakeholders agree on software development goals and requirements upfront and also have a plan to achieve those goals. [57]. 9, 88

T & M By signing a time and materials contract, an acquirer agrees to pay for the time the team: the number of hours multiplied by the hourly rate [9]. 88

Bibliography

Consulted references

[1] (Restricted access). *ISO/IEC/IEEE International Standard - Systems and software engineering – Software life cycle processes.* 2017, pp. 1–157. DOI: 10.1109/IEEESTD.2017.8100771 (cit. on pp. 3, 4).

[16] Janneke Frambach, Cees Van der Vleuten, and Steven(Restricted access) Durning. *AM Last Page: Quality Criteria in Qualitative and Quantitative Research.* Vol. 88. Apr. 2013, p. 552. DOI: 10.1097/ACM.0b013e31828abf7f. URL: https://www.researchgate.net/publication/236083370_AM_Last_Page_Quality_Criteria_in_Qualitative_and_Quantitative_Research (cit. on pp. 37, 38).

[18] Barney G. Glaser and Anselm L. Strauss. *The Discovery of Grounded Theory, Strategies for Qualitative Research.* Aldine Transactions, 2006, reprinted. URL: http://www.sxf.uevora.pt/wp-content/uploads/2013/03/Glaser_1967.pdf (cit. on p. 41).

[22] Judith Holton. *The Coding Process and Its Challenges.* Vol. 9. Feb. 2007, pp. 21–40. DOI: 10.4135/9781848607941.n13 (cit. on p. 52).

[47] Juliet M. Strauss Anselm L.; Corbin. *Basics of Qualitative Research : Techniques and Procedures for Developing Grounded Theory.* Sage Publications, Inc., 1998. URL: https://research-proposal.ir/wp-content/uploads/2019/06/Basics_of_Qualitative_Research__Techniques_and_Procedures_for_Developing_Grounded_Theory.pdf (cit. on pp. 52, 54).

[59] Karl E Wiegers and Joy Beatty. *Software Requirements 3.* USA: Microsoft Press, 2013. ISBN: 0735679665 (cit. on p. 13).

Consulted websites

[2] Judee Reel (Restriced access). *Critical Success Factors in Software Projects.* 1999. URL: https://api.semanticscholar.org/CorpusID:206448427 (cit. on p. 13).

[3] Tareq Aljaber. *Iron triangle project management and agile.* URL: https://www.atlassian.com/agile/agile-at-scale/agile-iron-triangle (cit. on pp. 11, 12).

[4] Rosemarie Anderson. *Thematic Content Analysis (TCA) Descriptive Presentation of Qualitative Data*. URL: https://rosemarieanderson.com/wp-content/uploads/2014/08/ThematicContentAnalysis.pdf (cit. on p. 15).

[5] Mikel Anwar. *How Much Does Custom Software Development Cost for the Industry?* URL: https://www.consultingwhiz.com/blog/custom-software-development-cost-for-industry/ (cit. on p. 20).

[6] Jitu Bhaskar. *A Complete Guide to Custom Software Development*. URL: https://semidotinfotech.com/blog/custom-software-development-guide/ (cit. on p. 19).

[7] Barry Boehm. *Get Ready for Agile Methods, With Care*. Feb. 2002. DOI: 10.1109/2.976920 (cit. on p. 13).

[8] Narciso Cerpa and June M. Verner. *Why Did Your Project Fail?* Dec. 2009. DOI: 10.1145/1610252.1610286. URL: https://doi.org/10.1145/1610252.1610286 (cit. on p. 13).

[9] Svetlana Cherednichenko. *Software Development Contract: Types and Main Points to Include*. URL: https://medium.com/mobindustry/software-development-contract-types-and-main-points-to-include-691aead995c4 (cit. on pp. 89, 90).

[10] Mike Cottmeyer. *Sharing risk with the team*. URL: https://www.leadingagile.com/2010/12/sharing-risk-with-the-team/ (cit. on p. 26).

[11] *Custom Software Development Market Size, Share & Trends Analysis Report By Solution (Enterprise Software, Web-based Solutions), By Deployment, By Enterprise Size, By End-use, By Region, And Segment Forecasts, 2022 - 2030*. URL: https://www.grandviewresearch.com/industry-analysis/custom-software-development-market-report#:~:text=The%20global%20custom%20software%20development,22.3%25%20from%202022%20to%202030. (cit. on p. 2).

[12] *Data analysis*. URL: https://aph-qualityhandbook.org/set-up-conduct/process-analyze-data/3-1-qualitative-research/data-analysis/ (cit. on p. 37).

[13] Benjamin Curry Emily Guy Birken. *Return On Investment (ROI)*. URL: https://www.forbes.com/advisor/investing/roi-return-on-investment/ (cit. on p. 90).

[14] *Ethnographic Research*. URL: https://delvetool.com/blog/ethnography (cit. on p. 40).

[15] *Expressions in ISO International Standards and other normative ISO deliverables [Technical Specifications (TS), Publicly Available Specifications (PAS), International Workshop Agreements (IWA)]*. URL: https://www.iso.org/foreword-supplementary-information.html (cit. on p. 11).

[17] Jake Frankfield. *General Data Protection Regulation (GDPR) Definition and Meaning*. URL: https://www.investopedia.com/terms/g/general-data-protection-regulation-gdpr.asp (cit. on p. 89).

[19] Siften Halwai. *How Bespoke Software is nothing like Off-The-Shelf Software?* URL: https://www.sigmadatasys.com/how-bespoke-software-is-nothing-like-off-the-shelf-software/ (cit. on p. 2).

[20] Geir Hanssen and Tor Fægri(Restricted access). *Agile customer engagement: A longitudinal qualitative case study.* Sept. 2006. DOI: 10.1145/1159733.1159759 (cit. on pp. 13, 14, 34, 73, 79).

[21] Marta Hlova. *Software development outsourcing: how to avoid contract loopholes.* URL: https://www.n-ix.com/software-development-outsourcing-how-avoid-contract-loopholes/ (cit. on p. 21).

[23] Lissie Hoover. *What Is Qualitative vs. Quantitative Study?* URL: https://www.gcu.edu/blog/doctoral-journey/what-qualitative-vs-quantitative-study (cit. on p. 37).

[24] *How to Calculate Earned Value in Project Management.* URL: https://www.wrike.com/project-management-guide/faq/how-to-calculate-earned-value-in-project-management/ (cit. on p. 89).

[25] *How To Do Axial Coding with Examples.* URL: https://delvetool.com/blog/axialcoding (cit. on p. 54).

[26] https://www.scribbr.com/author/raimo/. *Qualitative vs. Quantitative Research | Differences, Examples & Methods.* URL: https://www.scribbr.com/methodology/qualitative-quantitative-research/ (cit. on p. 36).

[27] Magne Jørgensen and Kjetil Moløkken-Østvold (Restricted access). *How large are software cost overruns? A review of the 1994 CHAOS report.* Apr. 2006. DOI: 10.1016/j.infsof.2005.07.002 (cit. on p. 13).

[28] Mark Keil et al. *A Framework for Identifying Software Project Risks.* Nov. 1998. DOI: 10.1145/287831.287843. URL: https://doi.org/10.1145/287831.287843 (cit. on pp. 13, 33, 34, 73, 79).

[29] Will Kenton. *Rate of Return (RoR) Meaning, Formula, and Examples.* URL: https://www.investopedia.com/terms/r/rateofreturn.asp (cit. on p. 90).

[30] Sarah Laoyan. *Organizational silos: 4 common issues and how to prevent them.* URL: https://asana.com/it/resources/organizational-silos (cit. on p. 71).

[31] Sarah Laoyan. *What is Agile methodology? (A beginner's guide).* URL: https://asana.com/resources/agile-methodology (cit. on p. 10).

[32] Zach Lazzari. *What Is a Contingency Factor?* URL: https://bizfluent.com/what-is-a-contingency-factor.html (cit. on p. 25).

[33] Timo O.A. Lehtinen et al. *Perceived causes of software project failures – An analysis of their relationships.* 2014(Restricted access). DOI: https://doi.org/10.1016/j.infsof.2014.01.015. URL: https://www.sciencedirect.com/science/article/pii/S0950584914000263 (cit. on p. 13).

[34] William Malsam. *What Is Proof of Concept (POC)? Examples for Business, Software & More.* URL: https://www.projectmanager.com/blog/proof-of-concept-definition (cit. on p. 90).

[35] *Minimum Viable Product.* URL: https://www.gartner.com/en/marketing/glossary/minimum-viable-product-mvp- (cit. on p. 89).

[36] Aparna Pasi, Kari Smolander, and Uolevi Nikula. *Customer-Supplier Issues in Software Development.* Jan. 2008. URL: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1089&context=ecis2008 (cit. on pp. 14, 35, 58, 60, 63, 64, 74, 79).

[37] *Phenomenological Research Design.* URL: https://delvetool.com/blog/phenomenology (cit. on p. 39).

[38] *Practical Guide to Grounded Theory Research.* URL: https://delvetool.com/groundedtheory (cit. on p. 41).

[39] Chavapong Prateep. URL: https://www.vecteezy.com/vector-art/7413163-the-waterfall-model-infographic-vector-is-used-in-software-engineering-or-software-development-processes-the-illustration-has-6-steps-like-agile-methodology-or-design-thinking-for-application-system (cit. on p. 9).

[40] Louise Principe. *8 Ways to Rule Out Bias in Qualitative Research.* URL: https://www.civicommrs.com/8-ways-to-rule-out-bias-in-qualitative-research/ (cit. on pp. 82, 83).

[41] Nayan B. Ruparelia (Restricted access). *Software Development Lifecycle Models.* May 2010. DOI: 10.1145/1764810.1764814. URL: https://doi.org/10.1145/1764810.1764814 (cit. on pp. 9, 18, 28).

[42] *SDLC models.* URL: https://www.javatpoint.com/software-engineering-sdlc-models (cit. on p. 18).

[43] Mitali Shukla. *Open-ended questions: Definition, examples, and tips.* URL: https://delighted.com/blog/open-ended-questions (cit. on p. 83).

[44] Lubna Siddique and Bassam Hussein. *Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers' Perspective.* Aug. 2016. DOI: 10.32738/JEPPM.201607.0005 (cit. on pp. 14, 36, 74, 80).

[45] David Woodruff Smith. *Phenomenology.* URL: https://plato.stanford.edu/entries/phenomenology/ (cit. on p. 38).

[46] *Software Companies Italy.* URL: https://bolddata.nl/en/companies/italy/software-companies-italy/ (cit. on p. 58).

[48] Indeed Editorial Team. *Project Selection: Definition, Importance and Steps.* URL: https://www.indeed.com/career-advice/career-development/project-selection#:~:text=Project%20selection%20is%20the%20evaluation,return%20on%20undertaking%20a%20project. (cit. on p. 22).

[49] Linda Wallace and Mark Keil. *Software Project Risks and Their Effect on Outcomes.* Apr. 2004. DOI: 10.1145/975817.975819. URL: https://doi.org/10.1145/975817.975819 (cit. on p. 13).

[50] *Waterfall change control process.* URL: https://www.mindk.com/blog/change-control-process/#c (cit. on p. 10).

[51] *Waterfall Methodology is Dead? Long Live Waterfall!* URL: https://reqtest.com/en/knowledgebase/what-is-waterfall-methodology/ (cit. on pp. 10, 66).

[52] Don Wells. *Extreme Programming: A gentle introduction.* URL: http://www.extremeprogramming.org/ (cit. on p. 28).

[53] *What are the Six types of qualitative research?* URL: https://www.acumenfieldwork.com/what-are-the-six-types-of-qualitative-research/ (cit. on p. 38).

[54] *What Is a Content Management System (CMS)?* URL: https://kinsta.com/knowledgebase/content-management-system/ (cit. on p. 89).

[55] *What is a core product? (Definition, uses and examples).* URL: https://uk.indeed.com/career-advice/career-development/core-product (cit. on p. 61).

[56] *What is Scrum?* URL: https://www.scrum.org/learning-series/what-is-scrum/what-is-scrum (cit. on pp. 15, 28).

[57] *What Is SDLC (Software Development Lifecycle)?* URL: https://aws.amazon.com/what-is/sdlc/ (cit. on p. 90).

[58] *What is Selective Coding?* URL: https://delvetool.com/blog/openaxialselective#selective (cit. on p. 57).

[60] *Your complete guide to grounded theory research.* URL: https://www.qualtrics.com/uk/experience-management/research/grounded-theory-research/ (cit. on pp. 42, 82).