POLITECNICO DI TORINO

Master's Degree in Engineering and Management

Master's Degree Thesis

Project management in the consultancy sector: comparing Waterfall and Agile approaches

Supervisor

Prof.ssa Anna Corinna CAGLIANO

Company Advisor

Dott.ssa Lorenza PALUMBO

Candidate

Lorenzo FOSCHINI

Aprile 2021

Summary

In modern society, project management has emerged as an essential component of any serious business development initiative. It has also become a key to success in a global economic environment where companies are constantly seeking a competitive advantage over their competitors. However, traditional companies do not have the capacity to select and manage countless projects with managerial and strategic benefits and therefore turn to outside professional help, usually in the form of consultancy firms.

The aim of this study is to understand how the discipline of project management is applied in consultancy firms and to study the main advantages and disadvantages of the approach adopted. To this end, an actual technological project carried out by Accenture and the methodology used for it are analysed, with references to the Waterfall and Agile models common in project management theory. Finally, a solution is proposed to overcome some of the critical issues encountered in the course of the project, including the pandemic that has disrupted people's lives over the last few months and the increased use of smart working.

Acknowledgements

I believe it is my duty, before proceeding to the actual discussion, to dedicate this small space to the people who have contributed, voluntarily or otherwise, to the realisation of this work.

First of all, I would like to thank my supervisor, Professor Cagliano, for the knowledge she imparted and for the honesty and practicality with which she offered me indispensable advice.

Special thanks go to my co-relator and tutor for the Accenture internship, Lorenza Palumbo and also to my Accenture boss Luca Bertoni, for helping me settle into this new and first work experience, for following and advising me throughout the project and for teaching me so many things in such a short time.

I would also like to thank the Accenture project team, for the great work we did together during these months of internship.

I *am infinitely grateful to my parents and my brother, with whom I lived during this time of pandemic, for supporting me at all times and continuing to hold my hand throughout my life. I am sure you will have the strength to let me go now that my working life will begin.*

Finally, I dedicate this thesis to myself, to my sacrifices and my tenacity that have allowed me to get this far.

Table of Contents

List of Tables

List of Figures

Acronyms

PM
: Project Management

PMI
: Project Management Institute

PMBOK
: Project Management Body Of Knowledge

ISO
: International Organization for Standardization

IPMA
: International Project Management Association

ICB
: Individual Competence Baseline

GPM
: Green Project Management

SDLC
: System Development Life Cycle

CPM
: Critical Path Method

PRiSM
: Projects Incorporating Sustainable Methods

PRINCE2

Projects IN Controlled Environments

UAT

User Acceptance Test

ARERA

Autorità di Regolazione per Energia Reti e Ambienti

REMSI

Regolazione della Morosità nel Servizio Idrico Integrato

RCM

Requirements Change Management

Introduction

The discipline of project management represents a fundamental element in the academic studies of a management engineer, which is not only applied to a single work sector, but manifests itself, as an ideology and mental approach, in all sector, as well as in everyday life.

Project management is, for instance, at the heart of the everyday working process of a consultancy firm. An increasing number of students in recent years, especially engineers and economists, is choosing a career in consultancy as their first work experience because of the wide range of topics it covers and the future opportunities that this sector offers. In fact, the consultancy market is currently one of the fastest emerging and growing industries, with a turnover, just in Italy, of almost 4.5 billion euros. This is mainly due to the ever-increasing competition that is arising between industries and the drive towards digitalisation and innovation, that are required to survive in the world of business today. It therefore becomes essential for these companies to ask for external expertise in strategy and management in order to outperform all competitors in this technological evolution.

This master's thesis paper aims to address these two topics, project management and consultancy, through the eyes of an engineering management student who has just entered the job market in one of the world's largest technology and strategy consulting companies. Specifically, the project methodology chosen and used by the company Accenture is analysed and studied, comparing it with the two most important project methods that exist in the world of project management: the Waterfall and Agile models.

In particular, the work is divided into four chapters.

Chapter 1 introduces in a theoretical way what project management is and what are its most important standards and models in the world. It is further divided into 6 sections. Specifically, the first and second sections define project management discipline and presents the main standards, guidelines and methodologies used worldwide. Section 3 studies and explores the Waterfall model and describes its advantages and disadvantages. Section 4 applies the same study with the Agile model. The fifth section compares the two approaches to project management presented above and provides an indication of a possible transition from one to the other. The theoretical discussion ends with section 6 that lists and summarizes some other significant PM methodologies. Chapter 2 explores the world of Accenture and the technology consulting industry. It consists of two sections: the first presents the evolution of consultancy, while the second narrates the birth and expansion of Accenture around the world and, in greater detail, in Italy. Chapter 3 presents and explains how Accenture approaches project management and how the life cycle of a project is structured. Specifically, it illustrates and details one by one all the stages of development of a project according to the one created by Accenture. Chapter 4 describes the application of this methodology in practice on a project carried out in the world of utilities. It provides a detailed explanation of all the stages and all the steps taken to complete this project. Finally, the main issues that emerged regarding the project methodology are discussed and an improved solution to them is offered.

Chapter 5 concludes the work, first by summarising what has been analysed, and then with an account of the main results. To this end, the main benefits obtained thanks to the transformation of the project methodology and the procedural advantages obtained with the client company are indicated. Finally, the limitations of this study and future objectives are also reported.

Chapter 1

Project Management

The chapter aims at defining the discipline of project management from its inception to its diffusion in today's world. It then aims to analyse and compare two of the most important methodological approaches: the Waterfall and the Agile model.

1.1 Introduction on Project Management

The use of project management as a business process goes back a long time. Indeed, the building of the Egyptian pyramids is believed by many to have been assisted by the use of simple project management principles. However, for much of its history, the predominant application type, in a business environment, was engineering and construction projects. Project management appeared in the military and construction industries, providing schedule and resource data to the top management [1]. Its processes, tools and techniques helped the development of weapons, defence and space projects with the birth of the first atomic bomb during the Second World War and the Apollo moon landing project later on.

Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements. Project Management is achieved through the application and integration of a set of designated and established processes for the administration of the project [2]. Project management is often defined also as the discipline of initiating, planning, executing, and controlling the work of a team to achieve specific goals and meet specific performance measures [3].

More easily, project management is about converting vision into reality. There is a vision of some future state someone wants to achieve. It may be everything, from a new computer system, to a new product, or more competent managers. It's foreseen that the operation of that new state will help to improve the business' performance or the current state in general, by solving a problem or exploiting an opportunity, and so provide with benefits that will repay the cost of achieving it. Project Management is the structured process of successfully delivering that future state [4]. As a consequence, this future desirable condition can be achieved by studying and handling with change. The need to manage change through projects, and so the management of projects, touches all the aspects of everyday life, both in working and social environments.

The PMI says that a project is a “temporary endeavor undertaken to create a unique product or service” [2]. However, projects may come in many guises. As for project management, there are traditional major projects from heavy engineering and industries. These are significant endeavours involving large dedicated teams, often requiring the collaboration of several sponsoring organizations. But the projects with which most people are involved are smaller.

Nowadays, product development times and market windows shrank, causing the requirement of new products to be introduced more quickly and effectively. So, organizations must adopt flexible structures to respond to the changing environment and, especially, in order to gain competitive advantage. Many clients expect every product to be made to a bespoke design, and so every product becomes a mini project. The project-oriented organization is now common and project management is the new general management. It involves much more, and people in every industry and every country manage projects. In the last years, it has come to support about 30% of world economic activity [5]. Not only physical products but much of the activities is due to modern applications of project management, for example, information technology systems implementations, software development, and research and development. Thus, projects at work may include engineering or construction projects for the construction of new facilities, the maintenance of existing facilities, the implementation of new technologies or IT systems, research, product development and launch, or management development or training programmes.

1.2 Global project management standards and guides

Today, in the world of project management, there are numerous professional societies and associations that support project management and promote standards, methodologies and good practice in general. It is therefore necessary to briefly list and study the most important standards and guides globally in this sense.

However, before starting the digression on the project management world scene, it would be better to clarify a few terms and definitions that will be used extensively later on. Above all, let's differentiate what is considered a standard, compared to a method(ology) or a guide. A standard is "something used as a measure, norm, or model in comparative evaluations". In this scenario, a standard is considered as a document adopted by an authority, a general or customised content that can be used as a model or example. On the other hand, a method is "a particular procedure for accomplishing or approaching something," whereas a methodology is "a system of practices, techniques, procedures, and rules used by those who work in a discipline". Finally, a guide refers to "a guideline or an approach that an organization can adopt for a specific practice" [6].

1.2.1 Project Management Body Of Knowledge

The Project Management Institute (PMI) was founded at the Georgia Institute of Technology in 1969 as a nonprofit professional organization for Project Management. Nowadays, the PMI serves more than 2.9 million professionals including over 600,000 members in 214 countries and territories around the world. Its services include the development of standards, research, education, publication, networking-opportunities in local chapters, hosting conferences and training seminars, and providing accreditation in project management [2]. Its main objectives at the time of the foundation was described as:

- Foster recognition of the need for professionalism in project management;
- Provide a forum for the free exchange of project management problems, solutions and applications;
- Coordinate industrial and academic research efforts;
- Develop common terminology and techniques to improve communications;
- Provide interface between users and suppliers of hardware and software systems;

- Provide guidelines for instruction and career development in the field of project management.

However, the most important contribute was to standardize project management procedures and approaches, with the introduction of the first Project Management Body of Knowledge (PMBOK) in 1996. It is the project management reference with an extensive global distribution. Its content summarizes many concepts obtained from decades of professional practice by the volunteers who wrote it and regularly update it. The evolution of the PMBOK Guide is reflected in editions of the guide, as can be seen from figure 1.1. The main changes over the years are referred to processes, knowledge areas and process groups, that can be also considered the founding principles of the guide.

Table 1.1: PMBOK Guide Changes [6]

Year	Edition	PG	KA	Processes	Pages
1996	1996	5	9	37	∼ 180
2000	2000	5	9	39	∼ 210
2004	Third	5	9	44	∼ 400
2008	Fourth	5	9	42	∼ 460
2013	Fifth	5	10	47	∼ 620
2017	Sixth	5	10	49	∼ 790

PG = Process Groups
KA = Knowledge Areas

It is necessary now to explain what process groups and knowledge areas are. In general, both of them, are ways to group or categorize the project Mmnagement 49 processes. Project management process groups are logical grouping of processes to achieve specific project objectives. Sixth edition of PMBOK Guide includes five PG: *initiating, planning, executing, monitoring and controlling, closing.* On the other hand, knowledge areas are identified by its knowledge requirements and described in terms of its component processes, practices, inputs, outputs, tools, and techniques. Sixth edition of PMBOK Guide includes ten KA: *integration, scope, schedule, cost, quality, resource, communications, risk procurement, stakeholder management.* Moreover, other crucial component of the guide are project life cycle and project phases. The life cycle of a project is the series of phases that it passes through from its start to its completion and a project phase is a collection of logically related activities that culminates in the completion of one or more deliverables.

The PMBOK Guide is based on The Standard for Project Management, included as second part of the guide itself, that identifies the processes, and its input and output, that are considered "good practice" on most projects most of the time. PMI specifies that PMBOK Guide is not a methodology for project management, it is instead a recommended reference for tailoring. Tailoring a project means actively selecting the appropriate processes with the aim of managing a project correctly. This selection is necessary, because every project is, by definition, unique.

Moreover, together with PMBOK, PMI introduces the Agile Practice Guide, created in partnership with Agile Alliance [7]. It provides tools, situational guidelines and an understanding of the various Agile approaches available to enable better results. It is especially useful for those project managers accustomed to a more traditional environment to adapt to a more agile approach. The Agile methodology and its principles will be dealt with in detail in section 1.5.

1.2.2 ISO21500

ISO 21500:2012 (Guidance on Project Management), developed by International Organization for Standardization (ISO), is the first guide by ISO specific to project management. It represents the international reference for providing general guidance on project management, explain core principles and what constitutes good practice in project management. It is clear the effort of ISO guidance in ensuring some degree of alignment with PMI and especially with PMBOK. Indeed, they share the concepts of process groups and knowledge areas, called subjects groups by ISO21500. The ISO Guidance on Project Management and the PMBOK Guide presents the same concept of process groups: *initiating, planning, implementing, controlling, and closing.* Moreover, ISO21500 refers to ten subject areas: *integration, stakeholders, scope, cost, time, quality, risk, communication, resources, and procurement.*

ISO 21500 identifies the recommended project management processes to be used during a project as a whole, for individual phases or both. There are defined 39 project management processes, divided in three categories: *project management processes, product processes and support processes.* These processes should be however tailored for each project and not be applied uniformly.

In order for a project to be successful, this international standard [8] lists the following actions to be accomplished:

- Select appropriate processes that are required to meet the project objectives;
- Use a defined approach to develop or adapt the product specifications and plans to meet the project objectives and requirements;
- Comply with requirements to satisfy the project sponsor, customers and other stakeholders;
- Define and manage the project scope within the constraints, while considering the project risks and resource needs to provide the project deliverables;

- Obtain proper support from each performing organization, including commitment from the customers and project sponsor.

1.2.3 International Project Management Association's Individual Competence Baseline

International Project Management Association (IPMA) is a membership-based professional organisation which aim is to develop project management competences in the own geographic areas of influence, interacting with thousands of practitioners and developing relationships with corporations, government agencies, universities and colleges, as well as training organizations and consulting companies [9].

Over the years, a leading IPMA reference has been the Individual Competence Baseline (ICB), version 4.0 of which was in place until 2015.

The IPMA's ICB is therefore the global standard for individual competence in project, programme and portfolio management. ICB does not describe how to manage projects or the processes or steps involved in project, program or portfolio management. Instead, it supports the development of individual competence through the presentation of a complete inventory of competence element across projects, programmes and portfolios. Competences that, if fully realised, represent complete mastery of these management domains. As a consequence, it offers the professional community three competence areas:

- People competences, such as personal and interpersonal competencies;
- Practice competences, specific to projects, program, or portfolios;
- Perspective competences, which would be more about the environment as well as the rationale that leads people, organisations, and societies to start and support projects, programmes, and portfolios.

In total, there are 29 elements of competence. While ISO and PMI offer the study of processes, process groups, and subject areas, IPMA, as said, offers the community the competence elements to enable the successful completion of the projects' work.

1.2.4 Global Green Project Management

Green Project Management (GPM) is a relatively young organisation with a focus on sustainable project implementation. GPM has published in 2019 the second version of the P5$^{\text{TM}}$ standard [10] that addresses the five Ps of *People*, *Prosperity*, *Planet*, *Product* and *Process*. In essence, GPM argues that organisations working on a project should consider the project's impact on people, identified as society and workers and the planet, meaning the environment, along with their considerations of economic prosperity. To translate the P5 Standard into a practical approach, the GPM has published a guide and method under the name PRiSM [6]. The latter is built on the ISO 21500 platform as a basis for project management and adds sustainable concepts and outcomes throughout the project life cycle. In essence, it is a project management method based on sustainable development. The aim is therefore to enable companies to manage their projects by integrating environmental sustainability into their processes and thereby reducing negative ecological and social impacts. The PRiSM method will be discussed in its entirety in the section 1.6.2.

1.3 Projects methodological approach: Waterfall

The purpose of the section is to present and deeply study one of the most important and most known methodological approach of Project Management: the Waterfall model.

1.3.1 Overview

The Waterfall Development model originated in the manufacturing and construction industries, where the highly structured physical environments meant that design changes became prohibitively expensive much sooner in the development process. The first formal description of the Waterfall model is often cited as a 1970 article by Winston W. Royce [11], although Royce did not use the term "waterfall" in that article.

It was, in any case, the first and most well known process model to be introduced, as well as the oldest System Development Life Cycle approach (SDLC), that was used for software development. When this approach was adopted for software development, there were no recognised alternatives for knowledge-based creative work in this specific field, which was quite immature.

The Waterfall method is defined as a linear project management approach in which stakeholder and customer requirements are gathered at the beginning of the project and then sequential project plan is created to accommodate those requirements. It is a breakdown of project activities into linear sequential series of stages called “phases” of the development life cycle, where each phase depends on the deliverables of the previous one and corresponds to a specialization of activities. Thus, the Waterfall method derives its name from the trend of scales by which development events cascade from one phase to another, constantly following the descent like a Waterfall [12]. Each phase must be completed before the next phase can begin and there must be no overlapping in the phases.

The development team, usually cross-functional, i.e. with different functional expertise, then follows the phases from idea to implementation, with each phase building into the next and not going back. The process assumes a transition from one step to the next, as each phase adds value to the product being developed. Because of the phase-by-phase structure of this method, it is often best used to develop complex products that require detailed specifications and efficient team communication [13]. The approach is typical for certain areas of engineering design, above all, for software development, where it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction.

However, the Waterfall model seems to be much better known and more important in the field of software even than in the mechanical industries. This is because each phase has its individual importance, so this approach leads to the generation of more productive and stable software. For this reason, this method will be studied with a focus on software development.

Finally, it is worth adding that Agile, in a. broad sense, is a different way of approaching the management of complex problems. While it was born as a methodology to be adopted in software development, it is also true that today it is applied to the most diverse sectors: from the design of machines and aircraft, to the re-engineering of processes, to the realisation of any type of product.

1.3.2 Waterfall life cycle phases

There are four phases that are essential to traditional development methods. The first phase is the create requirements phase, the second phase is the planning phase, the third phase is the development phase and the last phase is the testing and operations phase. These are the four main categories that other steps fall under [14]. Usually, also considering intermediate phases, the life cycle of the product, or software, includes the phases of (requirement) analysis, (system) design, implementation or construction, testing, deployment and maintenance [15], as can be seen from figure 1.1.

Of course, there can be many variations that describe the Waterfall methodology workflow. In any case, except for the name of the phase which may be expressed differently, they all describe the same set of activities.

In the following subsections the phases of the Waterfall model will be analysed and described one by one.

Requirement Analysis

The first phase, the requirements analysis phase, involves collecting all possible requirements of the process and documenting them in a requirements specification document. Requirements must be clear before starting the next phase and, in

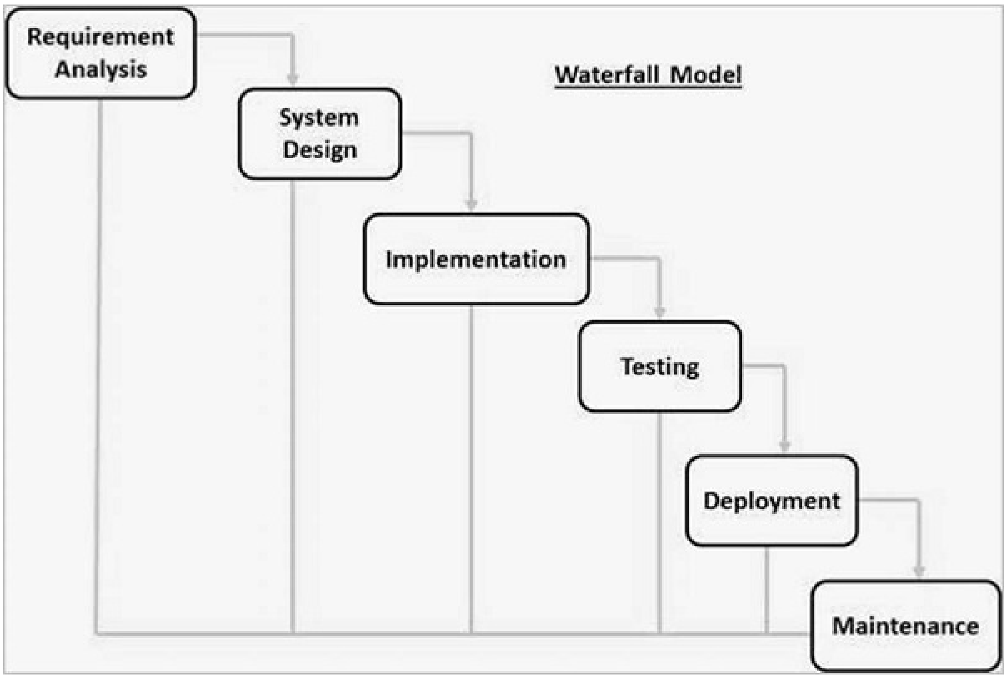


Figure 1.1: The Waterfall model life cycle [16]

many cases, changing requirements will not be taken into account. The key aspect of Waterfall approach is that all customer requirements are established at the beginning of the project, allowing every other step to be planned without further correspondence with the customer until the project is completed. Gathering and analysing requirements means knowing and understanding what the client needs to design, what to develop and the processes involved, what its functionalities will be, etc. This phase therefore provides input "material" to the product being made, i.e. it provides the scope for deciding the hardware or software requirements of the product to be designed and then it is studied, developed and released.

This phase is usually divided into two parts:

- **Requirement gathering**. First of all, all information and requirements for product development are collected from the customer and then processed; in this case processed means studied and treated for analysis. The main role of this part is to eradicate incompleteness and inconsistencies related to software product development by removing functionalities and limitations, if any.
- **Requirement specification**. The requirements analysed above are then documented in a specific document in a clear and unambiguous manner. It serves as a pathway between the customer and the development team and as an input for subsequent steps. Any future disputes are handled and resolved

only through this documentation.

In the case of software development, for instance, a software requirements specification document describes how a software system should be developed from the beginning. In simple terms, this document provides all those involved with a roadmap for that particular project. On the other hand, a product requirements document is necessary to establish a priori what will be built in order to allow the customer to know what the final product should do. In general, that kind of document obliges to put in writing the idea that will be used in order to cover all these details and, therefore, describes what the customer wants and what the developers will deliver. In this sense, having a clear set of requirements ensures that the development team will come as close as possible to the customers' needs [17].

System Design

The second phase is a planning phase. In this phase, the requirements specifications of the first phase are studied and the system design is prepared. This helps to specify the software and hardware requirements for the product design; it also helps in the overall architecture for the system design. The design process translates the requirements into concrete representation that can be evaluated for quality before the start of the generation. This allows potential problems to emerge. If this step is missed, it may lead to further problems during the development and testing phases, because these problems are not so easy to solve later in the development. At the same time a test plan is prepared: it describes the various tests that will be carried out on the system after the development is completed.

The design phase is best broken up into two subphases [12]:

- **Logical design**. The logical design subphase occurs when possible solutions are brainstormed and theorized.
- **Physical design**. The physical design subphase occurs when those theoretical ideas and schemas are made into concrete specifications.

Implementation

In this phase, work is done on the creation of the code or physical product until the project objectives are achieved. In the implementation phase, the work outlined in the documents of the previous phases is broken down into several modules or units, which are integrated in the next phase. Each unit is developed and tested to check its functionality, to ensure that it works and serves the purpose for which it was developed and to ensure that any remaining problems are corrected. This is called unit testing. So, unit testing basically confirms whether the units fulfil their specified purpose [16]. Then, at this stage, talking about software development, the system design is converted into source code with fully functional programme modules. It includes the development, testing and integration of the software. For product development, however, the process is the same, in which the actual implementation of the product translates into the manufacture of the product.

Testing

After the system is separated into units, developed, and tested for their functions, the integration and system-testing phase begins where the units are incorporated into the overall, total system. The units are tested to make sure they all coordinate with each other and the total system operates per specifications without any fault or failure. Testing is done, following the steps defined in the test plan, to ensure defined input produces actual results which agree with the required ones. A test report is then generated containing test results [16].

Moreover, testing can be done also as user testing, where the customer becomes a part of the testing and gives feedback.

Deployment

Once the solution has been successfully tested, the product is delivered to the end user through implementation in the customer's environment or release to the market. For a software development project, the implementation phase includes installation, migration and support of the complete system to the user's or customer's environment [16].

In addition, a verification phase may be added, in which the customer reviews the product to ensure that it meets the requirements established at the start of the project [12].

Maintenance

The next phase, characterised by maintenance operations, continues indefinitely. It involves making appropriate changes to the product or system, improving or modifying attributes related to system performance. ts main objective is to improve the performance of the system or product in general with maximum precision. These adjustments carried out during the maintenance phase are mainly related to changes requested by the customer or users after the installation and testing phase, which include problems with the created system or with the specific functionalities that were not discovered during its development life cycle and that have emerged after its use. Overall, this process is referred to as maintenance, as problems in the system emerge during its usage and need to be resolved by keeping the system running [16].

There are basically three types of maintenance which is explained below:

- **Corrective Maintenance**. During the design and development phase, some errors are not discovered, but are only taken into account when the customer first runs the software. It is called corrective maintenance, that means the correction of problems or errors that were not discovered in the development phase.
- **Perfective Maintenance**. This type of maintenance is carried out on customer request to increase and enhance the functionalities of the product or software.
- **Adaptive maintenance**. Relating only to software development, this is the maintenance required for switching the system environment. It is usually needed for porting the existing system to a new environment or a new operating system. This phase is important as it leads to better system performance.

1.3.3 Waterfall frameworks: Critical Path Method

With regard to non-software project development, one of the most popular instrument of Waterfall model is the Critical Path Method (CPM). In the CPM, all the activities needed to complete the project are classified, and then the expected duration of each activity and the dependencies between them are mapped. This helps to understand which activities can be completed simultaneously or the precedence between them.

Advantages

- **Better scheduling**. The emphasis on mapping the duration of activities and their interdependencies helps better scheduling of activities.
- **Prioritisation**. The success of the CPM methodology depends on the identification and mapping of critical and non-critical activities. Once these activities are mapped, resources can be prioritised in a better way.

Disadvantages

- **Need for previous experience**. If the team has no real-world experience with scheduling, it is difficult to calculate the time needed for each task.
- **No flexibility**. Like the Waterfall method, CPM is a front-heavy activity. It is necessary to plan everything at the beginning. If there are changes, the whole planning done until then becomes irrelevant.

CPM has a lot of application in complex but repetitive tasks such as industrial projects, for example, if tasks are required to be completed simultaneously, or one task finished before another can begin. It is less suitable for a dynamic area such as creative project management.

1.3.4 Advantages and disadvantages of Waterfall model

The purpose of the section is to present the main advantages and disadvantages of applying the Waterfall method.

Advantages

The advantages of Waterfall development consist in allowing departmentalization and control. A timetable with deadlines for each development phase can be established and a product can proceed through the stages of the development process one by one, proceeding in strict order. As it is subject to easily understandable and explainable steps, it overcomes many problems, so it is very easy to use. The easiness of use is also due to the rigidity of the workflow model, since each stage of the Waterfall method has specific review and delivery processes and timeframes. However, the Waterfall model works well for smaller projects, where the requirements are very well understood.

The advantages of the Waterfall model include the following [15]:

- The compartmentalization of work in each phase makes it easier to define a specific time period for the activities to be carried out;
- There is no overlapping of phases, as each phase must be completed before the next one can start;
- Since the project requires one phase to be completed before moving on to the next, any errors in the software or product in general can be detected early and corrected;
- Due to its linear design, the associated costs are lower than in other models, which in turn can help reduce the cost of the overall project;
- Unlike more recent software development methods, this method uses paperwork to document the development of the different phases. Therefore, it is easier for new workers entering the project to pick up where the previous worker left off;
- Testing is done after the development phase is completed, reducing the number of errors and maintaining the quality of the project;

- The Waterfall model is a well-known model among software developers, so tends to be easier to use. It is also easier to create different types of software using this method in a short time period.

Disadvantages

The disadvantage of Waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The disadvantages of the Waterfall model include also the following [15]:

- As it is very important to collect all possible requirements during the first phase in order to design the system correctly, there is a problem when not all requirements are received at the same time. Typically, customer requirements continue to be provided even after the completion of the first phase. This has the potential to affect the system development process and its success;
- External factors can influence the project to a large extent. For instance, when/if a client changes the requirements of the project, one has to start again from the beginning, as the Waterfall approach does not allow for changes in previous phases. Cost efficiency is therefore a negative aspect of this model;
- The issues with one phase are never totally solved before the next one begins, often resulting in a system that is poorly structured;
- The project is not divided in flexible phases and there is not overlapping of phases;
- As the customer keeps adding requirements, not all of them can be met, resulting in the development of a potentially unusable system. The cost of developing the system increases when the requirements have to be implemented in its new version;
- A large amount of time is potentially wasted on excessive documentation of the project;
- Software testing is not timely, i.e. it occurs quite late in the developmental process, so by the time errors are discovered, much time and money may have already been wasted on the project;
- High costs and effort are due to the number of documents to be approved at each stage, the difficulty of making changes, the difficulty of initiating and carrying out iterations and the problems that only arise at later stages;
- It is difficult to accommodate requests for change.

1.4 Projects methodological approach: Agile

The aim of the section is to present and study in depth another of the most important and well-known methodological approaches in project management: the Agile Model.

1.4.1 Overview

Agile software development is a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams [18]. In 1974, Edmonds formally defined Agile, also called the lightweight methods, in a research paper; However, it was created, in its entirety, during the 1990s when developers began to move away from highly traditional model with lack of ability to provide flexibility during software development lifecycle. In 2001, a group of 17 Agile software developers declared the Agile Manifesto [19], a set of guidelines that were suggested as an all encompassing framework for Agile software development models. It officially announced the foundation of modern Agility [20].

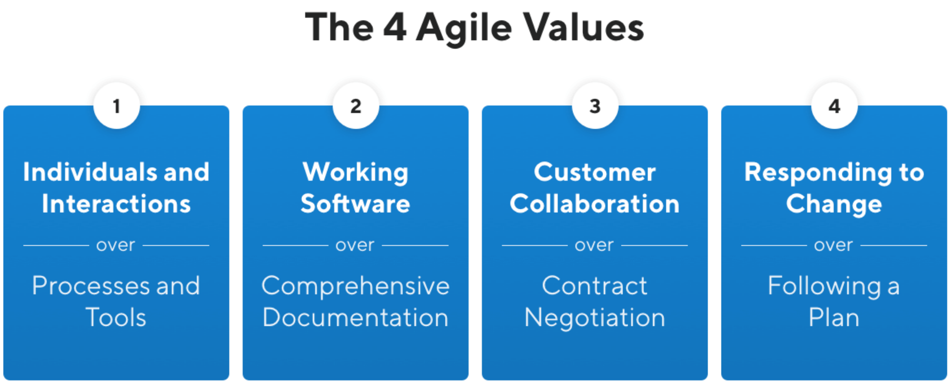
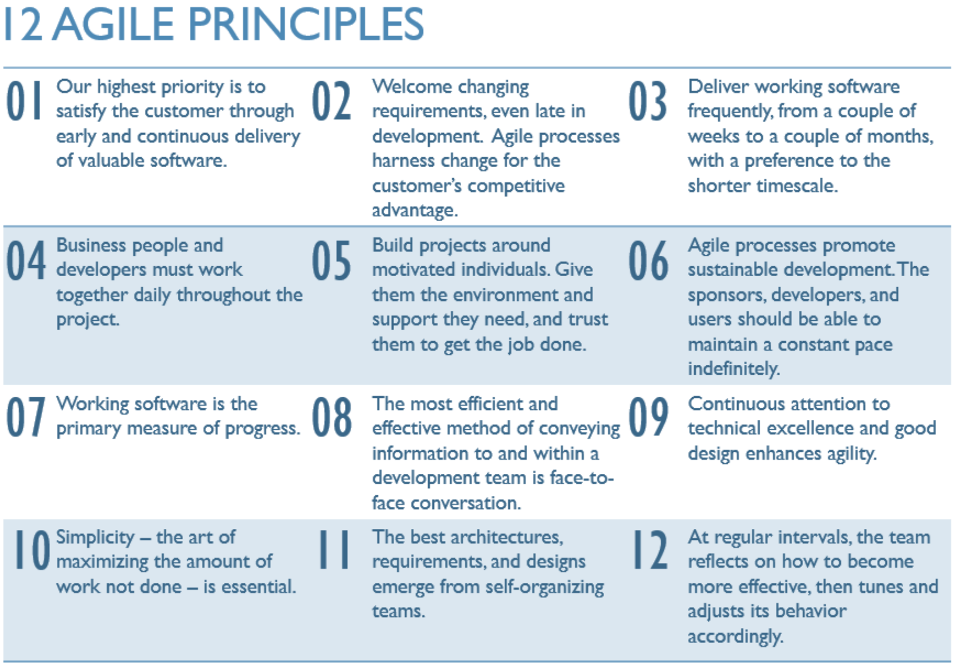


Figure 1.2: The 4 values of Agile Manifesto [21]

Agile approach practices are based on the discovery of requirements and the development of solutions through the collaborative effort of self-organised, cross-functional teams and their customers. Agile methods break down large projects into more manageable tasks that are completed in short iterations throughout the project lifecycle, with minimal adaptive planning and not directly involving long-term planning. Iterations, or sprints, are short timeboxes typically lasting one



12 AGILE PRINCIPLES

01 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

02 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

03 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

04 Business people and developers must work together daily throughout the project.

05 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

06 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

07 Working software is the primary measure of progress.

08 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

09 Continuous attention to technical excellence and good design enhances agility.

10 Simplicity – the art of maximizing the amount of work not done – is essential.

11 The best architectures, requirements, and designs emerge from self-organizing teams.

12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Figure 1.3: The 12 principles of Agile Manifesto [22]

to four weeks. Each iteration involves a team working through a complete software development cycle that includes planning, requirements analysis, design, coding, unit testing and acceptance testing when a working product is demonstrated to stakeholders. This minimises the overall risk and allows the project to adapt quickly to changes. Stakeholders produce documentation as required. An iteration may not add enough functionality to guarantee a market release, but the goal is to have a release available, with minimal bugs, at the end of each iteration. In addition, Agile allows teams to be better equipped to quickly change direction and to re-evaluate the work they are doing adapting to certain increments to ensure that, as the work and customer landscape changes, the focus also changes for the team.

One of the fundamental principles of Agile methodology is the idea of self-organising teams, which is a cultural challenge. Software engineering teams are empowered in unparalleled ways, reducing the role of the project manager to that of a facilitator.

With Agile, there is little documentation and planning, and once-a-day meetings work well when there is sufficient teamwork and motivation. The Agile methodology supports and encourages teamwork and collocation. Based on this assumption, therefore, is the idea that a working software is more important than the documentation of the whole process. The assumption is that the developers associated with the project will communicate with each other when necessary. Therefore, the basic principle is informal communication, rather than formal meetings, documentation and service level agreements.

Other very relevant aspects are shown in figures 1.2 and 1.3. In particular, the 4 values and 12 principles that characterise the Agile methodology are highlighted. They summarise the ideology and philosophy of the approach in consideration.

1.4.2 Agile life cycle iterations

The Agile methodology is defined by the completion of the objective. Agile requires that at each completed step, the product can be delivered. This means that by the end of the first step, the customer must have a usable product. However, many managers still say that all features must always be completed, the meaning of which in an agile respect is clearly not possible.

The Agile method suggests that within each sprint, or each step achieved, changes are well understood and perfectly created. It does not say that change requirements can be made within a sprint. On the contrary, it suggests that if there are serious changes within a sprint, it must be left and a new one started in its place [15].

Each iteration in an iterative development is a self-contained mini-project, with activities spanning requirements analysis, design, implementation and testing. It is well established that each sprint leads to an iterative release, which can only be an internal release that continuously integrates all the software developed by the team and is therefore a growing and evolving subset of the final system. The client adaptively specifies its requirements for the next release based on observation of the current release, rather than speculating at the beginning of the project [23].

According to Agile Alliance and PMI [7], the basic flow of Agile methodology is presented in figure 1.4 and consists of:

1. **Requirements**;

2. **Design**;

3. **Development**;

4. **Testing**;

5. **Deployment**;

6. **Review**.

As explained above, each of these stages is performed and repeated until the project is completed.

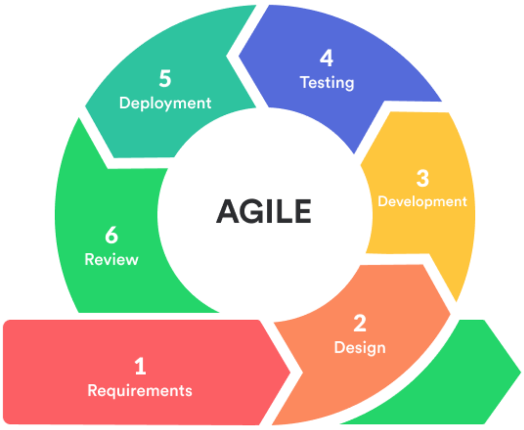


Figure 1.4: The Agile life cycle development [24]

Requirements

In this phase, requirements for iteration are defined based on the product backlog, sprint backlog and customer and stakeholder feedback. Initial documentation is created. It is usually recommended to lower these initial requirements, adding only those features that are definitely needed. This way, it will be easier to deal with the possible change of requirements during the next iterations, as the amount of structured elements that can be altered will be limited [25].

In subsequent iterations, the team revises the requirements to make them more relevant by adding any new requests.

Design

The next phase involves designing for future software or product development based on defined requirements. There are two ways of approaching design in product development: one is the architectural structure of the product and the other is its visual design [25].

- **Architecture Design**. During the first iteration, the team introduces the requirements created during the previous phase. Then they discuss how to deal with these requirements and propose the tools needed to achieve the best result.

  On further iterations, the developers discuss the implementation of specific functionalities and the internal structure under construction.

- **Visual Design**. During this phase, designers create a rough model of the interface. If the product is consumer-grade, the user interface and user experience are particularly important. The opposite is clearly true for a software product [25].

Further iterations are spent refining the initial design and/or reworking it to suit the new features.

Development

During this phase, the team works on delivering a working software/product by converting the design documentation. It is developed based on iteration requirements and feedback, and is modified and improved. This phase is divided into several stages, often called sprints [25].

In this phase, there are no particular changes between iterations, except for the inevitable inclusion of new features or updates required.

Testing

This phase is spent ensuring that the product is error-free and compatible with all previous versions. The team conducts a series of tests to ensure that there are no defects and that the business objectives of the solution have been met [25].

During the subsequent iterations of this phase, the tests become more and more specific and not only cover functionality, but also system integration, interoperability, user acceptance tests, etc.

Deployment

This phase includes the integration and delivery of the working iteration into production. The application is then deployed on the servers and/or delivered to customers [25].

Further iterations update the already installed product, introducing new functionality and fixing bugs.

Review

Once all previous development phases have been completed, the team reviews the progress made towards the completion of the requirements. Then, they introduce their ideas for solving the problems that arose during the previous phases, taking each proposal into account [25].

Thereafter, the phases of the Agile software development lifecycle start again, with a new iteration or moving on to the next phase.

1.4.3 Agile frameworks: Scrum

Scrum is a lightweight yet incredibly powerful set of values, principles and practices. The Scrum method describes an approach to Agile management with a focus on project teams, short 'sprints' and daily stand-up meetings. It works by breaking large products and services into smaller pieces that can be completed, and potentially released, by a cross-functional team in a short space of time [26]. The Scrum approach places the project team at the centre of the project. Often there is no project manager, but instead the team is assumed to be self-organising and self-managing. This makes it ideal for highly focused and skilled teams.

Advantages

- **Scrum "sprints"**. The scrum approach is strongly focused on "sprints" of about 30 days. This is where the project team breaks down a list of final objectives into smaller chunks and then works on them icon daily meetings. This makes it easier to manage large and complex projects.
- **Fast paced**. The “sprint" approach, with its 30-day limit and daily meetings, promotes rapid iteration and development.
- **Team-focused**. Since the project team is self-managing, Scrum teams have greater visibility over the project. This also means that project managers can set their own priorities based on knowledge of their capabilities. In addition to these, it has all the advantages of the Agile model, as rapid iterations and regular customer feedback.

Disadvantages

- **Scope creep**. As there is no fixed end date, nor a project manager for planning and budgeting, Scrum can easily lead to scope creep.
- **Higher risk**. As the project team is self-managed, there is a higher risk of failure unless the team is highly disciplined and motivated. If the team is not experienced enough, Scrum has a very high probability of failure.
- **Lack of flexibility**. The project focus on the team means that should a resource leave the team in the middle, it would have a huge impact on the net results. This approach is also not flexible enough for large teams.

The Scrum approach is best when used in the context of highly experienced, disciplined and motivated project teams that are able to set their own priorities and clearly understand the project requirements. It works for large projects, but fails if the project team itself is very large.

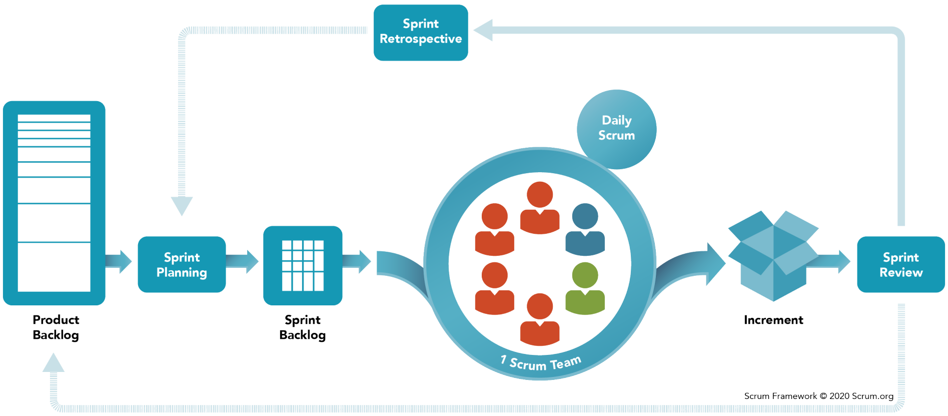


Figure 1.5: Scrum methodology [27]

1.4.4 Advantages and disadvantages of the Agile model

The purpose of the section is to present the main advantages and disadvantages of applying the Waterfall method.

The advantages of the Agile model include the following [28]:

- **Adaptability to the changing environment**. In the Agile development method, the software is developed in several iterations. After each iteration, the mini-project is handed over to the client for use and feedback. Any changes that update or modify the software are welcomed by the customer at any stage of development.

- **Ensuring customer satisfaction**. This methodology requires the active involvement of customers throughout development. The deliverables developed after each iteration are handed over to the user and improvement is made based on the resulting feedback. Thus, the final product is of high quality and guarantees customer satisfaction, as it is developed on the basis of the requirements taken from the customer.

- **Least documentation**. Documentation in Agile methodology is short and concise. Generally, no documentation is created on the internal design of the software. The main things that should be documented are the list of product characteristics, the duration for each iteration and the date. This saves development time and allows the project to be delivered in the shortest possible time.

- **Reducing risks of development**. The fact that work-in-progress software is released to customers after each short development cycle and feedback is collected allows developers to be alerted to impending problems that may occur in later stages of development. It also helps to discover errors quickly and correct them immediately.

The disadvantages of the Agile model include the following [28]:

- **Development based on customer interaction**. Customer interaction is the key factor of developing successful software. The Agile methodology is based on customer involvement as the entire project is developed according to the customers' requirements. Therefore, if the customer representative is not clear about the characteristics of the product, the development process will go astray.
- **Lack of documentation**. Although minimal documentation is one of the advantages of the Agile methodology as it saves development time, it is, on the other hand, a great disadvantage for developers. The internal design is changed several times according to the user's needs after each iteration, so it is not possible to keep detailed documentation of the design and implementation due to the project deadline. Therefore, due to the reduced availability of information, it is very difficult for new developers joining the development team at a later stage to understand the actual method followed to develop the software.
- **Constantly changing needs resulting in wasted time and resources**. If customers are not satisfied with the partial software developed by a certain iteration and their needs change, then that increased part is useless. The result is a total waste of time, effort and resources needed to develop it.
- **Higher utility for management than for developers**. The Agile methodology helps management to take decisions about software development, set goals for developers and set deadlines. But it is very difficult, for basic developers, to cope with a constantly changing environment and sudden architectural and design changes based on just-in-time requirements.

1.5 A comparison between Waterfall and Agile model

The traditional Waterfall life cycle is described as a linear, phased approach to software development, while the Agile approach seems to be diametrically opposed to this traditional development lifecycle [29]. Therefore, the question arises: *Agile or Waterfall?* This is the topic that the following section attempts to investigate.

1.5.1 Overview

As mentioned in the sections 1.3.4 and 1.4.4, there are favourable and unfavourable aspects to both Waterfall and Agile methodologies. Both can fulfil their task, but each methodology has its own strengths and weaknesses. A project that was not successfully completed using one of the two methodologies may have been better suited to the other.

The Waterfall methodology is known to have clear requirements, being easy to implement, use and manage. But it is also characterized by high documentation and inability to modify or update the project after the requirements have been defined. Waterfall also has a high risk and uncertainty. It is not suitable for complex or object-oriented projects; it is better for short-term projects.

The Agile methodology, on the other hand, is known for its adaptability and the need for the team to communicate face-to-face on a regular basis. But Agile is difficult for new developers to complete; they must have good technical skills.

1.5.2 Differences between the models

The next paragraphs show some aspects in which the two types of methodology differ. This helps to explain which types of projects are more suitable for one methodology or the other. In particular, aspects of both models can be classified as human or technical [14].

Human aspects

Human aspects are decisions and ideas controlled by human beings working on a project [14].

- **Fundamental hypothesis**. In traditional development models, such as the Waterfall model, systems are fully specifiable, predictable and developed through extensive and detailed planning. In Agile development models, on the other hand, high-quality adaptive software is developed by small teams using the principle of continuous design improvement and testing based on feedback and rapid change.

- **Management style**. The management style of the Waterfall model assumes command and control, while the Agile style favours leadership and collaboration.

- **Knowledge management**. Explicit for Waterfall and tacit for Agile.

- **Communication**. Waterfall model requires explicit and documented knowledge, whereas the Agile model advocates constant face-to-face communication.

- **Client involvement**. Low for Waterfall and high for Agile.

- **Organizational structure**. Waterfall models prefer a mechanical, bureaucratic, large-scale and highly formalised organisation. On the other hand, Agile models prefer an organic, small and flexible organisation that is participatory and encourages social co-operation.

- **Quality control**. The Waterfall model is characterised not only by arduous planning and strict control, but also by difficult and late testing. The Agile model, on the other hand, is more oriented towards continuous control, planning, testing and solutions.

- **Additional abilities required from developers**. No specific skills are required in projects based on the Waterfall model. In Agile projects, on the other hand, developers have to have strong interpersonal skills and basic business knowledge.

- **Developers**. Developers in a Waterfall model must be project-oriented, appropriately qualified and with access to external knowledge. In contrast, developers in an Agile project must be agile, competent, conversational and collaborative. They must also be talented, skilled and communicative. Agile approaches emphasise the importance of cross-functional teams of developers, testers, subject matter experts and "architects".

- **Clients**. Clients, in Waterfall models, must have access to project knowledge and be cooperative, representative and competent. In Agile models, similarly, they must be dedicated, competent, cooperative, representative but with decision-making power.

- **Appropriate scale of the project**. Large-scale projects are more suited to Waterfall models, while medium/low-scale projects are more suited to Agile models.

- **Requirements**. The formal, heavy Waterfall model has trouble keeping up with rapidly changing requirements. However, if the architecture can anticipate and thus accommodate changes in requirements, the Waterfall model can also keep developed applications within budget and on schedule. On the other hand, a specific formal requirements engineering, regardless of the solution, is not directly used by pure Agile practitioners. Therefore, requirements are very stable and known in advance for Waterfall projects, they are instead emergent and rapidly changing in Agile projects.

- **Testing**. In the development cycle of the Waterfall model, the testing phase is completed after implementation; in the Agile cycle, it must be carried out at each iteration. Conventional maintenance methods include design and architecture principles, dynamic testing, static analysis and internal and third-party reviews, evaluation and vulnerability testing. These methods are much more suitable for Waterfall development as they are well documented and focused on architecture. Agile methodologies, on the other hand, facilitate internal design and code review and motivate developers to adopt coding standards, but do not focus on architecture or documentation.

- **Primary objectives**. The Waterfall model states a number of important objectives, characterised by predictability, repeatability and optimisation. The Agile model focuses mainly on rapid value and response to change.

Technical aspects

The technical aspects are those that accompany the implementation plan [14].

- **Development model**. The Waterfall development model is based on a pre-established life cycle, the Agile development model is based more on an evolutionary-delivery model.

- **Development direction**. It is fixed for Waterfall model and easily changeable for Agile.

- **User requirements**. User requirements are detailed and defined at the beginning of the project on the basis of a traditional cascade approach, while they are interactive in the Agile model.

- **Architecture**. Waterfall architecture is designed for current and predictable needs; Agile architecture is designed for current and evolving needs.

- **Remodeling**. It is expensive for Waterfall plan and the opposite for Agile.

- **Size**. The Waterfall model is better suited to large projects as opposed to Agile projects. A bureaucratic organisation, driven by a pre-established plan and taking a long time just to get authorisation and start a project, will not be very efficient for small projects.

Project characteristics

Looking more specifically at the characteristics of the project, it is important to analyse which ones make the difference when it comes to choosing the right development methodology [23].

- **Customer relations**. Agile methods work best when clients work in a dedicated way with the development team and when their tacit knowledge is sufficient for the entire duration of the project. This method overcomes tacit knowledge deficiencies, which the Waterfall model avoids through documentation and the use of architecture and design review boards by independent experts to compensate for customer deficiencies.

- **Planning and control**. Formal project management plays an important role in the success of a software project. Project management requires careful planning, estimation, coordination, monitoring and control. Aspects formally addressed in the Waterfall Model. Agility, on the other hand, places more value on the planning process than on the resulting documentation.

- **Development**. Agile model values working software over full documentation, and emphasises simplicity, maximising the amount of work avoided. This is while the Waterfall model relies heavily on the software architecture as part of the development sequence.

- **Culture** Agile methods succeed better in a culture that 'thrives on chaos' than in one that 'thrives on order', whereas the opposite is true of the Waterfall Model.

The following table (1.2) summarizes the points of contact between the Waterfall and Agile models.

Table 1.2: A comparison table [14], [23]

Aspect	Waterfall model	Agile model
Fundamental hypotesis	Systems fully specifiable, predictable and developed through extensive and detailed planning	High-quality systems developed by small teams using the principle of continuous improvement in design and testing based on feedback and rapid changes
Management style	Command and control	Leadership and collaboration
Knowledge management	Explicit	Tacit
Communication	Formal	Informal
Development model	Lifecycle model	Evolutionary-delivery model
Organizational structure	Mechanic (bureaucratic, high formalization), targeting large organization	Organic (flexible and participative, encourages social cooperation), targeting small and medium organizations
Culture	Thrives on order	Thrives on chaos
Quality control	Difficult planning and strict control, difficult and late testing	Permanent control or requirements, design and solutions, permanent testing
User requirements	Detailed and defined before implementation	Interactive input
Cost of restart	High	Low
Development direction	Fixed	Easily changeable

Continued on the next page

Table 1.2 – *Continued from previous page*

Aspect	**Waterfall model**	**Agile model**
Testing	After coding is completed	Every iteration
Client involvement	Low	High
Additional skills required of developers	Nothing in particular	Interpersonal abilities and basic knowledge of the business
Appropriate scale of the project	Large scale	Low and medium scale
Developers	Oriented on plan, with adequate abilities, access to external knowledge	Agile, with advanced knowledge, co-located and cooperative
Clients	With access to knowledge, cooperative, representative and empowered	Dedicated, knowledgeable, cooperative, representative and empowered
Customer relationship	Independent experts to compensate	Dedicated way with the development teams
Requirements	Very stable, known in advance	Emergent, with rapid changes
Architecture	Design for current and predictable requirements	Design for current requirements
Planning and control	careful planning, estimation, coordination, monitoring and control	Planning over documentation
Remodeling	Expensive	Not expensive
Size	Large teams and projects	Small teams and projects
Primary objectives	High safety	Quick value

1.5.3 Methodology transition

Originally developed in the context of manufacturing, Waterfall methods work in a series of dependent steps that begin with gathering extensive requirements in advance and executing a project based on those requirements. While a Waterfall approach may work well for simple, straightforward projects, it does not work well for complex projects, including digital projects, in the sense of transforming or managing digital strategies and operations, which typically contain many unknowns. Indeed, complex projects are full of uncertainties. With Waterfall methods, the test phase is typically not used at every single stage of a project. This does not leave much room for collaboration or incremental feedback. If changes are needed late in a project, they can be both costly and time-consuming to implement. The whole system works better if all parties can work together to test hypotheses and validate, or refute, proposed solutions. The only problem is that, in order to work in this way, all parties involved in the project should feel comfortable with some uncertainty.

Unlike conventional methodologies, inspired by the Waterfall model, Agile approaches do not claim to know all the requirements that need to be developed in advance, but rather focus on small sets of functionalities that give rise to versions released to the client frequently [30]. Therefore, Agile methods can help teams to better manage complex projects by embracing uncertainty and incorporating continuous learning into the process. Agile methods are commonly used in project management and software development, but can also be applied to organisational processes.

There are many challenges that companies face when trying to move from Waterfall to Agile. For instance, when a company is working on several projects at once. In fact, the transition between Agile and Waterfall seems more daunting when companies have to manage the change of several projects at once. Another common example may be the problem of assigning tasks to team members and managing their time during such a transition. Communication management seems to be quite always a problem for team members. Completing a project according to the Waterfall methodology requires a lot of written communication and does not require as much verbal communication and collaboration; therefore this is a big change during the transition to Agile. Another very common problem is that people do not know what an Agile model looks like and how it should be interpreted. This is a problem because if people do not fully understand what they are moving towards and how they should work, they will not be able to achieve the goal. Finally, more generally, people are resistant to change. People's reluctance to change and learn new methods to complete projects is the biggest obstacle that companies face when moving from Waterfall to Agile [31].

Organization and management

The transition to an Agile process affects the whole structure of the company, e.g. development team, departments and management. Resistance can be found in all these elements, because the structure is different and people are typically used to spending a lot of time thinking about the project plan, attending meetings and adopting strict work processes. Therefore, they are typically oriented towards structured and planned processes. In addition, most developers in a Waterfall environment have a lack of business knowledge.

Agile processes encourage greater participation of all employees in the organisation. Knowledge sharing is one of the key elements for the success of this model. However, resistance is often also encountered due to a lack of communication and the ability to work in teams; Agile methodologies must be able to break through these barriers to be successful. Culture is also mentioned as a challenge in the process of adopting Agile methodologies. Indeed, cultural barriers may appear due to long periods of working with traditional development. However, culture is not only a barrier, but also an advantage: development teams can enjoy a big difference in their working culture, which results in more collective knowledge. Bureaucracy can also emerge as a problem, particularly in large projects that receive contributions from several companies. The rules and procedures these organisations commonly adopt can be very different, often, it is necessary to define a rigid schedule of activities in the kick-off meeting of a project, which goes against the working principles of Agile methodologies [30].

People

Lack of knowledge in Agile methodologies is usually mentioned as a challenge to overcome. When new software engineers move to new companies, they perceive some initial difficulties in understanding and using Agile practices. This situation is mitigated by training programmes; but some problems also remain, due to the lack of available courses and the consumption of time. Experience and commitment are two different concepts. On the one hand, and due to the fact that Agile methodologies are almost new, it is difficult for companies to adopt a set of practices based on the experience of previous projects. On the other hand, in Agile methodologies employees are asked to be more adaptable in order to be able to perform different roles at the same time. The difficulties of working in large and geographically distributed teams are also reported as a challenge. In this dimension, it is stated that the remote location of team members and stakeholders means that Agile practices have to be properly adapted.

To understand when the teams is ready for a Waterfall to Agile transition, there are a few things it's good to look at [32]. As far as team capacity is concerned, it is important to leave enough flexibility for unscheduled activities, which always occur. In addition, it is better to plan each person's workload individually rather than planning workloads by team in order to best meet the ever-changing project requirements while maintaining the sanity of the team. A key part of planning a successful sprint is estimating the difficulty of the task, this will help build team consensus in executing the required deliverables on time. It is also important to consider adjusting the length of the sprint, if necessary, until something is found that works for all project stakeholders.

Finally, the involvement of stakeholders throughout the process and the correct identification of user needs are a permanent challenge for the success of projects under Agile methodologies. Priorities for the implementation of user requirements are set for all iterations, providing a guarantee of desirable results and customer satisfaction [30].

Process

Team practices based on work sharing are an intrinsic feature of Agile methodologies. The main challenge is how to make it work considering the different personalities and knowledge of the teams. In the Agile paradigm, requirements identification is a process of discovery along the way to ensure that the final requirements meet the customer's needs. Generally, missing requirements can only be a problem in a scenario where stakeholder involvement is low. On the other hand, conflicting requirements are a common and somewhat desirable challenge to meet customer expectations. Besides that, not only functional requirements should be identified through user interaction, but also non-functional features of a project, such as portability, expandability, security, usability, among others. Dependencies between teams emerge in complex projects when several of them work together on the same release. Problems related to features and shared work may appear, especially when heterogeneous teams are involved.

End-to-end quality is an essential attribute of a good software development methodology. Agile development promises to provide quality assurance and is popular for delivering quality projects in small deliverables. Several Agile quality strategies can be applied, in particular the use of refactoring, reviews and inspections, and the adoption of standards and guidelines .

Risk management is not a widely adopted practice in Agile development. Consequently, there is still considerable scope for development and research. It is important to ensure their identification, prioritisation and management/control. Finally, scalability is also a topic that has been written about and discussed by practitioners. Some issues are still present, namely: how to guarantee the same speed as in small projects, how to coordinate dependencies in large and complex projects, and how to guarantee the same quality in large initiatives. In this context, the adoption of risk management techniques becomes crucial [30].

Project complexity

The technical complexity of the project is considered a significant issue with a strong impact on the success of an Agile project. The development of large-scale applications is particularly challenging due to their complexity and the high degree of interdependence between workflows. Agile methodologies and management approaches have to overcome risks and barriers at every stage of development of a complex technical project. Integration of systems, particularly in more complex projects, also presents a number of challenges. Many times in these projects there is a need to integrate modules from different suppliers. The integration of modules with different technologies is an additional challenge. A relevant issue in an Agile environment, where the identification of requirements is a constant process throughout the project development, is the creation of certain barriers and limits to achieve the desired results. Defining an acceptance criterion after several iterations is one of the best practices to apply [30].

Clients

Moving from the Waterfall model to Agile is a long process, but in the end it improves relationships with project stakeholders, including customers, as work becomes more collaborative. The Agile model also improves the ability to monitor the efficiency of the business and is great for morale since collaborative teams show their work and receive feedback at the end of each sprint. Customers also tend to love Agile processes because they can observe results in real time and offer feedback at the end of each sprint [30].

1.6 Other methodologies

For the sake of completeness, this section is intended to list and summarise other important project management methodologies, mentioning their strengths and weaknesses [33]. Some of these methodologies are part of the Waterfall and Agile methodologies, being variants or evolutions of them. However, they are considered separate from the core processes of the previously mentioned approaches and will therefore be presented in a separate section.

1.6.1 Hybrid

The hybrid approach, as the name suggests, is a combination of the Waterfall and Agile methodologies. It takes the best parts of both the Waterfall and Agile models and combines them into a flexible but structured approach that can be used in different projects. The hybrid methodology focuses on gathering and analysing requirements initially, a typical Waterfall approach, and from there takes the flexibility of the Agile approach with an emphasis on rapid iterations.

Advantages

- **Increased flexibility**. After the planning phase, the hybrid method offers considerably more flexibility than the Waterfall method. As long as the requirements do not change substantially, it is possible to make the required changes.

- **More structured**. Through the initial planning phase of the Waterfall model, the hybrid method overcomes the lack of structure and planning, a typical problem of the Agile model.

Disadvantages

- **Request for compromise**. As two opposing approaches are essentially being reconciled, both sides will have to compromise, especially on requirements and flexibility.

- **Mix of the models**. This approach, as said, combines the flexibility of the Agile model and the security of the Waterfall model. Any iterations will have to respect budget and scheduling constraints established in advance.

The hybrid approach is best suited for projects that have medium requirements compared to Agile and Waterfall, i.e. require structure and flexibility. These are mostly medium-sized projects with moderately high complexity but fixed budgets.

1.6.2 PRiSM

Projects Incorporating Sustainable Methods (PRiSM) is a principled and sustainable project management methodology developed by GPM Global. The fundamental difference from traditional approaches is that, on the one hand, it incorporates a value maximisation model that focuses on the total life cycle of assets and, on the other hand, it focuses on accounting for and minimising the negative environmental impacts of the project [34].

Advantages

The PRiSM approach is highly relevant for modern projects where environmental costs and sustainability are key success criteria. For large projects where reducing energy consumption, managing waste and minimising environmental impact are key, PRiSM offers a strong project management ideology.

Disadvantages

PRiSM is not suitable for projects where environmental impact is not a concern, such as software or creative projects. The success of the PRiSM approach also requires that every part of the project team agrees with the principle of sustainability, which is not an easy requirement in most organisations.

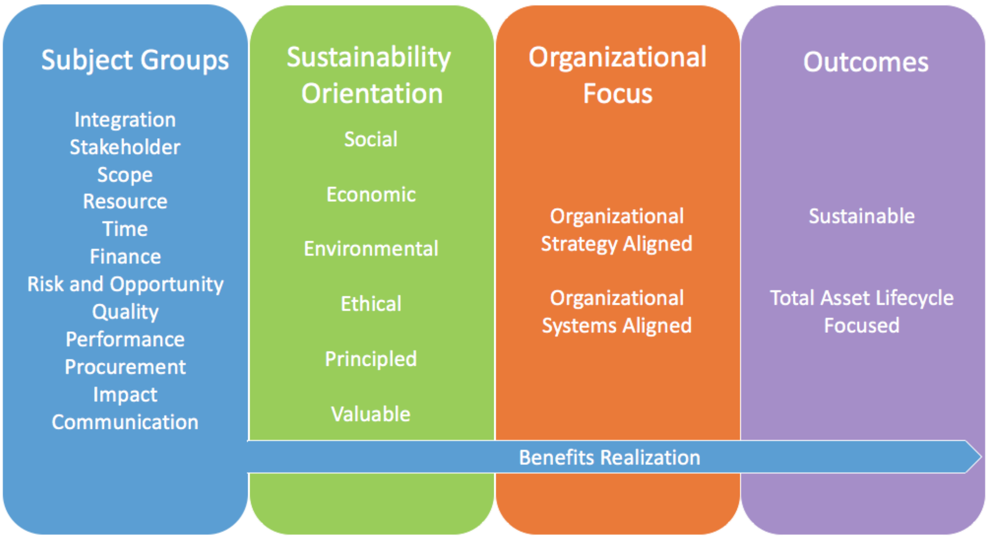


Figure 1.6: PRiSM methodology [35]

PRiSM is especially suitable for large, complex real estate and industrial projects where sustainability is a key concern.

Figure 1.6 shows the main subject groups of PRiSM methodology and the related beneficial aspects realised.

1.6.3 PRINCE2

Projects IN Controlled Environments (PRINCE2) is a globally recognised method launched by the UK government. It has a number of project management certifications based on this method and a private organisation manages the certifications on behalf of the UK government. It is a process-based approach to project management that provides an easily customisable and scalable method for managing all types of projects. Each process is defined with its key inputs and outputs together with specific objectives to be achieved and activities to be performed [36].

Table 1.3 sets out the main composition of the method, through its principles, themes and processes.

Table 1.3: The PRINCE2 framework [36]

PRINCE 2 FRAMEWORK		
PRINCIPLES	**THEMES**	**PROCESSES**
Continued business justification	Business case	Starting up a project
Learn from experience	Organization	Initiating a project
Defined roles and responsibilities	Quality	Directing a project
Manage by stages	Plans	Controlling a stage
Manage by exception	Risk	Managing a product delivery
Focus on products	Change	Managing a stage boundary
Tailor to suit the project environment	Progress	Closing a project

Advantages

Managing a PRINCE2 project requires extensive documentation. Furthermore, one of the guiding principles of PRINCE2 is to "learn from experience". This focus on documentation and past experience can help reduce risks.

Disadvantages

The disadvantage of PRINCE2's extensive documentation is that changes can be difficult to accept. If requirements change, the documentation has to be redone and resources reallocated, which can hinder the pace of the project.

This methodology is best suited to large, complex projects with fixed requirements.

Chapter 2

Company Profile

This chapter aims to present the company with which the project, the focus of the next chapters, was conceived, developed and concluded. First of all, the consultancy sector is specified, whose description is significant to explain how relevant the discipline of project management is in the context of the companies that are part of it. Next, the history, both ancient and modern, of Accenture spa is provided, giving a picture of the environment in which this project was carried out.

2.1 The consulting industry

Generating hundreds of billions of dollars in revenue annually, the consulting industry is currently among the most profitable, prestigious and fast-growing sectors in professional services. While it now encompasses six different sectors of consulting services, the industry as a whole has its roots in management consulting. This new industry began to develop shortly after the rise of management as a business sector in its own right at the end of the 19th century, along with the industrial revolution. The first companies were started by university professors. In particular, the first management consulting company was founded by chemist and MIT professor Arthur D. Little and took his name in the late 1890s. This company, which originally specialised in technical research, later became a general management consultancy. However, what is regarded in the history books as the first pure management consultancy in the modern sense was McKinsey & Company. It was founded in Chicago in 1926 by James O. McKinsey, a professor of accounting at the University of Chicago. It was the firm's belief that management consultancies should apply the same high professional standards as lawyers and doctors.

In the 1960s, a number of new management consulting firms were formed, among the best known being Roland Berger and the Boston Consulting Group (BCG). These firms helped bring a rigorous analytical approach to the study of management and strategy. During the 1960s and 1970s, the BGC, founded in 1963, Bain & Co, founded ten years later, and McKinsey, formed the Big Three of strategy consulting and, together with other firms such as Roland Berger and Booz Allen Hamilton, pioneered many of the analytical tools and approaches that would define the new field of strategic management. The publication of these concepts laid the foundations for many consulting firms to follow.

In the late 1990s, the consulting industry flourished, driven by a wide range of factors, such as a strong global economy, increased computing power, emerging market penetration, privatisation, globalisation and the new practice of IT consulting. Many established companies were increasing revenues at very high rates and new companies were springing up everywhere. In addition, Information Technology was in its infancy compared to today's era of digital disruption, but it had evolved rapidly since Steve Jobs founded Apple in 1976, and a new IT consulting industry was born.

After this boom period, this accelerated and diversified growth was briefly interrupted at the beginning of the 21st century as a result of the bursting of the dot-com bubble in 2001 and the Great Recession that shook the global economy in 2008, when many corporate clients had to start reducing their consulting budgets for the first time in decades due to the uncertainty generated by the economy in those years. As a result, many young/small firms had to downsize or withdraw from the market altogether, and larger firms drastically reduced their recruitment efforts. While the Big Three have explored the world beyond business, winning lucrative government contracts, the consulting arms of the Big Four professional services firms (Deloitte, KPMG, PwC and EY) have become titans of equal stature in the field.

Since 2004, the consulting industry has recovered substantially. All firms are aggressively recruiting again. Currently, most consulting firms are operating at full capacity and the outlook for the industry as a whole is very positive. Critical to success has been the role of advisers in helping clients understand, embrace and exploit digitisation, disruptive technologies and a business landscape evolving at breathtaking speed. The same phenomena are now beginning to reshape the consulting industry itself, which until recently has proved immune to disruption thanks in large part to the measured agility of the best consulting firms and their freedom from fixed capital and resource dependency.

The current market trend is a clear segmentation of management consultancies by function. Six major consulting sectors (Management, Strategy, Operations, Technology, Human Resources and Financial Consulting) frequently overlap with each other and oversee the provision of consulting services to clients in hundreds of different industries and functional areas. Management consulting also continues to expand increasingly into non-business fields, particularly working with governments, quasi-governmental agencies and non-profit organisations [37], [38].

2.2 Accenture

As mentioned at the beginning of the chapter, this section aims to present the Accenture company through the narration of its history, from its foundation to what it represents in the world today.

2.2.1 Overview

Accenture S.p.A., hereinafter only Accenture, is a leading global professional services firm providing a broad range of services and solutions in strategy, consulting, digital, technology and operations. Accenture combines unique experience and specialized expertise in more than forty industries and across all business sectors, supported by the world's largest network of delivery centers. With extensive research activities in the field of technology, it works at the intersection of business and technology to help clients improve their performance and create sustainable value for their stakeholders. Accenture's expertise, know-how and strategic partnerships with leading software providers make it an undisputed leader in providing IT services, innovation and digital transformation to meet the growing demands of the market and its clients. Therefore, with approximately 482,000 professionals serving clients in more than 120 countries, it drives innovation to improve the way the world lives and works.

Figure 2.1 shows Accenture's worldwide coverage.

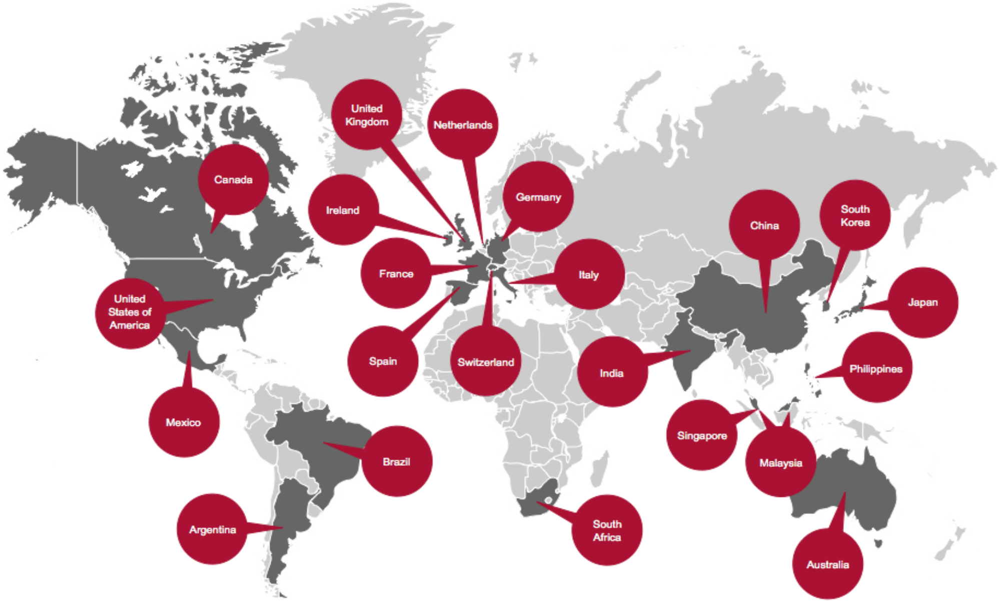


Figure 2.1: Accenture in the world [39]

2.2.2 Company history

Originally called Andersen Consulting, Accenture was formally established in 1989 when a group of consulting partners from Arthur Andersen firms around the world formed a new organization focused on consulting and technology services related to managing large-scale systems integration and business process improvement. That same year, Accenture formalized its business integration: a framework for aligning a client's people, processes and technology in support of its global strategy to enable all parts of the client's organization to work toward improving business performance. During its first 10 years, it evolved from a systems integrator to a global management consulting and technology services company, providing the full range of related consulting, outsourcing and technology services. Accenture then launched one of the largest and most successful re-branding campaigns in company history. Once the new logo was revealed, the challenging task of rebranding the organisation really began. At midnight on 31 December 2000, Andersen Consulting officially adopted the Accenture name following an arbitrary decision in August 2000 that severed the contractual ties between Accenture and Andersen Worldwide Society Cooperative (AWSC). As a result, on January 1, the company's corporate website was changed from *www.ac.com* to *www.accenture.com.* The new name reinforced Accenture's new positioning and reflected the organisation's further growth and expanded capabilities.

Since its birth in 1989, Accenture had operated as a group of independent, locally owned partnerships or other entities in more than 40 countries. Accenture is and always has been a global organisation and has never operated under a US holding company structure. In 2001, Accenture's partners realized that maintaining the organization's current partnership structure would limit the company's ability to continue to grow. Therefore, the partners, more than half of whom were from outside the United States, decided to transition to a corporate form, allowing Accenture to build and acquire the capital it needed to remain competitive and fuel its growth. In April 2001, Accenture's partners voted overwhelmingly to pursue an initial public offering, and Accenture became a public company on July 19, 2001, when it listed on the New York Stock Exchange under the symbol ACN [40].

2.2.3 Organization

The Accenture Group has eight operating companies that provide consulting, technology and business process outsourcing services: Accenture Outsourcing, Accenture HR Services, Accenture Managed Services, Accenture Technology Solutions, Accenture Back Office and Administration Services. There are also five innovation centres for the development of advanced solutions.

From an operational point of view, as shown in figure 2.2, Accenture is organized

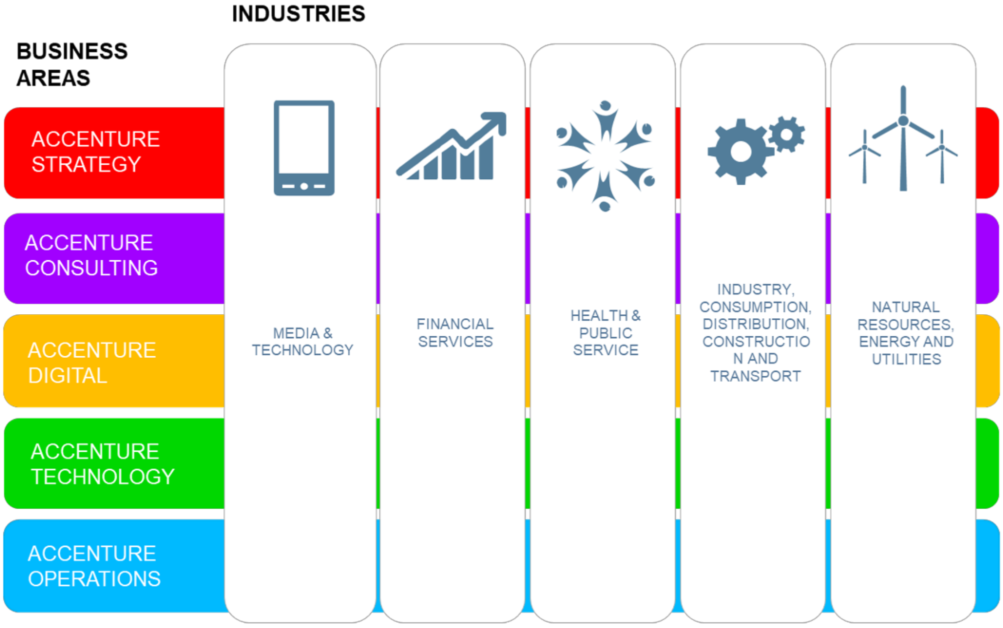


Figure 2.2: Accenture operating groups [39]

into five operating groups that cover all market sectors:

- **Communications, Media & Technology**. Communication, media and technology companies are the backbone of the information economy. That's why Accenture helps these companies move to experience-based, data-driven models and open platforms to improve their performance. In particular, Accenture serves communications & media, high tech, software & platforms and areospace & defence industries.

- **Financial Services**. Accenture helps financial institutions to become more innovative, agile and humane in order to manage disruption and create new value in a rapidly changing world. In particular, Accenture provides banking, capital markets, financial & risk management and insurance services.

- **Health & Public Service**. In the face of accelerating change, Accenture helps public sector organisations to embrace innovation and transform quickly in order to operate smarter, achieve more and improve services. Accenture provides services related to healthcare and central and local public administration;

- **Products**. As the market environment is constantly disrupted, Accenture helps companies digitise, transform and reach the always-connected consumer.

In particular, Accenture serves consumer goods, distribution and transport industries;

- **Resources**. Accenture provides consulting services related to energy, chemicals, natural resources and utilities.

And five cross operating groups divisions that represent the main service sectors:

- **Accenture Consulting**. It provides best-in-class experts with management and technology consulting expertise to transform the world's leading companies. Accenture Consulting has primary responsibility for directing and coordinating expertise across the organization to help clients transform their businesses. Accenture's consulting capabilities enable the design and implementation of transformation programs, both for one or more business units and for the entire organization. Accenture also helps clients implement digital transformation by enriching its services with expertise in digital, cloud, cybersecurity and artificial intelligence.

- **Accenture Digital**. Digital is gradually changing the way businesses and governments operate, from interacting with customers and suppliers to managing personnel. In this sense, Accenture provides analytics, interactive marketing and mobile services to help customers' businesses harness digital power to improve their performance, deliver the best experiences, and create new products and high-value business models.

- **Accenture Strategy**. By combining deep business knowledge with an understanding of how technology will impact the business model, Accenture helps clients to achieve specific business outcomes and increase shareholder value through the definition and execution of specific technology-based industry strategies.

- **Accenture Technology**. Accenture supports clients' businesses by providing best-in-class technology solutions, leveraging emerging technologies. Specifically, it designs, develops and implements complex technology solutions, drives software development and provides expertise in business planning, enterprise integration, data warehousing and delivery of "off-the-shelf" solutions. Accenture's technology platform includes the service areas of systems integration consulting, technology and IT consulting outsourcing services.

- **Accenture Operations**. Accenture provides business process, infrastructure, security and cloud services, including the Accenture Cloud Platform. Accenture works on infrastructure and business processes on behalf of clients, increasingly based on as-a-service, to help improve their productivity and performance.

Accenture Utilities

Since the topic and context of the project to which chapter 3 is devoted is the utility sector, it is interesting to focus on the latter.

In the Resources market, which includes the Utilities division, Accenture has worked with clients around the world for 30 years and has more than 22,000 people globally serving 93 of the 135 Resources companies in the Fortune Global 500. In the Utilities sector in particular, working with clients in the electricity, gas, water and renewable energy sectors and through significant investments, Accenture has been able to gain deep expertise in the area and use its experience to help companies achieve high levels of performance. Accenture serves the market through an industry-wide practice of professionals with specific industry process, technology and application expertise, as well as strategic partnerships with leading vendors. The level of specialization of Accenture's professionals and the strong collaboration with product owners, both globally and locally, allow Accenture to bring excellence in expertise and quality of service to clients in all phases of project activities. Over the years, Accenture has acquired specialized experience in all sectors of industry, in national and international contexts, which has enabled the development of Frameworks to shape the business of companies. In essence, Accenture helps clients cope with changing markets, obsolete infrastructure, regulatory changes, lack of expertise, volatile commodity prices and increasingly stringent environmental regulations. Finally, it can provide its clients with tools/assets/accelerators and best practices developed through its experience in the utilities sector.

2.2.4 Accenture Italia

Present in Italy since 1957, Accenture now employs around 16,000 people and has hired an average of 2,500 people per year over the last three years. The company has headquarters in Milan, Rome, Turin, Verona, Naples and Cagliari as well as several offices throughout the country. Italy is an integral part of Accenture's international network with:

- Six innovation centres for the development of advanced solutions in specialized sectors such as: broadband, telecommunications, food, fashion, automotive and enterprise 4.0: Accenture Automotive Industry Solution Center, in Turin; Accenture Customer Innovation Network, in Milan; Accenture Digital Acceleration Center and Accenture Digital Platform, in Rome; Industrial IoT Innovation Center in Modena and Accenture Cloud Innovation Center in Rome. In addition, Accenture recently acquired Sec Servizi in Padua, with the aim of creating an innovation hub for the financial sector.
- Two advanced technology centres in Naples and Cagliari. The Naples centre, opened in 2001, now employs more than 1,700 people and served as a model

for the Cagliari centre which, opened in 2015 together with Avanade, has a total of 500 employees. Both are part of a network of more than 50 centres around the world.

As regards utilities, in recent years Accenture has continued its strengthening strategy in Italy through the acquisition of New Energy Group (2016), a company specialising in Salesforce solutions, and i4C Analytics (2014), with the aim of consolidating its role in the digital and big data field. Accenture has a proven track record in the Utility sector in Italy, boasting partnerships with leading companies in the sector, providing the full spectrum of its services, from strategy to operations:

- It collaborates on Business and IT Transformation projects with leading companies in the market such as Enel and Eni;
- It works with the Italian branches of foreign groups such as E.ON, Edison, Engie, Axpo and Repower;
- He works with medium/small-sized companies such as Dolomiti, Sorgenia and Alperia, in the challenge of growth on the free market;
- It supports Italian multi-utilities such as HERA, A2A and, since 2016, also IREN in their efforts to tackle transformation/repositioning programmes.

The Accenture office where the project was developed is located in Milan and is also the company's registered office. During the information exchange and project presentation sessions, however, the work usually takes place directly at the client company's historical headquarters. Unfortunately, the entire project actually took place completely in smart working via the web, both for Accenture and the client company.

Chapter 3

Accenture Project Methodology

The purpose of this chapter is to present and explain in detail the semi-Agile project management methodology created and used by Accenture. In particular, the individual stages of the whole process are described, with reference to whether each stage belongs to the Waterfall or Agile method.

3.1 Overview

Accenture's project methodology envisages the modular activation of different working methods, depending on requested objectives, in order to accelerate the evolution process of the application map and guarantee high quality standards. In particular, Accenture applies a methodology that combines:

- An Agile, iterative and incremental approach, consisting of several "sprints" or "increments", for the design, construction and unit testing of the application;
- A more traditional Waterfall approach, for other project moments, such as the initial scoping phase, integration testing, user acceptance test (UAT), deployment and typically the integration areas that need an initial design phase and definite implementation times, which can only be managed within a waterfall model.

The Semi-Agile methodological approach is proposed based on sharing and moments of confrontation aimed at favouring the construction of software that is as close as possible to the requirements and at the same time aimed at favouring the monitoring of the progress of the work. The approach is implemented through conference room pilots in which, starting from high-level requirements, the design of the solution is carried out. Each section, as said, is also called a Sprint. Each Sprint is organised for the definition and verification of a set of requirements, grouped in consistent functional sub-processes or use cases. The elements characterising each Sprint are the following:

- Workshops functional to the design of the solution, in which the functionalities present natively, out of the box, in the application are deepened, where necessary also through the use of prototypes;
- Iterative validation process, which allows business users to constantly verify the alignment of the solution with expectations;
- Mixed teams, business and IT, at all stages of the project with the aim of having constructive discussions on the trade-off between custom versus standard solutions and making informed and shared decisions on the adoption of customizations or perimeter changes.

This methodology, specifically designed and deployed by Accenture, leverages elements of Agile approaches within the framework of Accenture's proven delivery methodology to combine flexibility and speed with the design rigour required in complex, enterprise-scale projects.

3.2 Project approach

The approach to the project, as previously stated, foresees a number of sessions ideally carried out in face to face meetings or workshops to best enable the iterative process of discussion based on the sharing of the application. The number of sessions required may vary according to the scale of the release and should be repeated for each of the capabilities in scope. Before starting with the initial scoping sessions, a number of elements must be available and in concrete terms: the organization process structure and the high level requirements and priorities. The sharing and consolidation of these elements will be carried out during the initial sessions in order to guarantee a shared starting point with the users, on which to build the subsequent sessions focused on the construction of the solution. In the following meetings, the implementation solution will then be shared according to an incremental process of consolidating the proposed product, based on the feedback received during the sessions and aimed at absorbing, where possible, modifications in the configurations that did not initially emerge.

The Semi-Agile approach may also cover a large number of processes in a relatively short period of time. For the adoption of the methodology to be successful, the effort of the whole team to observe the some "success factors" is essential. First of all, open communication, both vertically and horizontally, between functional management and technical staff, as well as between the development team and end users, is crucial. Then, an adherence to the approach taken and a commitment to using standard functionality wherever possible is necessary to ensure consistency and speed of execution. Minimising any customisation required will also be essential to enable the project team to complete developments in the shortest possible time. Commitment, active participation and adoption of the methodology, not only by the client's team, but also by all stakeholders, is also important. Finally, the parties involved must show strong decision-making capacity so that alternatives and choices can be decided online without the need for further consultation. Obviously, Accenture always provides an integrated team with different predominant skills and types of experience required to achieve maximum results.

The main benefits expected from this project approach are:

- Focusing solutions on business objectives and the value they bring rather than on requirements development;
- Valorization of standard solutions;
- Opportunities for innovation based on platform capabilities;
- Better quality in the design of the solution and respect for the expectations of users, depending on their involvement in the initial stages of the project;
- Transparency in the prioritisation process and in the effort to develop the solution;
- Empowerment of resources in decision-making and accountability throughout the duration of the project;
- Greater interaction and proactivity on the part of all the actors involved in the model.

3.3 Project methodology

As illustrated in figure 3.1, the methodological approach consists of 4 macro-phases: *Design*, *Build*, *Test* and *Run*. They are further divided into the following phases:

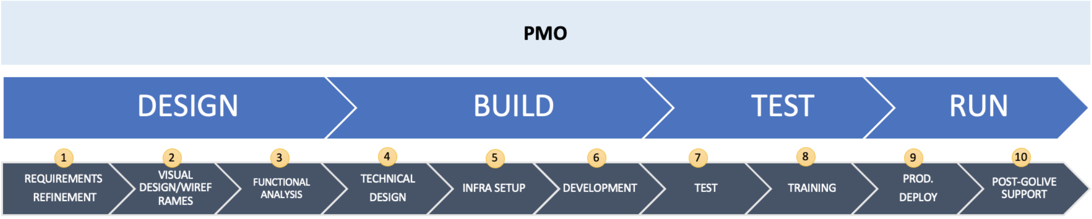


Figure 3.1: Accenture methodological approach

1. **Requirements Refinement**. The objective is to verify the project scope and functional requirements by carrying out a process analysis through a series of meetings and interviews with representatives of the client organisation;
2. **Visual Design - Wireframe**. The objective is to design the UX[1] in order to guarantee an application in line with the required usability expectations;
3. **Functional Analysis**. The aim is to consolidate the scope of the application and analyse the processes through a series of meetings and interviews with the client's corporate representatives. In this phase, the strategy and test documentation are defined;
4. **Technical Design**. The aim is to design the technical solution in compliance with the functional and technical requirements analysed;
5. **Infra Setup**. During this phase, the setup of all software environments necessary for managing the life cycle of the software solution is performed;
6. **Development**. The objective of this phase is to implement the technical solution through the various modules[2] that characterise the system;
7. **Test**. It allows the execution of the previously defined test strategy through the sub-phases of:
   - System Test, aimed at ensuring the correct communication of all the components developed and integrated with each other;

[1] User Experience Design

[2] Web App, API, DB

- System Application Test (SAT) and System Integration Test (SIT), aimed at guaranteeing the correct functioning of end-to-end functional scenarios. In SIT all possible system integrators involved in the agreed scenarios are involved;
- User Acceptance Test, aimed at validating the developed functionalities by assessing their business aspects;

8. **Training**. This phase is essential for training users toward the new application solution;
9. **Prod Deploy**. This phase consists of preparing the solution and releasing it to the customer in the production environment;
10. **Post Go Live Support**. The aim is to perform bug fixing on any anomalies found after go-live in production environment.

The client is supported in the governance and overall coordination of the project activities with the different actors involved in the initiative and in full alignment and synergy with existing initiatives. In particular, management activities, general coordination and the preparation and management of steering committees are foreseen.

As mentioned at the beginning of this chapter, the structure of the process that characterises the chosen methodological approach can be traced back to the Waterfall model of project management, enriched by some moments that follow the Agile philosophy, especially as regards requirements gathering, project design and unit testing.

The following paragraphs will discuss in detail all the phases outlined above that distinguish an Accenture project.

3.3.1 Design

The purpose of this macro-phase is to collect all the customer's requirements and design the architecture of the system to be developed.

Requirements Refinement

The Requirements Refinement (fig. 3.2) activities relate to passing all the necessary information to Accenture in order to be able to carry out the project in accordance with the initial objectives. In the initial phase of the project, it is necessary to plan a series of interviews and meetings with both the IT contacts of the application and any business contacts.

This phase corresponds to the "Requirements Analysis" phase typical of the Waterfall model, in which all the information needed for project development is

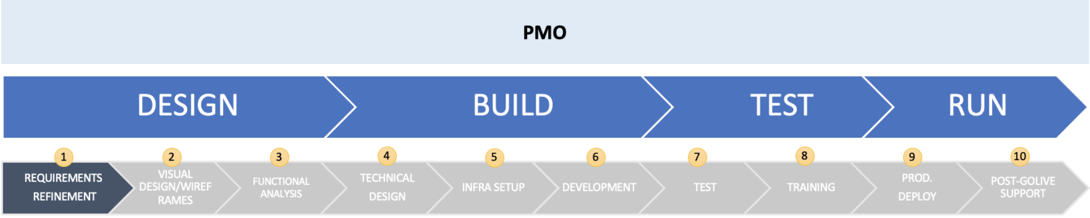


Figure 3.2: Requirements refinement phase

collected from the customer and analysed. Differently from the aforementioned model, these requirements are not fixed and established, but may evolve and change along the project according to an Agile approach.

For this reason, the described phase is characterised by a Waterfall structure as the requirements to be developed are collected, defined and written down, but there is a great tendency towards Agile methodology as these requirements, although written down, can change continuously to meet the customer.

Visual Design-Wireframe

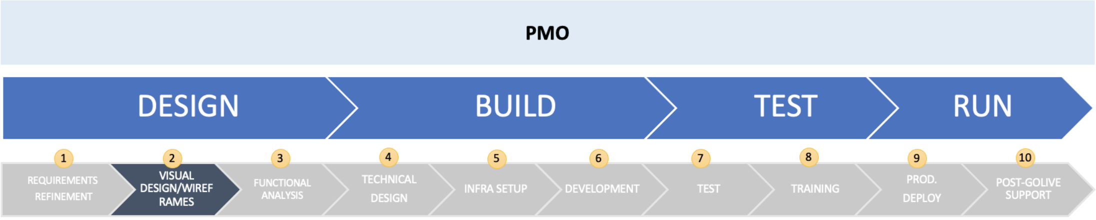


Figure 3.3: Visual design-wireframe phase

This phase (fig. 3.3) defines how the solution will be realised: wireframe and visual design. TA wireframe is a graphical representation of an application interface that focuses specifically on space allocation and content prioritisation, available functionalities and expected behaviours. Together with the visual design, it allows the user to identify in a complete and exhaustive way the features and functional specifications necessary to define the best user experience. This type of activity is fundamental as it encourages the sharing of ideas through design or co-design sessions.

Specifically, a series of workshops are conducted in which, based on its understanding, Accenture proposes a user experience that can be directly evaluated and refined during the meetings. The set of practical and actionable design rules for the development of the new digital experience is then defined. The navigation model, the interaction model and the logical structure for content selection, organisation and production will be identified so that the end result of the concept meets the real requirements of the stakeholder group. Low fidelity prototypes, also called wireframes, are used to illustrate the interaction dynamics of the interfaces of the new application. Their advantage is that they simulate real interaction, making them useful tools for subsequent testing and refinement. Next, the look and feel of the solution is established by creating visual assets to support the solution corresponding to the wireframes created. This phase will conclude with the implementation of the shared graphics.

This phase is mainly related to an Agile model, as it takes place during several face-to-face sessions between developers. In addition, this phase is iterated by involving customers to agree and improve the requirements, defining what will be the appearance and the main functionalities of the solution to be developed.

Functional Analysis

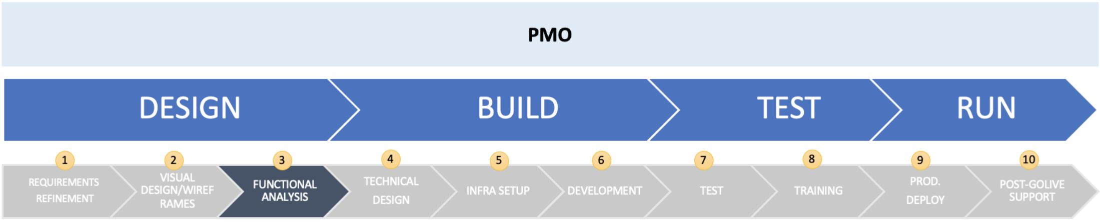


Figure 3.4: Functional analysis phase

This phase (fig. 3.4) is necessary to fully and exhaustively detail the functional characteristics and specifications of the product to be developed, in order to define the functionalities and processes that the offered solution will have to perform in fulfilling the requirements indicated in the conception stage.

In this stage, detailed meetings are held to define the functional aspects, the reference macro-logics of the calculation algorithms and the main processes that the system will have to perform. Furthermore, once all functionalities and their logic have been defined and rationalised, the test strategy will be defined to guarantee the quality of the code at the end of the software life cycle. Finally, the list of test cases to be executed in the relevant phase is drawn up.

This phase is the last step of the design phase, in particular it derived from the logical design part of a Waterfall approach, where all possible functional solutions are theorised until the functionality of the solution proposed to the customer is defined. Another aspect that tends towards the Waterfall approach, on the other hand, is the preparation of a formal documentation written and "signed" by the client that covers all the requirements gathered and the functionalities conceived.

However, as requirements and specifications may continuously change from the meetings carried out in the previous phases, there will be a need to keep these functional documents updated.

3.3.2 Build

The aim of this macro-phase, building on the functional architecture conceived up to this point, is to conceive the technical part and develop the solution offered.

Technical Design

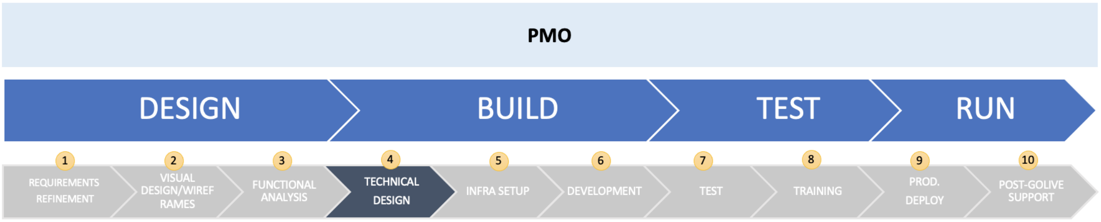


Figure 3.5: Technical design phase

A detailed technical analysis document will be drawn up for each module of the architecture described above, relating to the requirements that have emerged and to all the components included in the solution with reference to the technical modules of the application. The drafting of the technical analysis document will also involve sharing with the customer the reference application architecture for implementing the solution. The analysis and design phases envisaged therefore involve defining the application architecture and the necessary tools.

This phase (fig. 3.5) refers to the physical design sub-phase of a Waterfall approach, in which all previously conceived functionalities are transformed into concrete specifications. This phase is particularly important because, in addition to determining all the specifications of the application that will actually be created, it obtains the first and truly complete feedback from users. Therefore, like its predecessors, this phase is characterised by a Waterfall structure modified by a very communicative and collaborative approach towards the customer, typical of the Agile model.

Infrastructure Setup

In this phase (fig. 3.6), the customer will provide the Development, Test, Pre-Production and Production environments, after which the development team will make the necessary configurations to make them operative. These environments

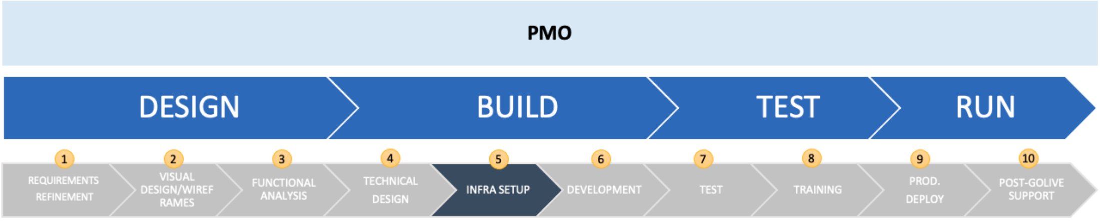


Figure 3.6: Infrastructure setup phase

are important for the implementation of the solution as they allow development and various test steps to be carried out without jeopardising the operation of the customer's production system.

This phase simply serves to prepare future user acceptance tests and can be remotely linked to the implementation step of a Waterfall model. On the other hand, this phase is almost entirely Agile, as communication with the customer and any changes are high on the agenda.

Development

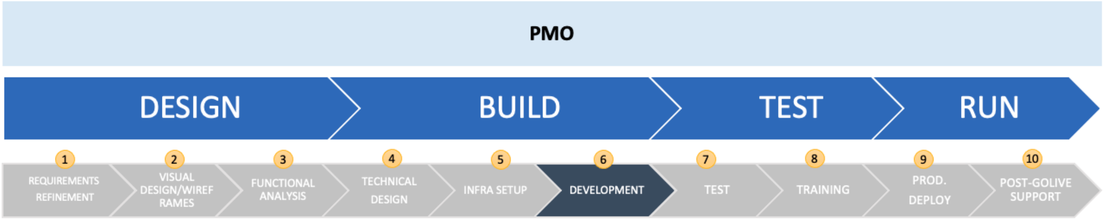


Figure 3.7: Development phase

In this phase (fig. 3.7), implementation and configuration activities are foreseen to make the new system operational for all functionalities in scope. In order to guarantee a high quality of the software, the *DevOps* chain will be used to manage the release and versioning of the application code. In fact, Accenture offers software release tools for code management and automation of build, test and deployment activities according to the DevOps methodology. DevOps is a software development methodology that takes advantage of new logics of sharing and collaboration but also of a more vertical crowdsourcing. It is characterised by a set of practices and cultural changes supported by automated tools and Lean Management processes, which automate the release of software with respect to its production chain, allowing organisations to rely on higher quality and secure software and applications much faster, to satisfy customers better and faster. The ultimate goal of DevOps is therefore to have software that can be released into production without disruption.

This phase combines the implementation phase of the Waterfall model with an Agile component thanks to the creation of the DevOps chain, in which the development and production teams no longer act separately. In fact, the two teams are merged into a unit in which technicians are active throughout the application lifecycle, from development and testing to deployment and production, and acquire a range of skills not limited by a single function.

3.3.3 Test

This macro-phase wants to verify that all that has been developed up to this point is working properly and that the customer's requirements are satisfied in full. Then, the aim is also to train customers in the use of the offered solution.

Test

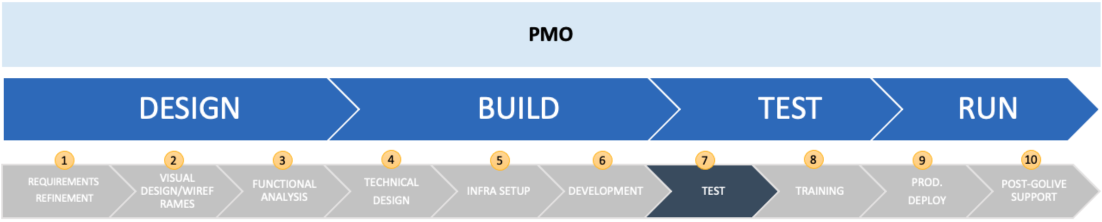


Figure 3.8: Test phase

This phase (fig. 3.8) is dedicated to the design and execution of the integration tests, to the definition of the test scenarios envisaged for the User Acceptance Test and to user-side support during the face-to-face test sessions expected. The test phases generally considered are described below:

1. **System Test**. This step is aimed at verifying all the system functions that will certify the correctness of the business rules applied in the various case management scenarios.

2. **Integration Test**. This step is intended to verify the proper integration between the new application and external systems with the aim of ensuring that the communication with the outside world and the implemented transformation logics are correct.

3. **User Acceptance Test**. This step is aimed at testing the developed software with the users who will be users of the system, after the live go-live of the

application. This test is a necessary milestone of the project and is crucial to receive feedback and confirmation of the developed solution.

This phase is associated with the Agile approach. In fact, since there are numerous changes in requirements and additions of functionality, the tests are repeated in an increasingly specific way and in a collaborative manner with the customer.

Training

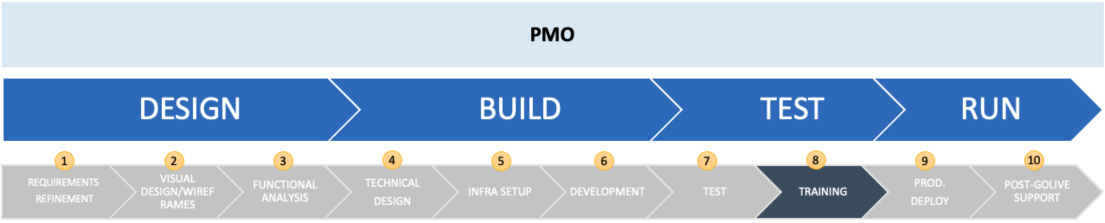


Figure 3.9: Training phase

In this phase (fig. 3.9), training is provided for users to enable them to use the developed platform independently. In particular, they are provided with material illustrating the main functionalities and highlighting use cases.

This phase tends towards Agile methodology and it can be understood as an initial part of the deployment phase.

3.3.4 Run

The objective of this macro-phase is to release the developed solution in all its functionalities and to guarantee full support and maintenance in the forthcoming period.

Production Deployment

The developed solution is released on the infrastructure provided for the production environment using the DevOps chain in order to make the release as automated and secure as possible. The activities foreseen for this phase include the preparation of the user manual that makes explicit all the characteristics and use cases of the application and the execution of the operational procedures aimed at the release of the components and a post operation audit.

This phase (fig. 3.10) mainly refers to the deployment phase of the Waterfall approach, made more agile by the use of the DevOps chain.

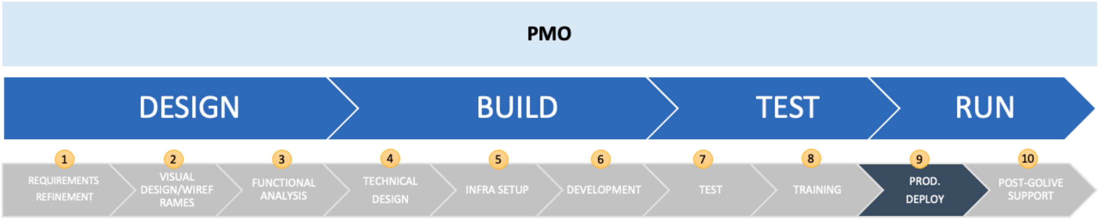


Figure 3.10: Production deployment phase

Technical Knowledge Transfer and Post Go-Live Support

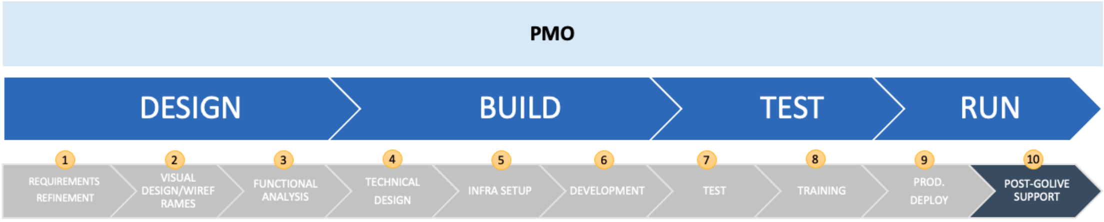


Figure 3.11: Technical KT and Post-Go live support phase

During the post go-live phase (fig. 3.11), the project team will perform knowledge transfer (KT) to the client company's core maintenance team in order to enable the latter to manage the future use of the software. In this phase, "problems" encountered by users will be analysed together to facilitate KT and to quickly perform potential post-deployment corrections to guarantee the released product. This phase can be related to the maintenance phase of a Waterfall approach.

It, on the other hand, adds a rather agile component compared to the one just mentioned. In particular, the Accenture team performs post-release support to ensure that users are completely satisfied with the developed product and how it works.

Table 3.1 concludes the chapter by summarising how each stage of the Accenture process belongs to the methodologies of project management theory.

Table 3.1: Methodologies followed by each phase

	Phase	**Predominant Methodology**
1	Requirements Refinement	Agile
2	Visual Design-Wireframe	Agile
3	Functional Analysis	Waterfall
4	Technical Design	Agile
5	Infra Setup	Agile
6	Development	Waterfall
7	Test	Agile
8	Training	Agile
9	Prod Deploy	Waterfall
10	Post Go Live Support	Waterfall

Chapter 4

Project Analysis

4.1 "Consumi Prescritti" project

This chapter will focus on the implementation of the "Consumi Prescritti" project, which was carried out during the working relationship between Accenture and a company that operates in the utility sector. In particular, it will analyse the various phases typical of the Accenture methodology covered in chapter 3.3 with reference to this project. At a later stage, the chapter will first identify the similarities and differences with the established Waterfall and Agile models, and then describe the advantages and difficulties encountered during the development of this project. But first of all, it is necessary to explain the context in which this project was undertaken and what is meant by its name.

4.1.1 Context

The context for this project is the world of utilities and, in particular, the integrated water system. The term "Integrated Water Service" refers to the combination obtained from the vertical unification of the various segments of the management of public services for the collection, adduction and distribution of water for civil use, sewerage and wastewater treatment. The company for which Accenture developed this project is therefore a company that administers the water system in Lombardy. In this scenario, the company asked Accenture for strategic and technological consulting aimed at managing and improving the efficiency of the systems and processes currently in use. However, this discussion will focus on a specific project developed within the customer's credit area. Specifically, the name of the project refers to the concept of "Prescrizione", but what does it mean? The dictionary defines the word "prescription" as *the process of acquiring rights, immunities or obligations as a result of the passage of time* [41]. In other words, many rights, especially those of a pecuniary nature, lose all protection if they are not exercised for a long period or, as they say, fall under prescription. The prescription is therefore the extinction of the right that occurs when the holder fails to exercise it for the time provided for by law. When the right is prescribed or time-barred, the holder can no longer make any claim and the debtor is not obliged to pay. In civil law prescription can be confused with the statute of limitation, in which case the first must be taken into account.

All households are equipped with essential services, i.e. those commonly identified with water, electricity and gas. In technical terms, we are talking about the so-called utilities, the regulation of which is entrusted to ARERA [1]. In practice, a company of choice supply customers with electricity or drinking water. Obviously, all this has a cost which is paid, regularly, following the issue of the relevant invoice or, better still, the so-called bill. It can happen, however, that the water supplier sends the consumer, by mistake, an old bill that was never actually invoiced by the company itself. The Italian Finance Act 2018 provides that, when a water bill is delivered relating to consumption that refers to a period of time prior to a certain number of years, the debtor has the right to refuse to pay it because the debt is now time-barred, or better, under prescription. Therefore, the debtor may decide, after filling in a special request form, to "opt out" [2] of the prescription, i.e. not to pay the specific debt.

Finally, another very relevant concept related to this project is that of debt collection actions towards the customer and, in particular, formal notice of payment or warning letters. In fact, if a customer continues not to pay his bills after a period of time fixed by law [3], he becomes insolvent towards the company. The company is therefore entitled to request payment by means of warning letters in which, first, it reminds the customer of the debt owed to the company, and then "threatens" him to take certain measures such as suspending the water service in the worst case.

4.1.2 Overview

In this context, the client company asked Accenture to create a process and a consequent software application able to optimise the management of consumption, and therefore of the bills, that were under prescription. This request was due to the fact that the client company did not consistently keep track of those customers who had the possibility to claim the prescription, which by legislation had to be shown to them in the corresponding bill. In addition, the current process was too manual and exposed to many risks of management errors and, therefore, had great

[1] Autorità di Regolazione per Energia Reti e Ambienti

[2] "Eccepire" in italian

[3] Regolazione della Morosità nel Servizio Idrico Integrato (REMSI)

potential for efficiency and automation improvements. In any case, the customer's requirements and the consequent functional analysis will be dealt with in more detail in the following sections.

In order to properly manage the debt collection phase as soon as possible, also considering the prescribed consumption, the implementation of the application was requested within a very tight deadline. In fact, the client company had to send out formal notices of payment as quickly as possible, taking into account the prescribed consumption, to all customers who were insolvent. In this case, the need was to adapt the management of prescribed consumption to include it in the warning letters, so that insolvent customers could proceed with the payment of their debts.

Therefore, Accenture decided to organise the project in two steps: a transitional phase and a full-scale phase. In the first phase, the main objectives were to consolidate prescription information within an ad hoc application and make it available to the credit function. The second step was aimed more at automating and improving the efficiency of the entire process. The division into two steps was mainly due to a time factor, so as to solve the general problem first and then make it as efficient as possible.

4.1.3 Plan

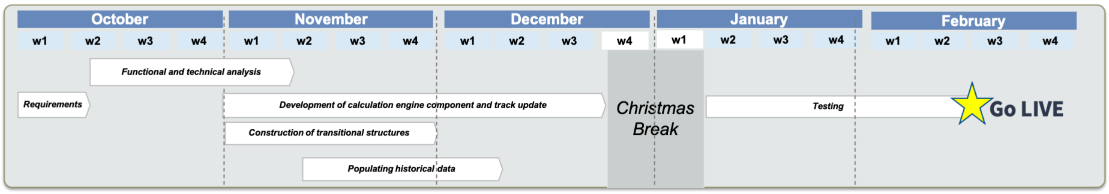


Figure 4.1: Consumi Prescritti Plan

The project plan devised and set out by the project manager is shown in figure 4.1. It shows the development phases of the project and the related timeframes which have been assumed for the realisation of the required system. Within the Gantt diagram, the phases of:

- **Requirements**: analysis and requirements gathering phase. It should be noted that the plan was produced after several meetings to gather requirements, so this phase was assumed to take only one and a half weeks;
- **Detailed Functional and Technical Analysis**. This phase was designed to last four and a half weeks, starting after the requirements had been gathered;
- **Technical Development of the Transitional phase**. This phase was

assumed to start towards the end of the analysis phase and would last exactly one month of work;

- **Development of calculation engine component and track update**: technical development of the full-scale phase, which includes the creation of a calculation engine enabling the automation of processes. This phase was the largest of the whole project as it created the structure of all the processes of the developed system. For this reason, it was also assumed to be the longest lasting with a duration of two months of work;
- **Populating historical data**: creation of the necessary database structures and loading of historical data.This step was designed to proceed during the structural development of the system and was scheduled to last four and a half weeks;
- **Testing phase**. The testing phase would start after the Christmas break and would last for different periods, depending on the progress of development and feedback from users. It was however assumed to last five weeks;
- **Release to the user**. The release of the system to the customer was scheduled for mid-February after exactly four and a half months of analysis and development.

Clearly, the realisation of the project followed a more complex design development which will be analysed in detail in the following sections. Within the plan, it was internally decided to show the client only the main phases and timescales.

4.1.4 Composition of the team and actors involved

The entire consultancy contract with the client company was overseen by a number of key Accenture figures, including an account manager, a service manager, a project manager and all functional, support and delivery teams. As Accenture's services included, in addition to this project, other major ones and ongoing maintenance of all the client company's systems, the team of professionals supplied was very wide and diversified.

Specifically, the project under consideration was led by a project manager who managed a functional team and a delivery team, also called development team. The project manager was particularly involved in creating the project plan, determining estimates and costs, coordinating the teams and organising the work. The functional and development teams were two parallel groups that interacted with each other to ensure the best value for the project. In particular, the former was responsible for the relationship with the client, the collection of project requirements, the construction of the functional architecture of the system and the testing and confirmation phase before the release. The latter was in charge of all the technical and development activities of the system.

With regard to the client company, Accenture interfaced on this project mainly with the billing area for the management of the prescription within the bills issued, with the credit area for the credit recovery actions such as the warning letters and, finally, with the IT area for the integration of the new system within the current ones.

4.1.5 Process

To explain the process by which the project was developed, Accenture's semi-agile methodological approach will be analysed step by step, according to the model specified in chapter 3.3.

Requirements Refinement

In this first phase, a series of meetings with the client were set up in order to try to understand and define the initial requirements for the project. Subsequently, meetings were held internally with the Accenture team to process these requirements, comparing them with the feasibility of development, and then proposing those again to the client for confirmation before initiating actual development.

First of all, the functional team gathered information on how the current process of managing the client company's prescribed consumption [4] was managed. Critical issues and points for improvement were then defined. Specifically, the concept of consumption prescription was analysed in the context of two distinct business scenarios and, for each of them, the operational model implemented by the client company was identified:

1. Management within the billing function of the prescription of consumption invoiced over two years;

2. Specific reference of bills with prescribed consumption in the warning letters.

Concerning the first point, the requirements collected referred to the request to replace the currently inefficient manual activity of extracting and managing the information on the bills with prescribed consumption, carried out at the end of each billing, with a special automated functionality parallel to the billing process. For the second point, it was requested that in the case of a letter of formal notice, information on bills with prescribed consumption should be available in order to

[4] To indicate the prescription applied on consumption, the concepts of prescription, prescribed consumption, prescribed volumes and short prescription will be used as synonyms

show them in the summary on that letter. In this case, too, the debt collection process by means of warning letters was not efficiently handled. On the basis of these two processes, the following overall requirements were also added, completing the basis on which the whole project was developed.

- **Automation of the calculation process**. In particular, the requests were related to the automation of the identification of customers to whom the prescription was applied, the automation of the calculation and inclusion of those customers' information in the bill and the reduction of the resulting billing process time.
- **Automation of customer request management**. In particular, the requirement concerned the monitoring, within the company's various systems, of the customer's claim for prescription and its complete administration. This implied a complete consolidation of the prescription management process.
- **Structured management of information transmission to the customer**. In particular, the need was related to the availability and usability of prescription information for all credit management and customer relationship processes.
- **Process tracking for regulatory purposes**, i.e. consolidating information on the whole process in order to be able to use it in case of internal or external audit.

All these requirements were then collected in a specific document. The main requirements and related points for improvement can be observed in the table 4.1. The document produced presented a detailed analysis of all requirements, as well as some real examples, use cases and structural model hypotheses. Within that, the project plan was also presented, containing a timeline for implementation. This document was then reviewed and confirmed by the client in order to proceed with the correct development of the project application, without forgetting any details.

Visual Design-Wireframe

Based on the results of the previous phase, several meetings were held within the project team in order to define the first functional specifications needed to transform the requirements into development objectives. By functional specifications are meant sets of functionalities or services to be provided by the system. They also describe the behaviour of the system in the face of particular inputs and how it should react in certain situations. To this end, the skeleton of the entire optimised process and the first ideas for the development of the prescription management application were hypothesised. The form of the process will be dealt with in detail

Table 4.1: User requirements

AS IS	Improvement Aspects
No structured management of prescription information	Need to manage and consolidate prescription information within information systems, so that it can be used by all business processes
Manual extraction and calculation of prescribed consumption	The manual nature of the process makes it more fragile and prolongs the process time.
Manual verification of customers' applicability to prescription	The manual nature of the process exposes the customer to the risk of non-compliance or dispute.
Manual adjustment of the bill to add the indication of prescribed consumption	Need to make the modification flow unique, eliminating defect risks
Manual administration of the process of contesting the statute of limitations by the client	The manual nature of the process makes the management of the related credit more complex
No prescription management in warnings	Legislative necessity to indicate prescription in warning letters

in the following phase. As far as the application is concerned, it was decided at this stage to add it as a new feature in the management software currently used by the client company. No prototype or wireframe was foreseen, but the structure of how the foreseen software should work was dictated in a general way. The functional specifications which have been defined in this phase can be summarised as follows.

Moreover, it was planned to create a specific table at database level to collect information, for each bill considered, on the prescription period, the prescriptive consumption and the prescriptive amount. This table must be populated after the billing phase. By storing the prescribing information in the database table described above, the information was then available for all business processes. Finally, for an automated and optimised visualisation and management, this information should have been available within the "Consumi Prescritti" application created for the situation.

Functional Analysis

This phase is crucial because it brings together the requirements gathered and studied up to this point with the structural hypotheses designed in order to define these requirements in a structured and coherent way, creating a document that tries to give a global vision of the project's objectives. In fact, the features that the software, to be developed in the implementation phase, will have to provide were determined at this stage. Another important aspect was also that at this stage the project team, over the course of several meetings held alongside the client company, all potential impacts with other systems and processes involved in everyday business were analysed. The document created was therefore the heart of the whole project, as it allowed to explain all the aspects related to the development of the project, from the initial requirements to the technical specifications of the software that would be created. It should be noted, however, that the technical specification part was determined in the next phase and only added to the document afterwards. It must also be said that this document, as said, was the result of numerous meetings, both internal to the functional and development teams and with the technical department of the client company.

The document was therefore composed as follows:

1. User requirements;
2. AS IS operating model;
3. Transitional operating model;
4. TO BE operating model[5];
5. Technical solution[6].

User requirements This first section of the document merged, listed and described the requirements that had been gathered through the user meetings. First of all, the structural requirements were set out, i.e. the general requirements concerning the architecture and functions of the new system. The model that would then have to be implemented for the management of the prescribed consumption and the consequent logic of applicability were then outlined. In fact, the model created, through specific tables created at database level, had to recognise the bills containing the consumption that had become prescribed for those customers who were actually entitled to it. This process of identifying applicability had to take place before the billing function, in order to determine and insert the prescription within the bills of these customers.

The general requirements were then defined, i.e. those relating to the automation of the process of calculating and managing prescribed consumption. Finally, the

[5] Also referred to as full-scale model

[6] Document section that was added in the successive phases

need to include information on the prescription in the warning letters was also outlined. Also in the latter case, it was important to identify those customers who were insolvent but had consumption prior to two years ago and were therefore entitled to prescription.

AS IS operating model At the beginning of the project, the management of the short prescription within the processes of the client company was structured as follows (figure 4.2).

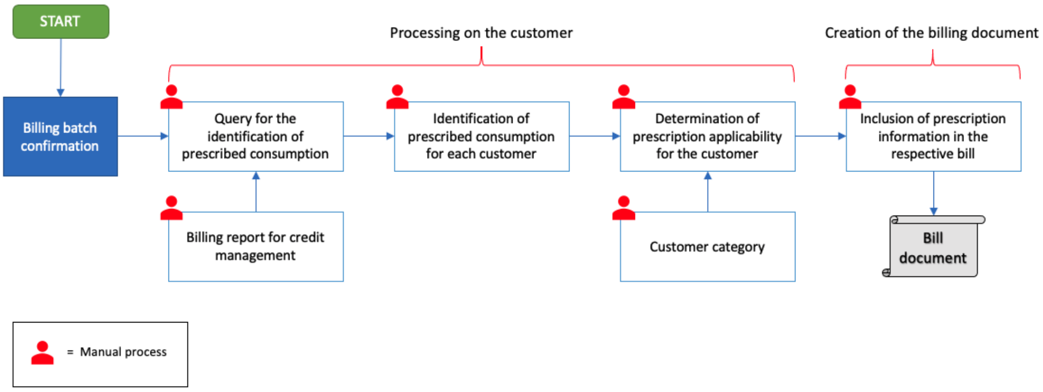

START
Processing on the customer
Creation of the billing document
Billing batch confirmation
Query for the identification of prescribed consumption
Identification of prescribed consumption for each customer
Determination of prescription applicability for the customer
Inclusion of prescription information in the respective bill
Billing report for credit management
Customer category
Bill document
= Manual process

Figure 4.2: AS IS Process

After the issue of a batch of bills, two separate procedures were launched by the IT department, which produced an Excel file containing the "net value" subject to prescription for each bill issued belonging to the batch. Then some manual checking procedures had to be carried out in order to detect any errors and identify the classifications of customers entitled to the prescription, excluding the others. Subsequently, starting from this file, it was necessary to manually fill in an attachment to the bill, which would then be sent to customers, containing all the information relating to the prescription.

With regard to the actual process required to recognise and formalise the client's objection of volume prescription, it follows what has been studied and explained within the analysis document. In order to manage the case in which a customer decides to make use of the right, i.e. decides to object the prescription, following the transmission of the necessary documentation by this customer, the request had to be created by the front office and assigned to the billing function for processing. After procedurally verifying the appropriateness of the claim and the documentation provided by the customer, the client company had to delete the previous bill and create a new one, the total amount of which had to be reduced by the prescribed amount as the customer was not under any obligation to pay for it.

Finally, it was concluded that there was currently no mechanism in place to make the information available in an automated and consolidated manner to the credit management and the prescription data was not managed within the customer's warning process.

Transitional operating model The transitional model, as mentioned above, was designed because of the short timeframe in which an initial implementation was required to manage the sending of warning letters to consumers indicating consumption under prescription. Therefore, this model was not going to decline all the customer's requirements in a definitive way, but was going to find a temporary solution but still effective in the required deadlines.

The transitional model therefore included the following process, which evolved from the AS IS process:

1. Launch of the procedure for identifying bills with prescribed consumption as in AS IS, the result of which is enriched by the addition of customer information suitable for identifying the applicability of the prescription;

2. Historization of the information obtained in a dedicated database data structure;

3. Creation of a user interface, implemented as a web application, necessary for the management of the information obtained. In which, with regard to the prescribed consumption, users could perform checks on calculations, process individual bills with prescribed consumption and extract the information to prepare the respective bills to be sent to customers.

The whole process detailed in the various complementary layers is outlined in figure 4.3.

It must be said that all information, both historical and current, collected in the database was viewable within the application created and available for possible modification and confirmation. In fact, the transitive process was such that users had to manually import the excel file, extracted as in AS IS, into the application. Subsequently, this allowed an effective visualisation in order to check all the information necessary for the management of the process, a punctual or massive modification of the data shown and, finally, a validation of them before moving on to the invoicing function.

As for the warnings, they were processed by adding the data contained in the database and related to the prescription. In particular, in such letters, a new section was introduced referring to the customer's right to claim prescription, or the amount the customer was warned to pay was reduced by the amount prescribed if the customer had already claimed prescription.

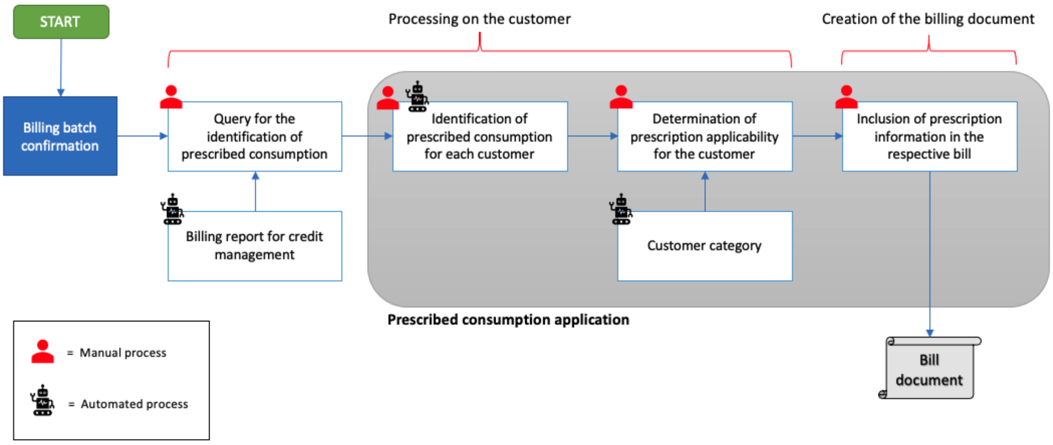

START
Billing batch confirmation
Processing on the customer
Creation of the billing document
Query for the identification of prescribed consumption
Identification of prescribed consumption for each customer
Determination of prescription applicability for the customer
Inclusion of prescription information in the respective bill
Billing report for credit management
Customer category
Prescribed consumption application
Bill document
= Manual process
= Automated process

Figure 4.3: Transitional Process

However, for the sake of simplicity, all specific functionalities provided to the customer related to the technicalities of managing bills with prescribed consumption are not listed

TO BE operating model The TO BE model was needed to evolve the transitional model by automating and optimising the whole process, as depicted in figure 4.4.

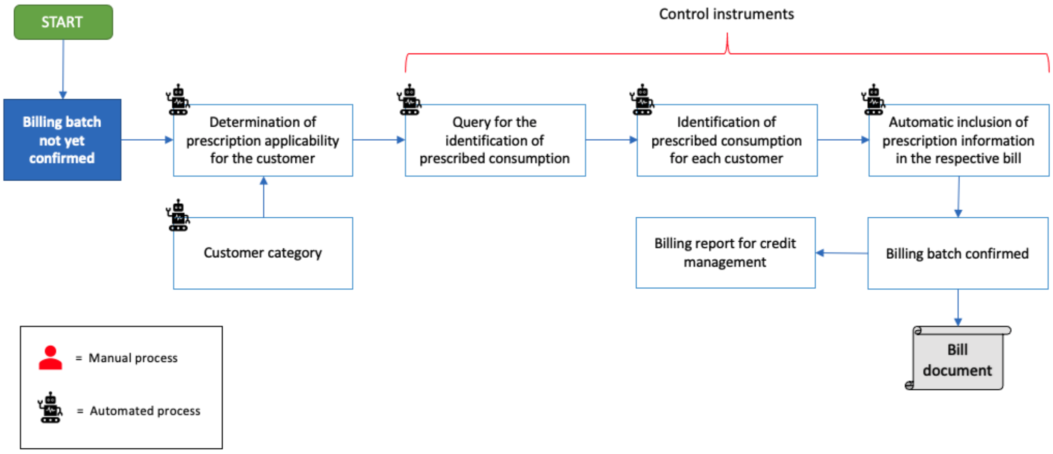

START
Control instruments
Billing batch not yet confirmed
Determination of prescription applicability for the customer
Query for the identification of prescribed consumption
Identification of prescribed consumption for each customer
Automatic inclusion of prescription information in the respective bill
Customer category
Billing report for credit management
Billing batch confirmed
Bill document
= Manual process
= Automated process

Figure 4.4: TO BE Process

In particular, the calculation of any prescribed volumes became automatic during the billing function. The tables and functionalities described in the transition will be evolved in the TO BE model and will allow the management, again automatically and at a stage prior to billing, of the correct application of the prescription. Therefore, the field was entered as an input during the billing process, making it possible to optimise the calculation of the prescribed volumes only for those customers who might actually be entitled to them. In any case, the functions foreseen in the transitional period for the management of this field were maintained even after the calculation.

Finally, all information concerning the prescription was inserted automatically both in the invoice document to be sent to the customer and in the warning letters.

Technical solution This step will be discussed in more detail in the next section.

Technical Design

This phase closed the functional analysis phase by describing all specifications necessary for the start of development. Therefore, a technical document was drafted which was later included in the functional analysis document. It explained how the requirements would be transformed into the final solution. First, an event catalogue was prepared, which tried to list and describe all events that would determine the realisation of the operational model in order to understand which systems are impacted and how.

The events detected were as follows:

1. Implementation of data structures and logic of transitional prescription model, consolidating logical and process information in the management software;

2. Adaptation of data structures and logic of prescription model TO BE;

3. Population of prescription database with historical data, in order to get all the information about the bills with prescribed consumption;

4. Implementation of user functionality in the application for the transitional model, in order to manage prescription information;

5. Adaptation of user functionality of TO BE model;

6. Creation of communication interfaces between systems, through the creation of interfaces allowing the connection between the management software, the database and any other affected software;

7. Revision of warning letter text and prescription data insertion;

8. Creation of functions for the management of IVA on prescription and feeding logics, important for credit accounting management.

As could be seen from the functional analysis and the event catalogue, the technical analysis was also divided into two parts: one relating to the transition model and one relating to the full-scale model. The whole technical document, being part of the functional analysis document, was then shared and discussed with the client, so that it became part of the whole planned system.

Transitional operating model This first part of the document described in a technical way everything that would be implemented for the transition model of the prescribed consumption project, i.e. all that section that would have been part of the first release to users. It was necessary to draw up a technical part of the transition as this too was discussed and reviewed with the client before the start of development.

First of all, the database tables in which all prescription information would be stored throughout the project were listed and explained. These tables had to be in communication with each other and with the various software, allowing the system to access all the necessary and relevant data for each process. The queries created to allow the extraction of the prescription information and the subsequent modification and adjustment of the warning letters were then described. Finally, at the code level, the document explained all the steps that would be taken to create the prescribed consumption management application, along with all the relevant user functionality. All data and information flows were also studied and exposed here. As mentioned above, for simplicity's sake they have not been described specifically.

TO BE operating model In the next part of the technical document, further implementations were listed that improved the transition model and determined the steady state model. As already mentioned, these improvements mainly concerned the automation of most processes.

Firstly, the data structure of the database was to be improved in order to ensure process automation. For example, a table was created which would collect information on the customer's eligibility for a prescription from the CRM system, in order to automate the applicability or otherwise of the prescription on these customers' bills. This improvement also had an impact on the setting of warning letters. Finally, the billing engine was modified to include in its calculation the definition of the prescribed amounts so as to automatically modify the bills that come out of this process, following the request to contest the prescription by the customer concerned. The printing flow of these bills has also been modified to allow this modification.

Infrastructure Setup

In this phase, the development team together with the IT department of the client company set up the development, testing, pre-production and production environments, in which the functional team, in the testing phase, would then test and validate the developed applications.

In short, the release of what was developed would be organised in the following way: the first transfer of code would be in the development environment, which is the most volatile and constantly changing by the development team. Once the first version was completed, it would be transferred to the test environment, where the functional team would, as mentioned, carry out the relevant tests. Once the processes had been validated, the code would first be transferred to the pre-production environment to confirm that it was working properly and then, in the production deployment phase, to the production environment to finally be available to the customer.

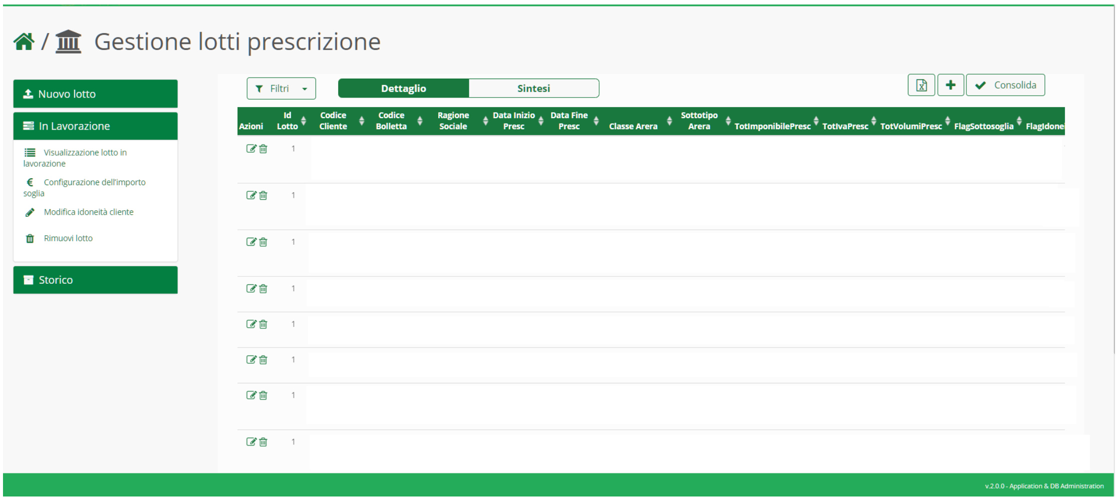
Development

At this stage, the ball was in the development team's court, who, with the support of the functional team so as not to miss a single facet, had to start writing the code and creating all the functionalities that would define the final system. This was the longest phase and the one in which the client was least involved, but obviously also the most important. It should be added that the development phase would not really end at the end of the implementation, but would continue throughout the testing and first maintenance phase, in order to give maximum correctness to the code and to consider last-minute requirements.

It is important to clarify that from this point onwards, two iterations have been carried out in relation to the phase performed, the first for the implementation of the transitional model and the second for the steady state model. Figure 4.5 shows, as an example, the user interface of the application developed to manage prescribed consumption

Test

In this phase, the functional team went into the customer's shoes, recreating all the processes and functionalities that characterise the implemented system. The test scenarios were taken from the functional analysis, where they were anticipated and updated according to the latest changes. In this case too, the testing phase was carried out at two different times, first to confirm and correct the transient model and then for the steady state model.As required by the semi-agile methodology, three types of tests were carried out.


Gestione lotti prescrizione
Nuovo lotto
In Lavorazione
Visualizzazione lotto in lavorazione
Configurazione dell'importo soglia
Modifica idoneità cliente
Rimuovi lotto
Storico
Filtri
Dettaglio
Sintesi
Consolida
v.2.0.0 - Application & DB Administration

Azioni	Id Lotto	Codice Cliente	Codice Bolletta	Ragione Sociale	Data Inizio Presc	Data Fine Presc	Classe Arera	Sottotipo Arera	TotImponibilePresc	TotIvaPresc	TotVolumiPresc	FlagSottosoglia	FlagIdonei
	1												
	1												
	1												
	1												
	1												
	1												
	1												
	1												

Figure 4.5: Prescribed consumption application

System test In this first test step, the correct functioning of the prescription management application was tested in detail and prototypes of warning letters were created to display the correct introduction of prescription information. In addition, the development team determined that all processes were smooth and there were no hiccups whatsoever in the various business situations. Any problems encountered were reviewed with the help of the development team for an update of the systems. The same procedure was also carried out for the fully operational model, taking into account different or modified business situations. In both test iterations, but especially in the second one, specific calculation tests were performed to verify the correct determination of the prescribed amounts and all related data.

Integration test In this sub-phase, with regard to the transitional model, it was verified and tested the correct communication and integration between the application developed, the management software in which it was inserted and the database that stored all relevant data. In the second test phase, on the other hand, the verification of the integration also with the billing calculation engine and the process of creating warning letters was added. Here, too, in case of incorrect functioning, the development team made adjustments in this respect.

User Acceptance Test This step was a very important milestone for the project. Several meetings were organised between the Accenture team, both functional and development, and the client company's managers. During these sessions, the users performed all the process and system tests with the support of the functional team and were informed about all the functionalities that had been implemented. Any last minute changes and requirements were also collected, which would be implemented and re-tested before release to production. This sub-phase and the related meetings were also carried out for both the transitional and the full-scale model.

Training

During the development of the "Prescribed Consumption" project there was no real training phase, i.e. no meetings were organised specifically for this purpose. On the other hand, the user was instructed on the use of the new application created both during the user acceptance test phase and by means of specific documentation, such as the relative user manual. As regards the other processes adjusted to take account of the prescription, the user was continuously followed and supported during their use in production. This constant support, very agile, obviously took place both for the transitional model and for the steady state model.

Production Deployment

Finally, the release phase of what has been developed has been reached. This took place in two phases: one for the transitional model and one for the full-scale model. As mentioned above, the production release took place only after the pre-production code had been transferred to check for any faults. At the same time, all new operations and business processes that would have to be executed once the application was available were reviewed with the client, as will be seen in the next phase.

Technical Knowledge Transfer and Post Go-Live Support

During this phase, the development team proceeded with the knowledge transfer to the client regarding the technical maintenance of what had been developed, while the functional team explained to the client the new management of operational processes, such as when to load prescription information into the database history, or how to modify the new warning letters.

4.2 Project insights

The aim of this section, after summarising the main aspects of the project management methodology used, is to describe what positive and negative traits were encountered during the course and conclusion of the project and possibly propose solutions to be implemented to resolve or improve these areas.

4.2.1 Similarities with Waterfall and Agile models

As outlined in chapter 3, the Accenture methodology lies somewhere between a Waterfall and an Agile approach. During the description of the Accenture project methodology, an attempt was made to find analogies with the two main models while describing the various phases encountered. This section aims to explore in more details the similarities and differences that have been identified in relation to the completed project.

The methodology adopted for the development of the "prescribed consumption" project was composed of a Waterfall skeleton with some Agile components. In fact, the main phases that determined the progress of the work were fixed and non-iterative, proceeding without ever having to repeat an entire phase. In addition, formal documentation was drawn up during the early stages of the project, enabling both the Accenture and client teams to see the system design in black and white. The formal documentation included, first of all, the project plan accompanied by the cost and time estimates that would be undertaken. Moreover, at the end of the initial analysis phases, detailed documents were produced aimed at outlining all the actions and changes made along the way. No assessment was carried out to determine when a phase was actually completed and the next one could be carried out, since the project's life cycle was somewhat unconstrained.

On the other hand, it should be noted that there was a tendency towards an Agile approach due to the number and frequency of meetings with the client, also due to the way in which these meetings were conducted. In fact, the health situation has led to a complete transition to web-based work from home, theme which will be explored in more detail in section 4.2.3. Thus, it was not only the number of meetings that brought the project methodology closer to an Agile approach, but also, on the one hand, the tendency to work together with the client with a high level of communication and, on the other hand, the frequent iterations of phases as a result of changes of direction by the client company regarding the project requirements. These situations occurred especially in the requirements gathering and testing phases.

As far as the testing phases are concerned, coordination between the different approaches applied must be considered. In fact, unit testing was carried out in a totally Agile manner, verifying any added functionality, while integration testing and UAT were carried out in a Waterfall manner, defining the various test moments in a structured way.

4.2.2 Results

The purpose of this first paragraph is, not so much to outline the results obtained by the master thesis work, but to illustrate what effects have been produced by the conclusion of the project itself.

On the whole, the "prescribed consumption" project has brought excellent results, succeeding in fully satisfying the client and structuring a process of fundamental importance that was previously not managed or managed superficially. In fact, a systematic and organised process for managing bills and customers with prescribed consumption has been created. In particular, the transitional model succeeded in allowing the billing function of the client company to efficiently and correctly determine what consumption was under prescription and to manage this data through a single application. This also meant that it was possible to improve the process of dealing with requests from the client company's users for claiming these amounts in a more efficient way. In addition, it was possible, downstream of the management of these bills, to proceed with debt collection actions, such as formal notice of payment. Subsequently, the last manual actions still performed by the operators, that were still possible causes of human error, were also eliminated with the release of the full-scale phase of the project. In this way, the whole process was definitely automated and structured. The definition of the logic behind the project then created the starting point to begin the study and analysis of the client company's entire credit process. This, in fact, became the new project requirement of the client company. In conclusion, it can be said that the requirements were largely met and the project, as a whole, achieved a very high quality result. The clients were very satisfied with both the outcomes delivered and the approach that was executed.

On the other hand, it is also relevant to look at what were the most problematic aspects encountered during the progress of this initiative. First of all, it must be said that the Accenture teams that worked on this project were partly composed of young and relatively inexperienced people in the world of integrated water systems. There was therefore a need for a period of adjustment and initial training in order to be ready to manage and carry out the work in a strong and reliable manner. Another crucial aspect relates to the project methodology approach used. On this regard, there were experienced some issues concerning the continuous change of project requirements by the client company. This was due on the one hand to a high number of unstructured workshops which led to an over-delivery of information to the client company's team causing numerous reconsiderations and general confusion about the process. On the other hand, these issues were linked to general communication challenges due in turn to the lack of face-to-face meetings. This latter aspect will be better analysed in the next sections.

4.2.3 The Smart Working factor

A crucial factor which has affected not only everyday life but also the realisation of the project topic of this work is the global pandemic the world has been experiencing over the past year. Apart from the obvious health implications, what has most influenced the labour environment is the adoption of remote working as the main approach to work. In particular, Accenture opted for a total work from home policy for all employees, which is the reason why the "prescribed consumption" project was developed in its entirety not in presence. In addition, the different members of the project team, due to the inevitable return to their own residences during this period, were scattered all over Italy without the possibility of physically setting up a working table.

This whole scenario led to both positive and negative implications as far as the conduct of the project was concerned. This section aims to analyse in detail what these were. It should be pointed out that in this section the pros and cons of agile working will only be listed and explained with regard to the project under analysis and not in a general way.

First of all, it should be mentioned that Accenture project teams are often made up of various sections located in different parts of Italy. As described in section 2.2.4, for instance, the technology centres where development specialists work are located in southern Italy, while the innovation centres where functional experts work are mainly located in northern Italy. For this reason, Accenture was already prepared to carry out the project with a certain degree of web-based communication. In addition, the project started around the beginning of October, so Accenture, like the client company, already had time to prepare for smart working. However, this does not mean that the project was carried out in the same way and with the same results as in a conventional environment.

On the positive side, the most important is certainly the drastic reduction of commuting time to work. This has meant that hours have been saved on the project in question, allowing workers to be ready to start their day within minutes. In addition, through smart working, working hours have for obvious reasons been extended by being able to be contactable almost all day and being able to discuss and solve a problem as soon as it arises. However, this aspect can clearly be seen in its negative conception as well. Another advantage is the savings due to the cutting of physical locations, with a reduction in rent and electricity, water and gas costs. An advantage that has, in a certain sense, made it possible to expand the project budget.

On the other hand, as regards the negative aspects of working from home in relation to the "prescribed consumption" project, the increased difficulty of communication has to be mentioned as first. In fact, it was obviously not as effective and qualitative as at a work table. This applied both to the internal work between the team members and to the dialogue and workshops with the client. It is important to specify that Accenture's consultancy work is often implemented directly in a space provided within the client company, allowing for a very agile relationship and approach. This situation has consequently led to a considerable number of almost daily meetings via the web in order to allow work to be as organised as possible, but often leading to confusion and misunderstandings. This last point still led to several changes of requirements and decisions by customers, resulting in a constant analysis and costly, in terms of money and time, re-engineering of processes and solutions.

4.2.4 A new way

As stated in section 4.2.3, the project work undertaken by their own houses led to some communication issues. These problems, in the project management context, resulted in the following consequences. Especially in the early stages, such as requirements analysis and construction design, this often unclear and non-qualitative dialogue with the client led to vague requirements that were never confirmed with any degree of certainty. In fact, the phases of requirements gathering and analysis characterised by an almost joint work with the customer is a typical aspect of the Agile methodology, where the needs expressed by the customer are not fixed and can change as the work progresses. This obviously has its positive aspects, especially from the client's point of view, but for a project team accustomed to a hybrid methodology, it had negative results such as a process study that was never entirely definitive and a technical development characterised by continuous re-engineering stages. The almost daily poorly structured meetings and web-based communication led the client company and the functions involved in the project to follow the progress in an unclear and inconsistent way, resulting in numerous variations of views on the requirements and functionalities. Costs and implementation time, due to the constant alterations required, obviously increased and would become burdensome especially as the complete solution approached. Therefore, what was conceived thanks to the analysis that led to the drafting of this master's thesis, was the adjustment of the approach to some phases of the Accenture project methodology with a view to fitting the peculiar work situation. The solution offered to overcome the problems outlined above can be divided and described in two different analyses. In order to solve the problems arising from the inconsistency over time of the specifications collected from the customers, a new requirements acquisition process was proposed thanks to the mechanism, typical of the Agile methodology, called Requirements Change Management. In addition, a way of improving communication via the web between the various project teams was also studied, in order to avoid the creation of misunderstandings and other problematic issues. The implementation of this last proposal also led, as will be seen, to the optimisation of the testing phase.

Requirements Change Management

One of the most serious causes of software project failure is requirement changes and late elicitations. Managing these changes throughout the project life cycle, especially at later development phases, is a challenging task. RCM can be defined as the process of dealing with the requirements changes that occurs during or after software development. These changes cannot be well managed without a well-designed process model.

First of all, it is important to identify more carefully the various sets of elicited requirements that may arise during the early stages of the project. For this goal, it is proposed to use the scheme offered by Hori et al. [42] on figure 4.6, where the vertical axis indicates the requirement elicitation rate of each project component determined by the project architecture of the system and the horizontal axis represent the project progress time. The aforementioned rate is defined as:

$$\text{Requirement elicitation rate} = \frac{cumReq}{allReq} * 100$$

where *allReq* is the total number of requirements gathered for the software component until the end of the project, and *cumReq* is a cumulative number of requirements gathered until the target elapse date. This results in three types of identifiable requirements:

- **E-type**: early maturing type. This type is recognised and collected at an early stage of the project.
- **L-type**: later period maturation type. This type is characterised by a continuous process of requirements elicitation during project development phases.
- **U-type**: type of unexpected maturation. They are represented by the process of requirements elicitation that is caused in an unforeseen way in the phases following development.

It is thus obvious that requirements may change during the project lifecycle for a variety of reasons. Because of this, a method must be developed to understand and anticipate these inevitable changes during the development process in order to reduce the possible risks. The identification of factors that cause or influence requirements uncertainty is, therefore, a necessity. While type E requirements are

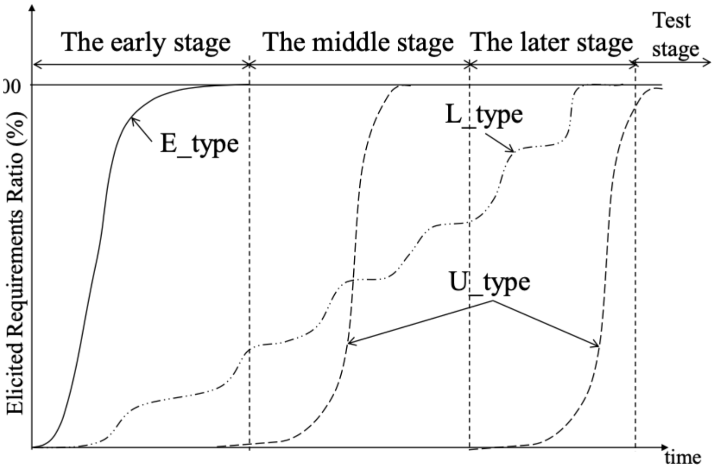


Figure 4.6: Types of elicited requirements

desirable, as eliciting all system requirements at an early stage of development is beneficial, the other two types, if not identified and managed in time, can lead to major problems in project progress.

McGee and Greer [43, 44] use five areas/domains to classify the causes of change. In particular, two of these change areas were recognised as problematic within the analysed project: customer organisation and requirement specification. The first one refers to changes in requirements triggered by events and uncertainties arising from an individual customer and its procedural uncertainties. Such changes have a tendency to influence the client's needs and, consequently, to influence the design and requirements of the project. The second category relates instead to requirements that are triggered by events and uncertainties related to the requirements specification itself. These events are based on a developers' point of view and their best understanding of the problem space and the resolution of ambiguities related to the requirements.

Given this, RCM is an inevitable tool in any development project, being useful to identify which factors may cause these changes. The knowledge gained through such findings will enable all actors in a project to better manage changes when they occur, develop change-based systems and anticipate certain changes.

The RCM approach that is proposed in order to try and improve Accenture's design methodology is intended to modify the Requirement Refinement phase and its impact on subsequent phases. Therefore, a new way of structuring this phase is recommended through a process adapted from the model proposed by Pandey et al. [45]. The new phase, as shown in figure 4.7, will be characterised by the following process:

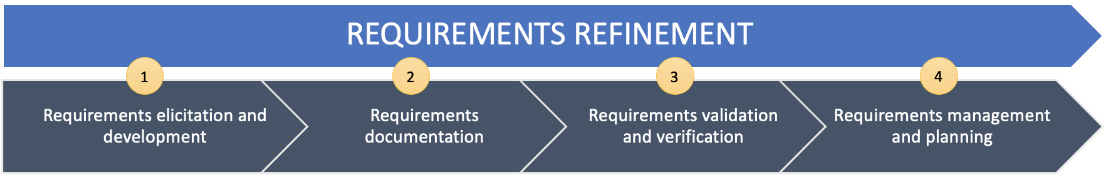

REQUIREMENTS REFINEMENT
1
Requirements elicitation and development
2
Requirements documentation
3
Requirements validation and verification
4
Requirements management and planning

Figure 4.7: New Requirements Refinement phase

1. **Requirements elicitation and recognition**. This phase mainly focuses on examining and collecting the desired requirements and objectives for the system from different perspectives, such as users, constraints and system operating environment. The requirements elicitation phase starts with the identification of system stakeholders and the collection of type E requirements from various points of view. Next, an attempt is made to understand, with the help of the client, what other needs may emerge in the future and then to identify the L-type requirements. In this stage, negotiation, agreement and communication become very important activities, hence, structured and well organised meetings with all stakeholders of the client company have to be organised.

2. **Requirements documentation**. During this second phase, a formal requirements document, or as described by the Waterfall methodology, an SRS, must be prepared.

3. **Requirements validation and verification**. When all verifiable requirements have been described and specified in the SRS, then the different parties involved must agree on its nature. It must therefore be ensured that the correct requirements are stated (validation) and that these requirements are stated correctly (verification). For this phase, some techniques for requirements validation shall be used, such as technical reviews of requirements with interested parties and prototyping. Verification of the SRS includes the correctness, consistency, unambiguity and comprehensibility of the requirements. At the end of this phase, a sort of requirements traceability mechanism is created that can generate an audit trail between the agreed requirements and what is changed during subsequent phases.

4. **Requirements management and planning**. The requirements management and planning phase monitors and tracks changes in agreed requirements, relationships between requirements and dependencies between requirements documents and other documents produced during project development. It is a continuous, cross-cutting process that starts with requirements management planning and continues with activities to identify and control changes during

and after the development and project maintenance phases. In this stage, U-type requirements must also be intercepted and managed. The actual process of requirements change management must be applied, through the ability to manage changes to requirements during the project development life cycle. Change management includes identifying, analysing, prioritising and deciding on the implementation of a change, with the possibility of validating or not validating change requests.

Web based communication

In order to manage communication with the client in the best possible way and make the new requirements analysis phase more stable and organised, it is suggested to schedule progress meetings on a weekly rather than daily basis. The concept is to use these workshops in a useful and positive way for the project, so that it did not disrupt what had been done so far. Therefore, efforts had to be made to exploit the advantages that web-based communication, through platforms such as Zoom, Microsoft Teams and Webex, offers. These benefits are reflected for instance in involving as many stakeholders as possible, so as not to risk receiving late feedback, and consequently possible problems, from other areas of the client company. In addition, it is possible to share detailed plans and show all progress, technical and non-technical, via screen sharing more quickly and efficiently. It is important to emphasise that the advantage of not scheduling these meetings too frequently is that it avoids sharing too many technical details that can confuse clients and cause them to constantly change their vision of the project. During these sessions, following the previously introduced principles of the RCM, changes in requirements can also be discussed in a structured way through the prioritisation and validation processes.

Consequently, a slight modification to the testing phase of the Accenture methodology is also proposed. In fact, instead of having single UATs performed before the final release of the product, it is now suggested to integrate unit tests with user acceptance tests, in order to organise review and acceptance workshops throughout the final stages of the project. Thus, advancements are seen and verified, increasing the collaboration of all interested parties and reducing the severity of possible defects. This practice effectively reduces the need for changes due to such defects.

Chapter 5

Conclusion

This study sought to take a critical look at the ways in which modern consulting firms initiate, develop and conclude a project, following the main theoretical approaches of project management.

First of all, the emergence and concretisation of the project management discipline in all industries was presented. To accomplish this, the most popular standards and models in the world were studied, with a focus on Waterfall and Agile methodologies. These two approaches are in fact the most famous and used in the project management scene and especially in the software environment. Therefore, the main characteristics and, most importantly, the composition of the model's life cycle were investigated, with reference to the major advantages and disadvantages of their use.

Afterwards, following the presentation of the technology consulting company chosen as the protagonist of this study, Accenture spa, the project methodology that distinguishes it in all its divisions around the world was studied.

This led to the most important part of the study in which, in essence, a project carried out by this company was followed from the inside. Therefore, the outcomes from the finalisation of this project are reported and the issues encountered at methodological level in its progress are examined in detail. On the one hand, it was found that the greater tendency towards the Agile approach at the level of requisitioning had led to difficulties in the concreteness of the results while respecting the planned budget of time and expense. In addition, it was observed that the pandemic situation has affected not only the working practices, but especially the quality of communication. The paper concludes, therefore, with an attempt to resolve the critical methodological and communication criticalities, proposing the improvements to the approach on certain project phases in order to further strengthen Accenture's Waterfall-Agile hybrid method.

5.1 Benefits

The main benefit of this master's thesis paper was to revisit some phases of Accenture's project methodology in order to propose a solution to the critical issues encountered. This research can then be used as a basis for devising the structure of such stages and enable the project team to reproduce these suggestions appropriately.

The benefits of this research were not only evident in the improvement of the methodological steps offered at the conclusion of the analysis, but also in the relationship with the client. In fact, this work was useful at the end of the project to observe how the relationship with the client company could be improved in order to make the progression of future works more coordinated and fluid. From the knowledge of the difficulties and the possible improvements offered, the client company became even more aware of the Accenture methodology and this will allow the client to follow the progress of future projects even more effectively, without the risk of getting lost in the previous problems and misunderstandings. In other words, this work can be used by the client company as a framework to carefully and effectively monitor the progress of future projects.

5.2 Limitations

The limitations of this master's thesis paper are mainly related to the fact that the observation of a single project lasting a few months does not allow all the problems that may arise in its development to be identified. Consequently, the solution offered can only represent a small part of all the improvements that can be carried out.

A second limiting aspect of the research conducted relates to the fact that, on the one hand, the analysis was made by a student who had just entered the consultancy sector and also the world of work and, on the other hand, the project was analysed internally from a functional analyst level. Consequently, all the smallest aspects that a project manager would have identified in the course of his career and experience could not have been successfully captured and understood. On a different note, it has to be said that a student still in academia may have found connections and theoretical solutions during such research that are certainly effective and concrete. Further, this has also led to the consequence that the analysis carried out is purely qualitative in nature as it is based on project management theory and there is no numerical evidence of the results obtained.

Finally, a final aspect that may have limited the scope of this study, but also made it more peculiar, are the changes in the approach to work and to the project in more general terms, due to the global health situation. In fact, as described in the course of this paper, "prescribed consumption" project was carried out completely in a smart way, operating from one's own home, making this work certainly different from how it would have been done under normal conditions.

5.3 Future Steps

Future steps are certainly aimed at applying these findings and solutions in subsequent projects. As mentioned in the analysis section (4.2), the conclusion of the prescribed consumption project opened the door to new possibilities for improvement of the client company in the whole credit context. This means, as far as the Accenture team is concerned, moving forward with new activities by following the findings of this work. That is, avoiding to leave too much room for change in requirements during the last stages of the project and trying to make all the agile meetings necessary for advancement more structured and effective. On the other hand, as far as the client company is concerned, new opportunities mean a greater understanding of project methodology and a willingness to support the exact realisation of each phase. Future developments may relate to studying and experimenting with new solutions to modify the Accenture methodological approach and make it even more effective. The idea could be to make the approach even more hybrid, considering aspects from the Waterfall and Agile variants, such as the use of the PRINCE2 framework in order to make the project documentation more defined and avoid the risk of unfixed requirements. The possibilities in this scenario are innumerable, so care must be taken not to excessively modify an approach that is currently effective and leads to very positive results, as could be observed from the completion of the prescribed consumption project.

References

[1] Nicholas G Hall. «Project management: Recent developments and research opportunities». In: *Journal of Systems Science and Systems Engineering* 21.2 (2012), pp. 129–143 (cit. on p. 3).

[2] P.M. Institute. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)–Sixth Edition*. PMBOK® Guide. Project Management Institute, 2017 (cit. on pp. 3–5).

[3] Shiva Cohen Kashi, Shai Rozenes, and Irad Ben-Gal. «Project Management Monitoring Based on Expected Duration Entropy». In: *Entropy* 22.8 (2020), p. 905 (cit. on p. 3).

[4] J. Turner. «The Handbook of Project-Based Management». In: 92 (Jan. 2014) (cit. on p. 4).

[5] Bo Chen and Nicholas G. Hall. «Incentive schemes for resolving Parkinson's Law in project management». In: *European Journal of Operational Research* 288.2 (2021), pp. 666–681 (cit. on p. 4).

[6] Mounir Ajam. *Project management beyond waterfall and agile*. CRC Press, 2018 (cit. on pp. 5, 6, 9).

[7] Project Management Institute. *Agile practice guide*. Project Management Institute, 2017 (cit. on pp. 7, 19).

[8] ISO. *Guidance on project management*. ISO 21500:2021(E). Geneva, Switzerland: International Organization for Standardization, 2012 (cit. on p. 7).

[9] International Project Management Association. *Individual Competence Baseline: For Project, Programme & Portfolio Management*. IPMA Global Standard. Publications. International Project Management Association (IPMA), 2015 (cit. on p. 8).

[10] Joel Carboni, Michael Young, Peter Milsom, and Monica Gonzalez. *The GPM Global P5 Standard for Sustainability in Project Management v2*. Sept. 2019 (cit. on p. 8).

[11] Winston W Royce. «Managing the development of large software systems: concepts and techniques». In: *Proceedings of the 9th international conference on Software Engineering.* 1987, pp. 328–338 (cit. on p. 9).

[13] H. Bidgoli. *Encyclopedia of Information Systems: K-R.* Encyclopedia of Information Systems. Academic Press, 2003 (cit. on p. 10).

[14] Oriana Karina Eason. «Information Systems Development Methodologies Transitions: An Analysis of Waterfall to Agile Methodology». In: (2016) (cit. on pp. 10, 25, 27, 29).

[15] Mitra Pedersen. «A quantitative examination of critical success factors comparing agile and waterfall project management methodologies». PhD thesis. Capella University, 2013 (cit. on pp. 10, 15, 16, 19).

[18] Nripesh Nrip and Gunjan Behl. «A STUDY OF AGILE SOFTWARE DEVELOPMENT MODEL». In: I (Mar. 2012), p. 195 (cit. on p. 17).

[19] Kent Beck et al. *Manifesto for Agile Software Development.* 2001. URL: `http://www.agilemanifesto.org/` (cit. on p. 17).

[20] Arun Anson. «Agile Software Development». In: (June 2010), p. 49 (cit. on p. 17).

[23] Wilfred Van Casteren. «The Waterfall Model and the Agile Methodologies: A comparison by project characteristics». In: *Research Gate, February* (2017) (cit. on pp. 19, 28, 29).

[28] Sheetal Sharma, Darothi Sarkar, and Divya Gupta. «Agile processes and methodologies: A conceptual study». In: *International journal on computer science and Engineering* 4.5 (2012), p. 892 (cit. on pp. 23, 24).

[29] B. Unhelkar. *The Art of Agile Practice: A Composite Approach for Projects and Organizations.* Advanced & Emerging Communications Technologies. CRC Press, 2013 (cit. on p. 25).

[30] Fernando Almeida. «Challenges in Migration from Waterfall to Agile Environments». In: *World Journal of Computer Application and Technology* 5 (Sept. 2017), pp. 39–49. DOI: `10.13189/wjcat.2017.050302` (cit. on pp. 31–34).

[31] Kalpana Sureshchandra and Jagadish Shrinivasavadhani. «Moving from waterfall to agile». In: *Agile 2008 conference.* IEEE. 2008, pp. 97–101 (cit. on p. 31).

[42] Shouzo Hori, Takako Nakatani, N Ubayashi, K Katamine, and Masaaki Hashimoto. «Project management patterns to prevent schedule delay caused by requirements changes». In: *Proc. ICSOFT* (2009), pp. 115–120 (cit. on p. 79).

[43] Sharon McGee and Des Greer. «Towards an understanding of the causes and effects of software requirements change: Two case studies». In: *Requirements Engineering* 17 (June 2012) (cit. on p. 80).

[44] Sharon McGee and Des Greer. «Software requirements change taxonomy: Evaluation by case study». In: Oct. 2011, pp. 25–34 (cit. on p. 80).

[45] D. Pandey, U. Suman, and A. K. Ramani. «An Effective Requirement Engineering Process Model for Software Development and Requirements Management». In: *2010 International Conference on Advances in Recent Technologies in Communication and Computing*. 2010, pp. 287–291 (cit. on p. 80).

Websites List

[12] *Waterfall Model: The Ultimate Guide to Waterfall Methodology.* Available on line. 2020. URL: https://www.projectmanager.com/waterfall-methodology (cit. on pp. 9, 12, 13).

[16] *SDLC - Waterfall Model.* URL: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm (cit. on pp. 11, 13).

[17] Andrew Burak. *Your Guide to Writing a Software Requirements Specification (SRS) Document.* URL: https://relevant.software/blog/software-requirements-specification-srs-document/ (cit. on p. 12).

[21] *What is the Agile Manifesto?: Agile Principles and Values.* Dec. 2020. URL: https://www.productplan.com/glossary/agile-manifesto/ (cit. on p. 17).

[22] *What is Agile and Why is it Important?* July 2020. URL: https://www.business2community.com/strategy/what-is-agile-and-why-is-it-important-02329437 (cit. on p. 18).

[24] Olha Anurina. *Agile SDLC: Skyrocketing Your Project with Agile Principles.* Jan. 2021. URL: https://mlsdev.com/blog/agile-sdlc (cit. on p. 20).

[25] Ihor Feoktistov. *Agile Software Development Lifecycle Phases Explained.* Nov. 2020. URL: https://relevant.software/blog/agile-software-development-lifecycle-phases-explained/ (cit. on pp. 20, 21).

[26] SCRUM alliance. *Overview: What is Scrum?* Available online. Dec. 2020. URL: https://greenprojectmanagement.org/prism-methodology (cit. on p. 22).

[27] *What is Scrum?* URL: https://www.scrum.org/resources/what-is-scrum (cit. on p. 23).

[32] Mightybytes. *Transitioning from Waterfall to Agile Methods.* Available on line. 2020. URL: https://www.projectmanager.com/waterfall-methodology (cit. on p. 32).

[33] Ester Cohen. *The Definitive Guide to Project Management Methodologies.* Available online. July 2019. URL: https://www.workamajig.com/blog/project-management-methodologies (cit. on p. 34).

[34] Green Project Management. *GPM Driving Sustainable Change - PRiSM Methodology.* Available online. Dec. 2020. URL: https://greenprojectmanagement.org/prism-methodology (cit. on p. 35).

[35] *Prism Methodology.* URL: https://greenprojectmanagement.org/prism-methodology (cit. on p. 36).

[36] PRINCE2. *PRINCE2 Processes.* Available online. Dec. 2020. URL: https://www.prince2.com/eur/prince2-processes (cit. on pp. 36, 37).

[37] Consulting Interview Training. *Consulting Industry Overview.* 2013. URL: http://www.streetofwalls.com/finance-training-courses/consulting-interview-training/consulting-industry-overview/ (cit. on p. 39).

[38] Consulting.us. *Consulting Industry.* URL: https://www.consulting.us/consulting-industry (cit. on p. 39).

[39] *Accenture.* URL: https://www.accenture.com/it-it (cit. on pp. 40, 42).

[40] Essay Examples. *CAccenture: History.* URL: https://benjaminbarber.org/accenture-history/ (cit. on p. 41).

[41] *Prescription.* URL: https://legal-dictionary.thefreedictionary.com/prescription (cit. on p. 59).