QBAi®

BUSINESS ANALYST CERTIFICATION

HANDBOOK FUNDAMENTAL LEVEL

September 2020 - release 2020 v.1

REVISION HISTORY – Fundamental Level

Date	Version	Note
September 2020	1.0	First release – Fundamental Level

TRANSLATION HISTORY – Fundamental Level

Date	Change	Note
January 2022	First release	

Contents

I Section 1 **14**

1 Fundamentals of business organization and management **15**

1.1 The Organization 15

1.2 Performance Evaluation 17

1.3 The enterprise and the types of activity 18

1.4 Types of Organizational Structures (O.S.) 20

1.5 The organizational chart and evolution of the organizational structure . . 32

2 Business Management Systems **36**

2.1 The Enterprise as a System 36

2.2 Planning and coordination mechanisms 38

2.3 Business Management Systems 40

2.4 Control management system and control cycle 42

2.5 Regulatory Systems 43

2.6 Cybernetic Systems 44

2.7 Generative Systems 46

3 Quality Systems **49**

3.1 The Quality Management Systems 50

3.2 The modern root of quality 51

3.3 Deming Wheel 55

3.4 Total Quality Management 57

3.5 Process management (BPR and BPI) 59
3.6 Business Process Reengineering. Methodology and phases. 60
3.7 Business Process Improvement. Methodology and phases. 62
3.8 Business Process Management. Methodology and phases 63

4 Project implementation methodologies 67

4.1 Definitions (Process - Organization - Project) 67
4.2 Project and Process 68
4.3 Project and Project Management 69
4.4 Project life cycle 70
4.5 Components of the project 72
4.6 The enhancement of the project 74
4.7 Planning Tools - Large projects 75
4.7.1 WBS – Work Breakdown Structure 75
4.7.2 OBS - Organizational Breakdown Structure 78
4.7.3 RAM – Responsability Assignments Matrix 80
4.7.4 GANTT – Henry Lawrence Gantt's chart 82
4.7.5 CBS - Cost Breakdown Structure 84
4.7.6 RBS - Resource Breakdown Structure 86

II Section 2 88

5 Project Management 89

5.1 Inter-functional communication 90
5.2 Project Management Methods 91
5.2.1 Waterfall 91
5.2.2 Agile 92
5.2.3 Scrum 93
5.2.4 Hybrid 95
5.2.5 Critical Path Method (CPM) 96

5.2.6 Six Sigma 97
5.2.7 DevOps 98

6 Risk and probability of occurrence 100

7 Risk Management 102
7.1 Risk management tools. The Fishbone Diagram 104

8 The project roles 106
8.1 The sponsor 107
8.2 The stakeholders 108
8.3 The Program Board 110
8.4 The Project Manager – PM and the Project Initial Document (PID) . . . 110
8.5 The Project Management Office (PMO) and the Project Status Report (PSR) 112
8.6 The Quality Manager 114
8.6.1 Quality plan 116
8.7 Head of administration 119
8.8 External consultants and suppliers 120
8.9 The Team 121

9 The phases of the project team 122
9.1 Forming or phase of formation 122
9.2 Storming or phase of conflict 123
9.3 Norming or cohesion phase 124
9.4 Performing or stage of performance 124
9.5 Adjourning or suspension phase 125

10 ICT projects 126
10.1 The method 127
10.1.1 TimeBoxing and Agile method in ICT 128

10.1.2 Waterfall or traditional method in ICT 129

11 The team of an ICT project 131

11.1 The team leader 133

11.2 The designer 134

11.3 The analyst 134

11.4 Technical specialist or technical analyst 136

11.5 The Tester 136

11.6 Professional sub-levels 137

12 Functional analysis 138

12.1 Objective of the functional analysis document 138

12.2 Analysis activities 138

12.3 The minutes 139

12.4 Recap 140

12.5 Requirements engineering 141

12.5.1 Requirements engineering objective 141

12.5.2 Definition of requirement 141

12.5.3 Classification, characteristics, and collection of requirements 143

12.6 Requirements analysis 144

12.7 The tools of the functional analyst 145

12.8 Feasibility Analysis and the Feasibility Specification Document 146

12.8.1 Business processes 148

12.8.2 BPMN- Business Process Modeling Notation 149

12.8.3 BPD - Business Process Diagram 151

12.8.4 The standard graphic elements of a BPD 151

12.9 Graphs and diagrams 154

12.9.1 Flow Chart 154

12.9.2 UML Diagram 155

12.9.3 Class Diagram 159

12.9.4 Use case diagram . . . 161
12.9.5 Scenario . . . 164
12.9.6 Activity Diagram . . . 165
12.9.7 Decisions . . . 166
12.9.8 Competing Paths . . . 168
12.9.9 Signals . . . 168
12.9.10 Swimlanes . . . 168
12.9.11 Testing principles . . . 181
12.9.12 Main types of tests . . . 182
12.9.13 Test Process . . . 184
12.9.14 Test plan . . . 187

13 The functional analyst in the ICT field - Conclusions 190
13.1 Phases . . . 191
13.2 The tools . . . 192

III Section 3 194

14 Project 1 196

15 Project 2 204

16 Project 3 208

Introduction

The Quality Business Analyst Institute (QBAI®) promotes the Business Analysis certification among companies, universities, and other organizations, defining the elements of knowledge, skills, and competences, as well as their continuous update. QBAI®'s aim is to consolidate the profession of the Business Analyst, identifying and qualifying specific training processes useful to reach different levels of knowledge in the field of functional analysis.

QBAI®provides three specialization levels for which different missions are established according to the specialization that applicants want to obtain.

Level I – Fundamental of Business Analysis
Aimed at applicants with a basic level of specialization (without any work experience)

Level II – Advanced Business Analyst
It is aimed at professionals with a medium level of functional specialization (with 3 to 5 years of work experience). The testing module provides the basic notions for drafting test plans. Collector flow of the several areas: project implementation methods.

Level III – Master – Strategic Business Analyst Management
It is aimed at professionals with a high level of functional specialization (with more than five years of work experience). Implementation of real projects related to technological innovation through the use of the Agile method and the management of project teams.

Purpose of this document

This document concerns all the training aspects related to the first level of the QBAI®certification, defined ”Fundamental Level”.

The Handbook provides contents for the Training Providers who decide to become "QBAI®accredited Trainer Providers" and to contribute to the drafting of the Association's training materials. Every part of this document shall be included in the training materials.

The Handbook is necessary for the preparation for QBAI®'s Fundamental Level certification test and represents the only official document on which the certification test is based (and that a candidate can use to get prepared either through an accredited course or independently).

All the areas included in this document are relevant to prepare for the test and to achieve the certificate through a qualified Test Provider. This document can be used by the international software engineering community as the basis for articles, publications, or books as long as the source is specified.

Accreditation process

All entities wishing to use this material must be recognized and accredited by QBAI®. Accreditation is granted after careful evaluation by a Council of experts appointed by QBAI®., which will verify the documentation provided by the training entity applying for the accreditation. All materials produced by the accredited entities are to be evaluated by some Business Analyst experts before they can become test materials. The accredited course complies with the general notions of QBAI®. Handbook, as specified in the QBAI-STAND/CERT234-1:2021 standard.
QBAI®Association follows the rules defined by the QBAI-STAND/CERT234-1:2021 standard. Therefore, the accredited independent QBAI®certification bodies are authorized to manage the tests thanks to the Qualified QBAI®Test Providers.

Certification Standard

The Handbook ”Fundamental Level” has been developed according to QBAI®'s international standard, recognized by the accreditation procedure obtained through the certification body. Such standard complies with the QBAI-STAND/CERT234-1:2021 requirements. The certification levels of QBAI®are organized in a matrix structure as described also on the official website www.qbai.org.

The Standard is based on:

1. sections: the topics covered in the various QBAI®'s certification levels;
2. prerequisites: the minimum requirements needed to access each specific QBAI®'s certification levels;
3. tests: the candidate is evaluated according to a test that meets the requirements of the standard.

Sections

Each certification level is divided into three sections:

- Section 1st: introduction to the topic, terminology, and fundamental notions
- Section 2nd: presentation and use of tools related to the area of interest
- Section 3rd: training. Examples are given based on real cases

PREREQUISITES

For each level of QBAI®certification there are some specific prerequisites. For the “Fundamental” level:

a) bachelor's degree in science/technology/humanities studies (or alternatively)

b) a minimum of 6 months of work experience in IT sectors

A “Fundamental Level” are a mandatory prerequisites for “Advanced” and “Master” level.

TEST

The tests for QBAI®'s "Fundamental" level are designed according to the QBAI®standard, in compliance with the QBAI-STAND/CERT234-1:2021 standard. QBAI®Fundamental of Business Analysis certification test is based on this Handbook. The Fundamental test is a prerequisite for the Advanced and Master levels. All sections of this document can be part of the QBAI®Fundamental of Business Analysis certification test.

Attendance to an accredited course is not mandatory in order to obtain the QBAI® "Fundamental level" certification.
The certification can be achieved only by passing the QBAI®Fundamental of Business Analysis test.
The test's calendars are published on the official QBAI®website www.qbai.org.

Application

The QBAI®Fundamental of Business Analysis certification is applicable to all professionals involved in the European competence framework, as defined by the UNI EN 16234-1:2020 standard.

This certification represents the first QBAI®certification level called 'Fundamental' according to the structure defined by the QBAI-STAND/CERT234-1:2021 standard. This certification is recommended to anyone who wants to certify an understanding of the fundamentals of Business Analysis, getting to learn terms, tools, and operating methodologies of this area of expertise.

Anyone who obtains the QBAI®Fundamental certificate can achieve higher levels of certifications in the field of Business Analysis.

Knowledge levels and objectives of QBAI®Fundamental of Business Analysis Tests

The knowledge of the contents of this Handbook are tested for granting the QBAI®Fundamental of Business Analysis certificate.
The evaluation model for the candidates who take the QBAI®certification test follows the principle described by Anderson, Lorin W., Krathwohl, David R., etc. “A taxonomy for learning, teaching, and assessing: A revision of Bloom’s taxonomy of educational objectives.” Allyn and Bacon. ISBN 978-0-8013-1903-7.

According to this model, the first three evaluation steps of the pyramid are used for the Fundamental level:

Processes	**Evaluation**
Remembering	The learner can recognize and recall relevant knowledge for long-term memory
Understanding	The learner can construct meaning from oral, written, and graphic messages: interpret, exemplify, classify, summarize, infer, compare, explain, paraphrase, discuss
Applying	The learner can use information in a new way: demonstrate, dramatize, interpret, solve, use, illustrate, convert, discovery, discuss, prepare

The test is structured using concepts contained in one or more paragraphs of the Handbook. The Fundamental test usually includes a set of **50 multiple-choice questions to be answered in the limited time of 60 minutes.** For each question within the set, the number of requested answers is indicated. The QBAI®Fundamental Certificate has no expiration. The certificate will indicate the latest version of the Handbook available at the time in which the test was taken and the test date.

How this Handbook is organized

The Handbook structure complies with the QBAI-STAND/CERT234-1:2021 standard, according to which the manual is divided into three sections:

- introduction to the subject, terminology, and fundamental notions;
- presentation and use of tools related to the area of interest;
- training. Exercises based on real cases.

At least 18 hours of lessons of a course carried out by a qualified QBAI®Trainer Provider are required to correctly acquire all the concepts expressed in this Handbook, needed for the preparation for the QBAI®Fundamental certification test.

Part I

Section 1

Chapter 1

Fundamentals of business organization and management

1.1 The Organization

In order to correctly operate in the context in which the Business Analyst is called to work, it is necessary first of all that he/she recognizes the type of organizational structure that he/she has been placed in. Business organizations differ from each other, however they are often characterized by common key elements. Knowledge of the classical definitions of organization provided by the literature is a good starting point in order to recognize the main types of organization.

According to the academic of management sciences Henry **Minzberg** (1983) what really identifies an organization is ≪the complex of modalities according to which the division of labor into distinct tasks is carried out and how coordination between these tasks is achieved≫; while for the psychologist Edgar **Schein** (1985) ≪*An organization is the rational coordination of the activities carried out by a number of people for the achievement of some common explicit purpose or goal, through division of labour and functions, and through a hierarchy of authority and responsibility*≫. For sociologist Howard **Aldrich** (1979): ≪*The role that the individual employees of the company must play and the re-*

lationships that must exist between them≫ is decisive for the purpose of identifying a company organization, so that the coordination of their work ensures an optimal contribution for the achievement of corporate objectives. Having said that, we can try to give a comprehensive but not exhaustive definition of the concepts expressed by theorists, according to which an organization is a **planned set of people, activities, and processes** aimed at achieving a common goal, through the principle of labor and functions' division, as well as through a hierarchy of authority and responsibility.

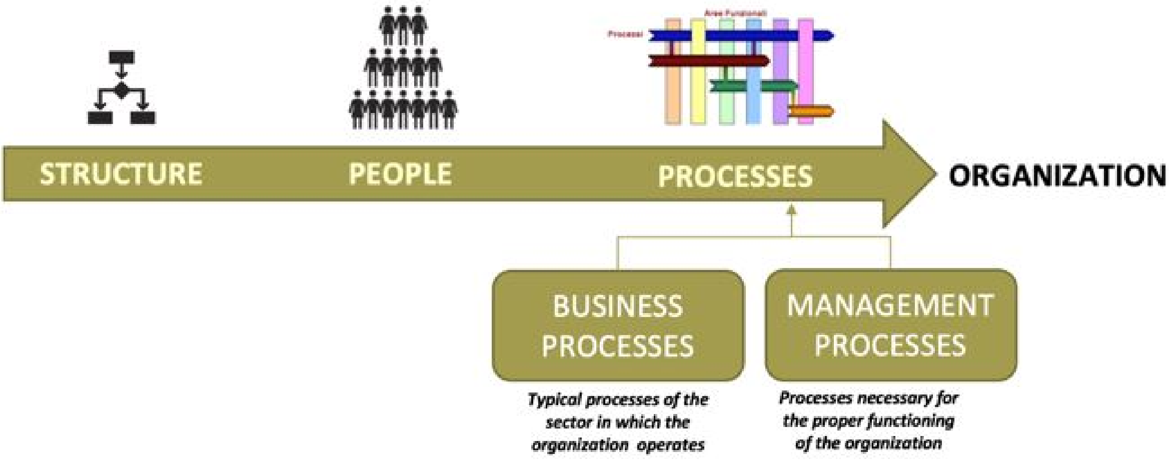

Aree Funzionali
Processi
STRUCTURE
PEOPLE
PROCESSES
ORGANIZATION
BUSINESS PROCESSES
Typical processes of the sector in which the organization operates
MANAGEMENT PROCESSES
Processes necessary for the proper functioning of the organization

Figure 1.1: Structures - peoples - processes

Given that a unique definition of organization does not exist, it is possible nonetheless to identify the common terms belonging to it, as they are the key elements that compose and characterize it.

The *organizational vocabulary*, containing some of the most useful terms, can be formulated as follows:

- **Organizational structure:** is the result of the implementation of criterias for labor division and tasks' aggregation into organizational units.
- **Organizational unit/people:** is a system (composed of people and activities) that carries out activities aimed at achieving the business objectives. People and

activities that contribute to the achievement of such goals are grouped and linked to each other in order to work efficiently and effectively.

- **Processes:** is a set of activities developed within an organization, connected to each other, that are aimed at achieving a shared goal.
- **Operational mechanisms:** are rules and procedures that link and coordinate the activities of the several organizational units.
- **Sector/industry:** the context and the environment in which the organization operates.

1.2 Performance Evaluation

Regardless of the type of organization that you may want to scientifically identify and adopt for a company, it is necessary that the chosen organizational model is tested so that performances and fitness for purpose can be measured. The performance parameters that we are about to describe are also useful for measuring the health of a pre-existing organization.

1. **Efficiency:** measures the ability to use some resources (the organization's productivity);
2. **Effectiveness:** measures the ability to achieve the planned goals;
3. **Flexibility:** it expresses the time needed to switch from the current output to a new output.

Parameters related to efficiency, effectiveness, and flexibility can be used to evaluate the performances of any type of organization.

1.3 The enterprise and the types of activity

For a deeper understanding, it is now advisable to introduce a definition of enterprise in our *organizational vocabulary* and at the same time identify the possible types of activity. An enterprise generally corresponds to the ”complex of goods and people organized for the production (supply) of certain goods and services (the amount of which is determined in relation to the demand that the market will make) with the aim of achieving specific economic objectives” (Antoldi F., 2003, *Conoscere l'Impresa*). However, an enterprise can also be identified as a system that dynamically assumes a specific connotation according to the activities that it incorporates within its own boundaries. This is what economist Michael Porter (1985) theorized with the elaboration of the ”Value-chain”.

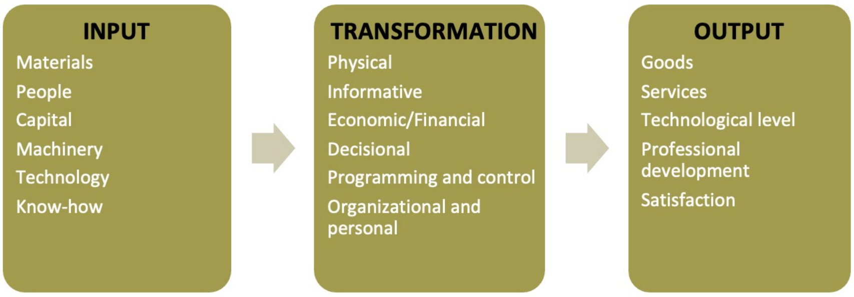


Figure 1.2: Processes

Such theory identifies two types of fundamental activities (processes and functions) within the company which can generate economically appreciable value: primary activities and support activities (that are valid for a manufacturing or a processing company). The ”company system” as a whole will encompass all those activities that provide a technical-economic convenience for the organization. The **primary activities** are those that directly contribute to the creation of the organization's output (products or services):

1. Inbound logistics - include all the activities for managing the flow of material goods within the organization;

2. Operational activities - production of goods and/or services;

3. Outbound logistics - include the activities for managing the flow of material goods outgoing from the organization;

4. Marketing and sales - promotion of the product or service in the markets and management of the sales' process;

5. Customer care and services - all the customer-support after-sales activities.

The **support activities** do not directly contribute to the creation of the output but are equally necessary for it to be fully defined a "product":

1. Procurement - all the activities related to the purchase of the resources necessary for the production of the output and for the functioning of the organization;

2. Human Resources - research, selection, recruitment, training, update of the personnel's compensation, and trade unions' negotiations;

3. Research and development of technologies – R&D) - all the activities aimed at improving products or processes;

4. Infrastructural Activities - planning, accounting, organization, IT, legal affairs, general management.

Michael Porter classifies the activities of companies in 2 macro categories:

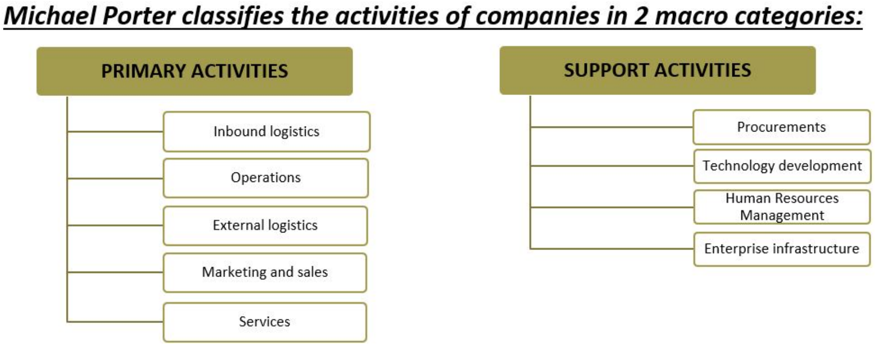


Figure 1.3: Primary and Support Activities

The difference between the value (price) of the produced good or service and the cost of the activities (primary and support) per unit of product, is the margin (profit).
The profit is obtainable when the activities are carried out in accordance with the strategy and the objectives of the company. To make this happen, in practice, the primary and support activities coincide with the activities carried out by the managerial functions of the company.
Such corporate functions are embodied in specific organizational units (each chaired by a responsible manager) accountable for carrying out specific tasks and equipped with employees that have specialist skills in specific functional areas.

1.4 Types of Organizational Structures (O.S.)

In order for a company to actually generate profit it is necessary that the activities (primary and support) coordinated by the corporate functions are organized and directed in concert for the execution of the corporate strategies decided by the owners and/or the management.
The choice, taken by the management, of the most suitable and performing organizational structure based on the company's purpose affects the organization and coordination of the corporate functions.
The key elements that characterize the organizational structure of a company (or of any other entity) are:

- the type of hierarchy that is formed within the organization;
- the quality and quantity of the functional units;
- the type of internal communication system.

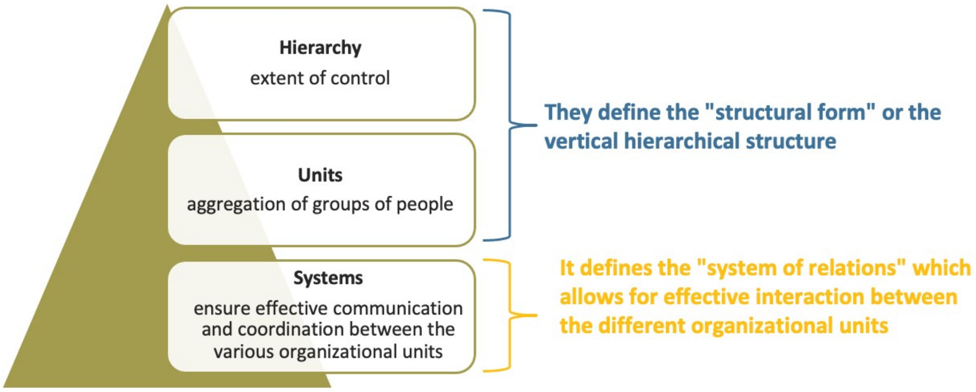
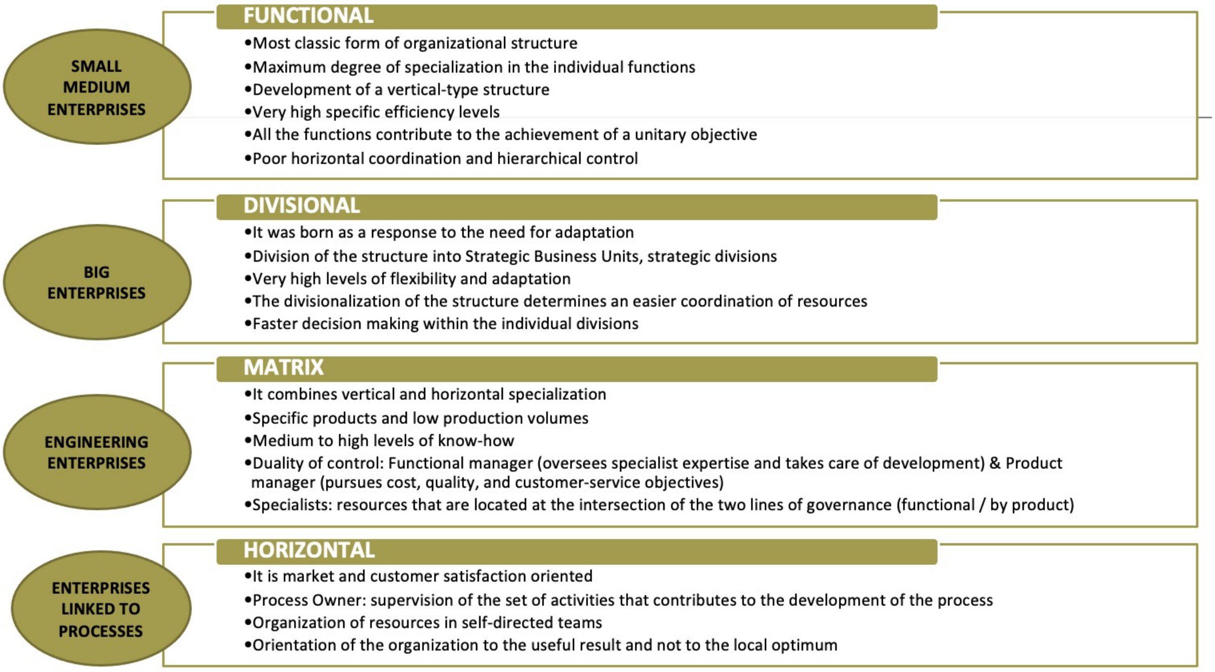

Hierarchy
extent of control
Units
aggregation of groups of people
Systems
ensure effective communication and coordination between the various organizational units
They define the "structural form" or the vertical hierarchical structure
It defines the "system of relations" which allows for effective interaction between the different organizational units

Figure 1.4: Hierarchy - Units - Systems

The first two factors of the list define the *framework* (or structural form) of the organization, while the third factor is a dynamic element that enables an effective interaction between the several units within the organization, and that defines the system of relations applied to the *framework*.

Both studies and practice have identified four major combinations of choices that lead to the creation of an organizational structure and that are mostly used for organizations with large dimensions. We can generally identify 4 types of organizational structures (Mintzberg, H. (1979)):

1. **functional:** it is the most classic organizational chart structure;
2. **divisional:** it is the most flexible structure;
3. **matrix:** it is a type of structure that allows scarce resources to be used in the most effective way;
4. **horizontal:** it is the most customer satisfaction oriented structure.



SMALL MEDIUM ENTERPRISES

FUNCTIONAL

- Most classic form of organizational structure
- Maximum degree of specialization in the individual functions
- Development of a vertical-type structure
- Very high specific efficiency levels
- All the functions contribute to the achievement of a unitary objective
- Poor horizontal coordination and hierarchical control

BIG ENTERPRISES

DIVISIONAL

- It was born as a response to the need for adaptation
- Division of the structure into Strategic Business Units, strategic divisions
- Very high levels of flexibility and adaptation
- The divisionalization of the structure determines an easier coordination of resources
- Faster decision making within the individual divisions

ENGINEERING ENTERPRISES

MATRIX

- It combines vertical and horizontal specialization
- Specific products and low production volumes
- Medium to high levels of know-how
- Duality of control: Functional manager (oversees specialist expertise and takes care of development) & Product manager (pursues cost, quality, and customer-service objectives)
- Specialists: resources that are located at the intersection of the two lines of governance (functional / by product)

ENTERPRISES LINKED TO PROCESSES

HORIZONTAL

- It is market and customer satisfaction oriented
- Process Owner: supervision of the set of activities that contributes to the development of the process
- Organization of resources in self-directed teams
- Orientation of the organization to the useful result and not to the local optimum

Figure 1.5: Organizational Structures

In the **Functional** O.S., the activities of the value chain are grouped according to a criterion of specialist expertise.

Graphically, this O.S. represents the typical vertical form with the sub-functions descending from the main functions that are placed at the top and aggregating personnel with the same skills (e.g. plant management →production departments).

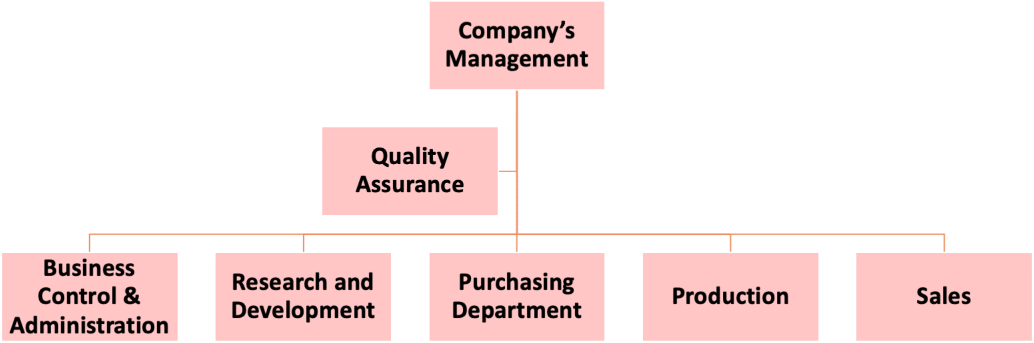


Figure 1.6: Example Functional O.S.

This type of structure is typical to some production plants where high process' technologies are required (e.g. heavy or precision engineering industry) where horizontal coordination is not relevant to achieve the "product objective" (e.g. due to lack of need for information's exchange between different production departments) but the entire control of the production line is entrusted to vertical connections (hierarchical control). The individual functions grouped by specialty (departments) allow for a quick staff training and the development of serial productions (producing products according to economies of scale).

The major weakness of this type of O.S. consists in high complexity for business transformations in the event of crises or market changes.

In such scenarios a business transformation is achievable only through extremely detailed and long-running industrial conversion plans.

STRENGTHS	WEAKNESSES
It favors objectives of efficiency, reduction of structural costs, economies of scale	Weighting of coordination and a tendency to bureaucratization
Specialized employee training	Resistance to product/market diversification; innovation is not favored
Centralized control	Integrated managerial skills are not developed
Centralization of profit responsibilities	Difficulties in managing environmental variables

STRENGTHS	WEAKNESSES
Cost control of functions	
Quick management of operational exceptions of the communication and decision-making system via the hierarchy	
Good operational flexibility	
Good possibilities for dimensional development	
People/units oriented in a diversified way, according to the specific characteristics of the different functions	

Table 1.1: Functional structure strengths/weaknesses

The **Divisional**O.S. arises from the need to grant autonomy to some company functions in a coherent division, separating them from the less coherent ones. In this O.S., each division contains all the necessary resources and functions within it to support a certain product line or geographical area.

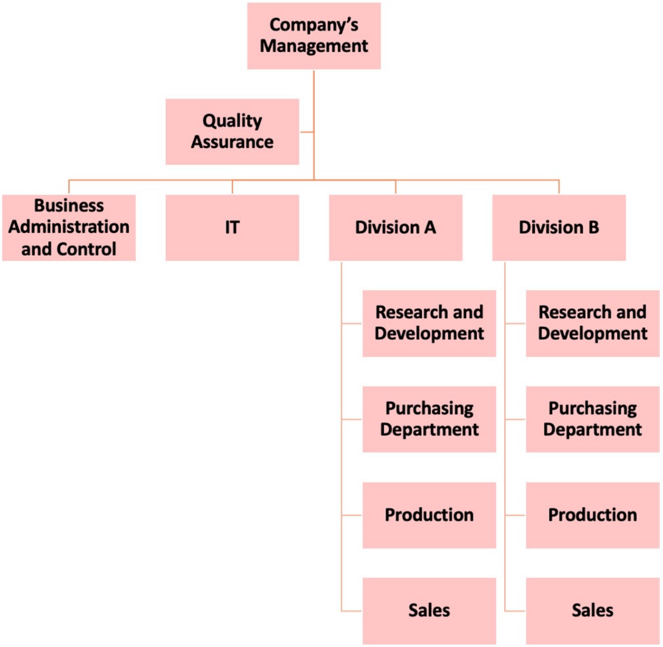


Figure 1.7: Example Divisional O.S.

An effective example can be represented by the need of a company with different product lines (e.g. clothing, objects for the house, perfumes, etc.) to divide each function by type of production, creating new, autonomous, and flexible groups of functions, called divisions. Another example is the case of an international company where a divisional O.S. allows for greater flexibility and fast adaptation to the needs of new markets.
In a divisional O.S., the coordination of the several units (divisions) of the company is easier and faster, however it becomes more difficult to implement global policies involving the company as a whole. In fact, conflicts may arise between the objectives of the individual divisions and those of the upper management responsible for the performance of the entire company.

STRENGTHS	WEAKNESSES
Dimensional development by aggregation of units and through the growth and diversification of products	In the case of significant overall dimensions, each single division can be transformed into a functional structure, with its specific problems; possible decision overload for management and coordination
Decision-making and behavioral autonomy of the individual units, adapted to their respective reference markets	Possible conflicts between the need for innovation and stability in the divisions
Attention and use of management on individual products	Possible difficulties of coordination between global objectives with divisional objectives
Decentralization of profit responsibilities	Duplication of specialist resources, distributed in the various divisions
Training and development of managers with global management skills. Possible lack of managerial skills	
Operational flexibility and management flexibility and product improvement, even in multi-product situations	
Strengthening and development of planning and management control tools	

Table 1.2: Divisional structure strengths/weaknesses

The **“a Matrix”**O.S. is typical for small or medium-sized companies characterized by a high level of specialization (e.g. technical firms, consulting firms, engineering firms, etc.). This model can be represented as a diagram (or matrix) with the top roles of the functional model (on the X axis - Function Managers: Marketing, Design, Controller, etc.) having to choose a Product Manager (on the Y axis) most suitable for the objective to be achieved in the short term (the Project).

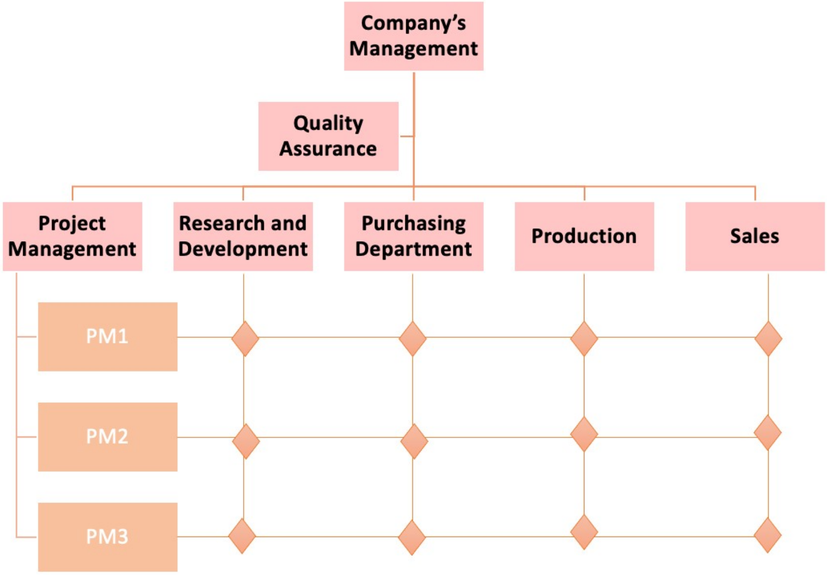


Figure 1.8: Example Matrix O.S.

This structure allows a wide interchangeability of resources within the company, trying to maximise the use of skills and know-how already possessed (and so reducing the time needed for study and development) for a rapid completion of the project (project = product of the consulting firm). The advantage linked to the speed with which high-profile performances can be obtained using this O.S. gets, however, often held back by a greater complexity of coordination to be addressed by the top management and can lead to significant decision-making overloads.

STRENGTHS	WEAKNESSES
Flexibility	Double command and therefore double dependence
Interfunctionality	Potentially conflicting and changing situations
It favors autonomy and professionalism in employees	Partially inefficient structure that can lead to strong decision-making and coordination overloads at the top
It orients all levels towards more coordination and cooperation rather than control	

STRENGTHS	WEAKNESSES
Tension towards goals	
Innovative thrust	

Table 1.3: Matrix structure strengths/weaknesses

Finally, the **Horizontal** O.S. introduces some cross-functional managerial units called Process Owners. They are responsible to ”oversee” the whole set of activities that contribute to the development of a process aimed at producing a certain good or service. Adopting a Process Owner, a company takes advantage of a cross-functional supervisor who has the ability to affect the production process, by leading all the necessary changes needed to improve its performance. In this O.S., the function managers are still accountable for managing the human resources, placing them in the most appropriate manner within the organization according to their specialist skills.

Resources on an operational level are organized into self-directed teams with some working groups having a wide access to a certain process and thus being accountable for all the process' activities. Such an approach facilitates a greater awareness of the performance objectives. A horizontal structure bases each production process on the customer-supplier model in order to steer the organization towards the needed result (the customer's request) ensuring the product's sale.

STRENGTHS	WEAKNESSES
Useful result-oriented organization	Notable training programs
Visibility of the process by individual actors	Potentially conflicting and changing situations
Process performance based on customer needs	

Table 1.4: Horizontal structure strengths/weaknesses

A S.O choise depends by three objectives tipologies:

1. **STRATEGIC GOALS**
   - expansion of products and markets
   - diversification towards new products and new markets
2. **DIRECTIONAL GOALS**
   - resource specialization level
   - acquisition and development of new resources
3. **OPERATIONAL GOAL**
   - methods of carrying out production activities

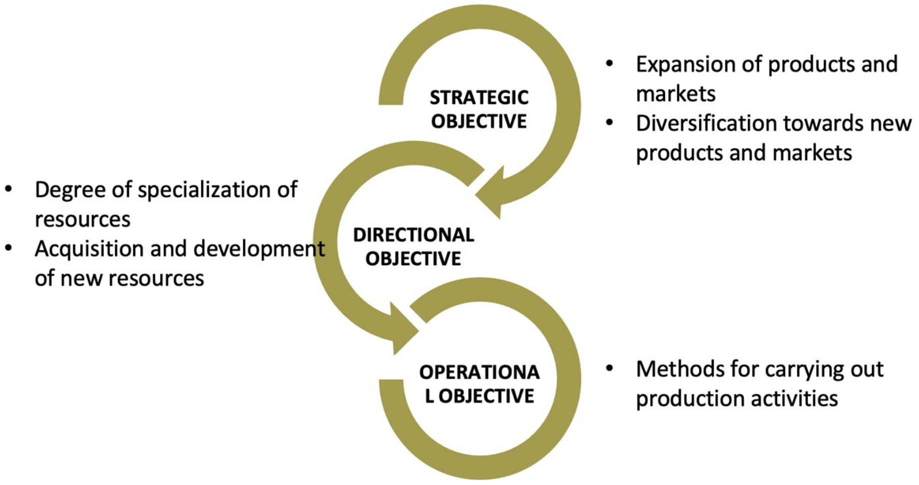

STRATEGIC OBJECTIVE
• Expansion of products and markets
• Diversification towards new products and markets
• Degree of specialization of resources
• Acquisition and development of new resources
DIRECTIONAL OBJECTIVE
OPERATIONA L OBJECTIVE
• Methods for carrying out production activities

Figure 1.9: Objectives Organizational Structure

Therefore, to identify the most consistent structure with the objectives of the organization, it is sufficient to ideally place oneself on a continuous line that has the two models as extremes "Limit": the 'vertical' and the 'horizontal' structure. The vertical structure is oriented towards obtaining specialization and efficiency objectives; control, achieved through vertical connections it ensures strong consistency with the objectives

of maximizing the use of specialist resources. A horizontal structure responds to the organization's need for flexibility and adaptation; this structure induces behaviors in the organization that are consistent with the development of resources (learning), with the search for innovative solutions (creativity) and with the sharing of information (involvement and participation).
The figure below indicates the relationship that binds the ”dominant” objectives of the organization with the structural options analyzed so far.

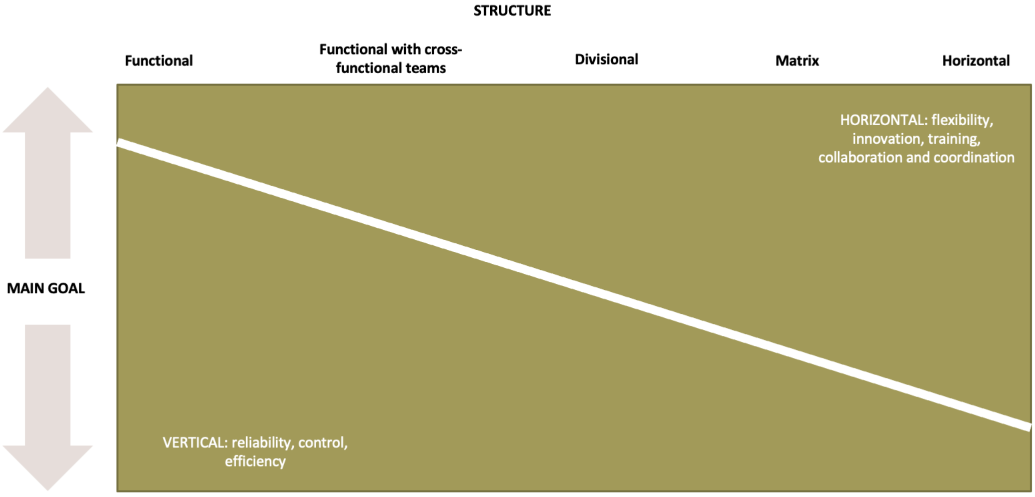

STRUCTURE
Functional
Functional with cross-functional teams
Divisional
Matrix
Horizontal
MAIN GOAL
HORIZONTAL: flexibility, innovation, training, collaboration and coordination
VERTICAL: reliability, control, efficiency

Figure 1.10: Organizational Structure - Objectives

Once the description of the different types of organization is completed, it is possible to identify the organizational structure that mostly fits the considered company. This selection process is mainly affected by the link between environment and structures. Scholars like Lawrence and Lorsch (1960) underlined the need to adapt the organizational structure to the environmental changes and elaborated two organizational models: the mechanistic organization and the organic organization. The two different paradigms reflect the environments in which they operate, the first being typical of stable environments, whereas the second being better suited to more competitive environments.

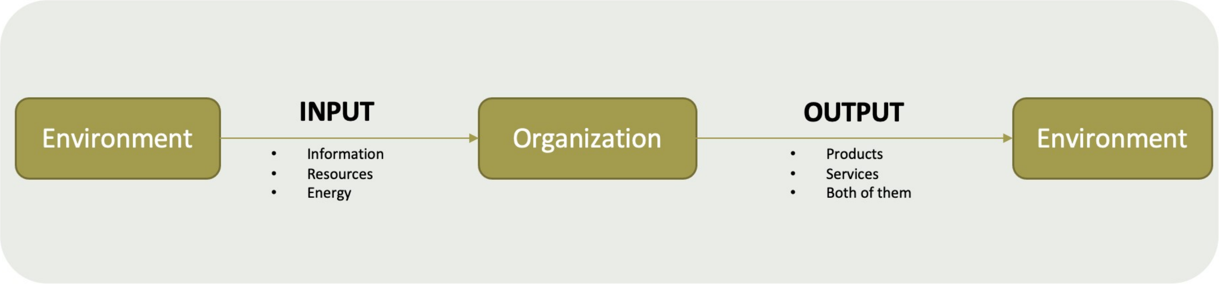


Figure 1.11: Organizational model

According to these theories, the organization has to be viewed as an ”open system” that receives inputs from the external environment, elaborates them into new outputs and inserts those outputs back into the external environment, closing the circle (G. Capaldo, E. Esposito, L. Iandoli, C. lo Storto, P. Rippa, 2015, *Introduction to the study of the Company and Organization*). These theories show the importance of the Structure-Environment relationship which helps to explain, as a corollary, a company's need to adapt over time its organizational structure to the environmental changes in order to effectively produce value.

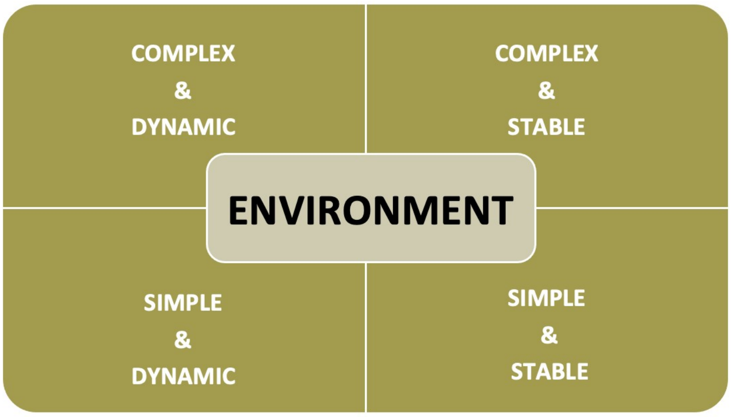


Figure 1.12: Organization - Environment

When the environment is uncertain, the organization responds to the organic model (which refers to the metaphor of an organism that simply tends to make all functions indispensable and equates them at a hierarchical level). In order to be productive, this type of organization needs to be flexible, acquiring data from the external environment, obtaining information about all the possible variables (e.g. market trends, competitor activities), proposing new products to counteract competitors, and activating a good corporate promotional function (like a Marketing office). The organic model often works on a project basis, with the company gradually setting up project teams dedicated to single activities. Operating in a stable environment, with only few competitors, it is common for companies to develop a rigid, highly bureaucratized organization, with a pyramid-type governance hierarchy. In such cases, organizations appear to frequently be mechanistic as these companies are not pushed to change by the external environment. The mechanistic model is mostly characterized by vertical communication processes that follow the chain of command.

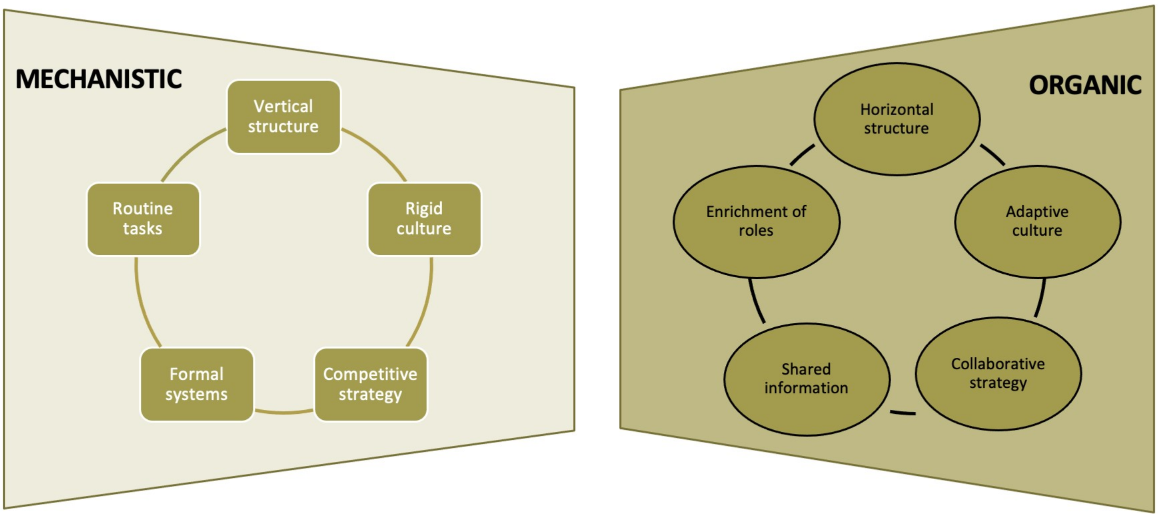


Figure 1.13: Mechanical and Organic System

1.5 The organizational chart and evolution of the organizational structure

Before illustrating the main internal factors that lead to the organizational change, it is advisable to focus on the chart (called "organizational graphic structure" or "organizational chart") that graphically shows the organizational structure of the company, indirectly indicating the type of model adopted by it.

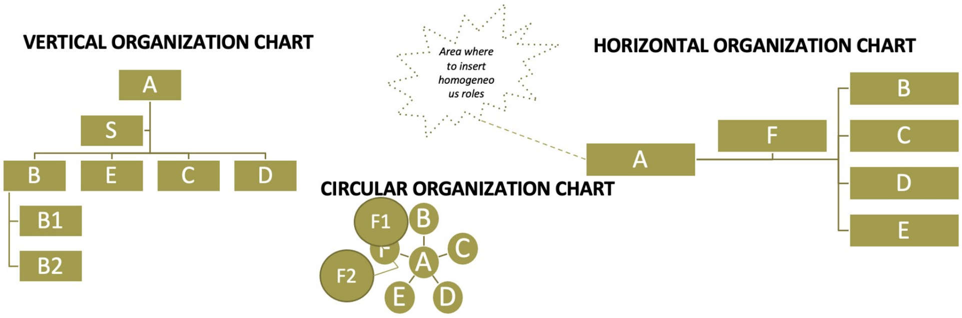

VERTICAL ORGANIZATION CHART
A
S
B
E
C
D
B1
B2
Area where to insert homogeneo us roles
HORIZONTAL ORGANIZATION CHART
A
F
B
C
D
E
CIRCULAR ORGANIZATION CHART
F1
B
F
A
C
F2
E
D

Figure 1.14: Organization chart

The organizational chart is a system of conventional symbols that illustrates the organizational structure of a company. Through the help of shapes, colors, and layouts (e.g. layout: Vertical organization, Horizontal organization, Circular organization) the organizational chart shows the roles, the hierarchy, the levels of authority, and the control ranges within a company.

The organizational chart gives a brief and immediate representation of the corporate situation, by photographing it as a whole, highlighting the links existing between the various positions. On the other hand, it does not provide sufficient information regarding the tasks performed by each unit or division, nor it informs the reader about the relationships between them and the corporate functions that go beyond the hierarchical lines.

The organizational chart is a static photograph of the company structure at a given moment and does not immutably represent it throughout its entire existence. A change in the organizational structure can be dictated by various factors, however it is often linked to a change in the company's size: a growing company develops an increasingly complex organization and tends to implement an integration of its internal corporate functions with the aim of maximizing the use of its resources, homogeneously leading them towards addressing the planned strategy.

Therefore, it is possible to state that a company develops throughout its whole life, if managed correctly, a natural growth both quantitatively (e.g. the number of employees, the turnover, etc.) and qualitatively (eg. organizational growth, technological complexity, management systems, references, etc.). Such growth process generally goes through six stages:

1. **birth** (absence of any structure) →it corresponds to the company's foundation by an entrepreneur, with a production focused on one or few products. Having a limited number of employees (generally from 5 to 20), the company can operate even without a formal organizational structure. The entrepreneur acts also as a manager and is directly responsible for multiple business processes and activities. Delegation is very poor. At this stage, a possible cause that could determine the non-survival of the company is the entrepreneur's excessive reluctance to decentralize the management processes or to delegate the business' activities;

2. **consolidation** (simple structure, embryonic organization) →at this stage there is still no official organizational chart with formal descriptions of the organizational units and of the managers in charge of the activities. However, the entrepreneur is ”forced” to substantially (sometimes even formally) delegate some activities to his trusted collaborators who are accountable for their success or failure. This necessity is generally dictated by the increase in the volume of business that consequently leads to a higher number of employees and workers (from 20 to 50 employees);

3. **formalization** (formal structure, usually functional O.S.) →at this stage the process of creating areas of competence and of delegating responsibilities is already

started and gets officially formalized with an explicit organizational structure (in most cases a functional O.S.). Responsibilities are identified in written documents (called job descriptions) focused on the management of resources, on the performance measurement, on the drafting of summary information needed to support operational management, etc.

4. **innovation management** (structures divided into cross-functional teams) →companies generally reach this level of maturity when they start to form cross-functional teams composed of specialists "borrowed" from the several functions of the organization, temporarily allocating them into project groups in charge of completing innovation projects (such as the development of a new product, process, plant, etc.).

5. **diversification by growth** (divisional O.S.) →not all companies feel the need to develop a highly innovative potential (as in the innovation management stage), nevertheless they can still survive in stable competitive environments adopting different strategies. These companies can often reach further maturation through a process of product or market diversification (e.g. developing new products to access new market sectors or targeting different countries). A company adopting a product/market diversification strategy can opt for a divisional O.S. (e.g. an international company). The new divisional unit can come from the acquisition of an external company or from the creation of a foreign subsidiary.

6. **intensification of innovation** (structure by matrix project) →this evolution of the operational structure can also be obtained directly starting from product/market diversification phase. Furthermore, this type of activity, or the inclusion of innovation in products/processes already in place or innovation understood as a focus/objective on which the creation of a new product, is limited in time, associated with very specific moments of the cycle life of the company, but above all also linked to specific products.

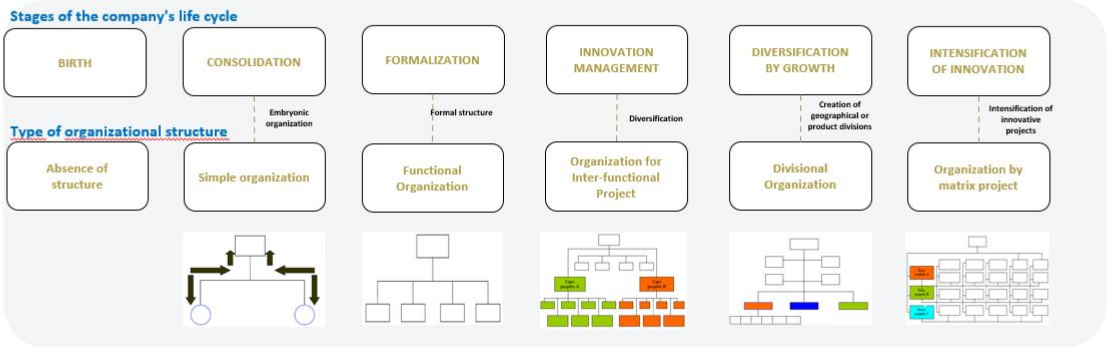

Stages of the company's life cycle
BIRTH
CONSOLIDATION
FORMALIZATION
INNOVATION MANAGEMENT
DIVERSIFICATION BY GROWTH
INTENSIFICATION OF INNOVATION
Embryonic organization
Formal structure
Diversification
Creation of geographical or product divisions
Intensification of innovative projects
Type of organizational structure
Absence of structure
Simple organization
Functional Organization
Organization for Inter-functional Project
Divisional Organization
Organization by matrix project

Figure 1.15: Company life cycle associated with the Organizational Structure

In the figure below, the development stages of the organization are highlighted, in order to represent the life stages of a company in relation to its size of the same.

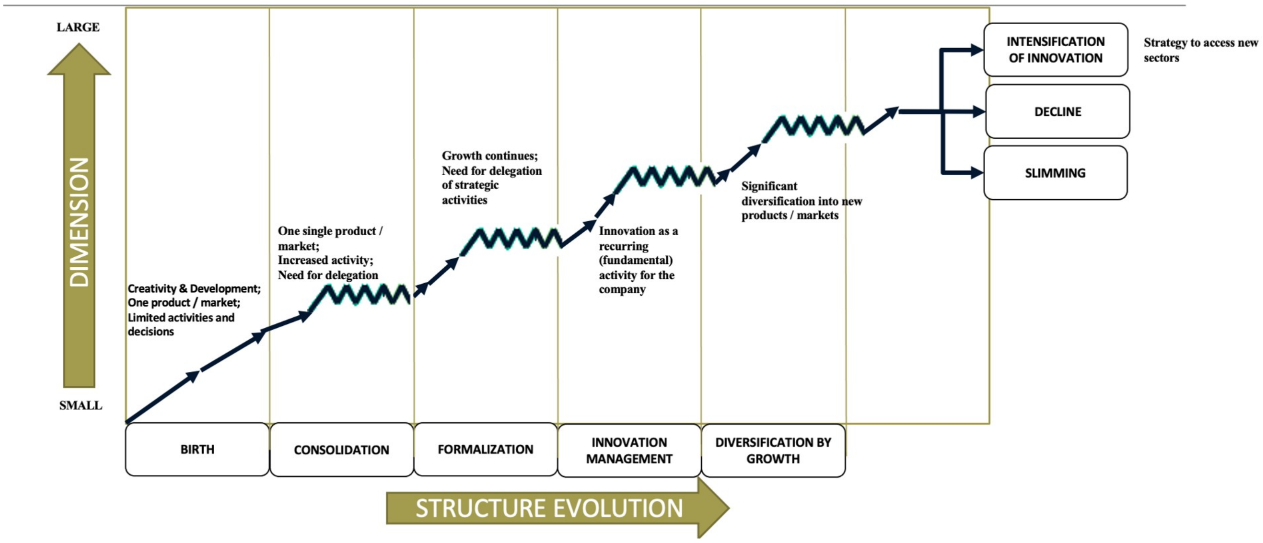

LARGE
DIMENSION
SMALL
Creativity & Development; One product / market; Limited activities and decisions
One single product / market; Increased activity; Need for delegation
Growth continues; Need for delegation of strategic activities
Innovation as a recurring (fundamental) activity for the company
Significant diversification into new products / markets
INTENSIFICATION OF INNOVATION
Strategy to access new sectors
DECLINE
SLIMMING
BIRTH
CONSOLIDATION
FORMALIZATION
INNOVATION MANAGEMENT
DIVERSIFICATION BY GROWTH
STRUCTURE EVOLUTION

Figure 1.16: Evolution S.O.

Chapter 2

Business Management Systems

2.1 The Enterprise as a System

Some research conducted in the late 1970s has shown that organizational change is not resolved only through structural interventions, related to the simple reworking of the organization chart. Scholars Waterman, Peters and Phillips through their article published in 1980 in the magazine "Business Horizons" they had come to affirm that: the corporate structure is not the organization - the representation of an object is not the object. Organizing involves much more than creating a mere graphic representation, because every organization consists of a set of different variables that, together with the corporate structure, affects the performance of the company.

Taking the steps from such axiom, the theory of the so-called "7S model" was elaborated, according to which an organization is a complex system consisting of multiple interacting factors, classified into seven macro-variables:

1. **strategy** – it represents the strategic choices: e.g. entering new markets, developing new products, improving customer services, drawing up development plans, etc.;

2. **structure** – it indicates the relationships of formal dependence, including the number of hierarchical levels and the extent of control/responsibility of managers

and supervisors;

3. **systems** – it indicates the management control systems, the human resource management systems, the planning systems, the information systems, the distribution systems;

4. **staff** – it indicates the staff, its background, its skill levels, its training, and the career development, etc.;

5. **skills** – it indicates the available expertise, which includes the skills, the knowledge, and the know-how necessary to carry out the business activities;

6. **style of leadership** – it indicates the style of leadership adopted by the top management;

7. **set of values** – it indicates the core values used as guiding principles.

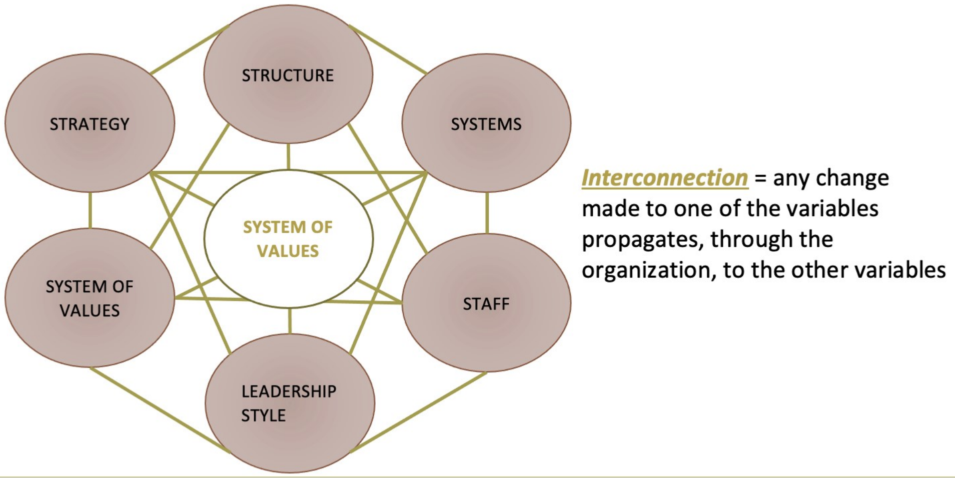

STRUCTURE
STRATEGY
SYSTEMS
SYSTEM OF VALUES
SYSTEM OF VALUES
STAFF
LEADERSHIP STYLE
Interconnection = any change made to one of the variables propagates, through the organization, to the other variables

Figure 2.1: 7 S

The organization of a company, therefore, cannot be explained by its structural characteristics alone, because it is a complex system of multiple interrelated variables. At the same time, in order to obtain a certain desired effect on one of the ”7 S”, it is often

necessary to implement consistent changes also to the other factors of the scheme.
Assuming that a company is an organized system composed of a multiplicity of interdependent functions, we can now define the possible meanings of the term ”Enterprise”.
An Enterprise can be considered as:

- an economic organization – consisting of a combination of coordinated resources (human, material, intangible) used in a logic of planning, for the achievement of the targets established by the company;
- a social system – as the set of rules that regulates people's behaviour, and as the relationships that bind human resources to the organization with the peculiar characteristics of each resource who carries out a specific activity;
- an equity structure as a set of assets organized for the production of income that has to generate operating profit.

Note. *Difference between company (C.) and enterprise (E.): the C. is the means (the operational tool used by the company to achieve its goals), while the E. is the organizing activity carried out by the entrepreneur.*

2.2 Planning and coordination mechanisms

In order for the entire business system to be strategically oriented towards its goals, it is necessary that a full organizational coordination is achieved, harmonizing the functioning of all the production units which are generally located at different levels of the organizational structure (both vertically and horizontally).
There are two factors that most influence coordination: the uncertainty of the task to be attributed to the subjects and the complexity of the functions performed in the units (so-called task or task). The uncertainty in the definition of the task can be considered, in modern approaches, as the most impacting benchmark in the choice of the business organization. The uncertainty of the task lies above all on the differences between the necessary information (that depend on the diversity of outputs and inputs, as well as on

the expected level of performance) and the available information (that are based on the skills and specializations available in the information system). Variability and complexity is also another factor that makes a task uncertain: e.g. the number of exceptional cases that may arise, the time required for the analysis (a long time for the analysis makes a task more uncertain), the level of know-how required, and so on.
Three categories of tasks can be identified:

1. a defined or a to-be-defined task identified or to be identified;

2. a subjective task (that varies according to the company, the sector in which operates, the corporate strategies, and the available resources);

3. an objective task (that operates exchanges with the external environment reacting to exogenous changes).

The planning pattern must take into account the variability and difficulty of the tasks. A task characterized by little variability facilitates a high level of organizational structuring (like a mechanization, routinization, and standardization of the processes) with the possibility to utilize methods of organizational coordination ”on budget” at a relatively low cost (external figure to the organization that deals with coordination of activities and that for such work is rewarded in relation to previously agreed contract).
The degree of difficulty of the task influences the use of more or less sophisticated decision-making and coordination models, forcing the organization to increase or decrease the level of professional competences (specialization) and to increase or decrease the level of organizational complexity (horizontal and vertical).
Once the right weight of the tasks has been defined, the planning strategy has to organize the work of human resources. According to the objectives to be achieved, the definition of the most cost-effective combination of the resources who work within the company is the basis for an effective and efficient planning. The coordination mechanisms contained in the plans will have to identify:

- the functions to be performed and the people to whom they will be entrusted (who does what?);

- the roles, tasks, and duties to be attributed to each person and the related responsibilities (what are the individual activities to be performed and what are the responsibilities of each individual?);
- the methods and tools needed to organize coordination and collaboration between the various corporate units in order to create the best possible system of relationships (each individual is responsible for whom?).

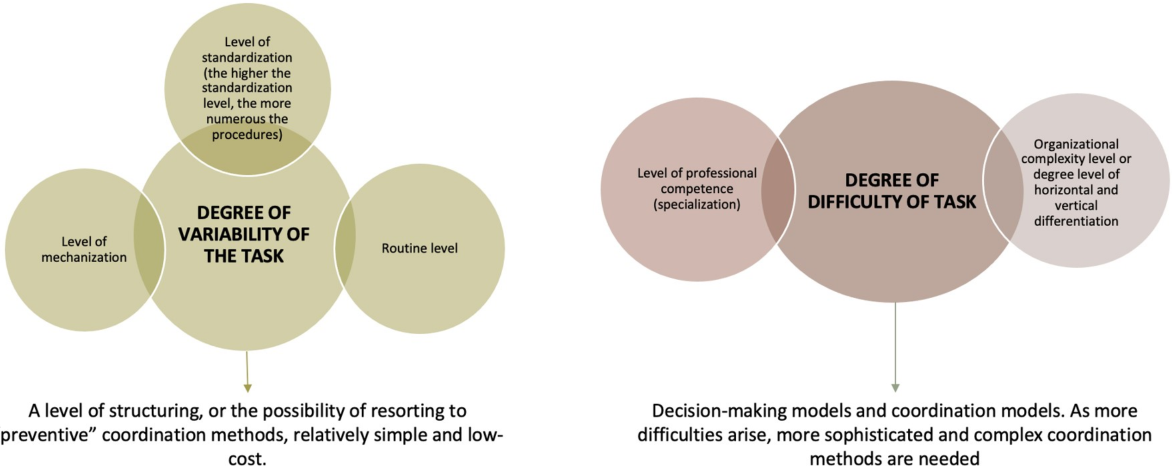

Level of standardization (the higher the standardization level, the more numerous the procedures)
Level of mechanization
DEGREE OF VARIABILITY OF THE TASK
Routine level
A level of structuring, or the possibility of resorting to "preventive" coordination methods, relatively simple and low-cost.
Level of professional competence (specialization)
DEGREE OF DIFFICULTY OF TASK
Organizational complexity level or degree level of horizontal and vertical differentiation
Decision-making models and coordination models. As more difficulties arise, more sophisticated and complex coordination methods are needed

Figure 2.2: Characteristics of Task

2.3 Business Management Systems

A Business Management System is defined as the set of tools that supports and guides the behavior of organizational actors in the pursuit of a collective action.

A system is generally composed of many interconnected elements (tools, methodologies, procedures, regulations, algorithms, technologies, etc.). Several systems often interact with one another in a way that does not allow to easily determine the extension of each individual system. For example, let us consider a quality management system: this type of system deals with regulations to comply with, responsibilities to activate, documentation to produce, detection technologies to adopt, training, monitoring, and so on. Building a quality management system means to identify and design each of these components so that the behaviors of company actors could be strengthened or limited as needed. The worker is the main recipient of the rules of a business management system. He/she is the major vector of the information (judgments, evaluations, decisions, statements, reasoning, speeches, etc.) that run throughout the company, using the most varied methods of interaction (from the physical interactions during meetings or gathering, to the interactions mediated by some sort of communication systems, etc.) and incorporates a fundamental capacity of executive action. The organizational actor (such as the administrator) has a role of regulating the worker's behaviors. The administrator has the ability (both cognitive and executive) to direct the workers of the system that manages (pursuing a collective action).

The cognitive and executive capabilities of the administrator must therefore be dedicated to the development of an organizational structure that best suits the characteristics of the company. Corporate organizations are forms of spatial and temporal arrangement of human activities with the aim of carrying out a collective action that has the requisites of effectiveness and efficiency, however defined. Each collective action necessarily requires a work distribution among the individuals who participate in that action and, at the same time, requires the identification and implementation of a method for coordinating the individual actions. An economic choice always underlies the decision of how the work is organized: work consumes physical, cognitive, and emotional resources that are scarce by their nature; human work can be enhanced with appropriate tools, or can be integrated or replaced with machines (even the relationships between individuals, groups, tools, and machines needs to be defined and coordinated by the management). Allocating labor resources for the company's purpose represents not only an organizational but also an economic problem.

Human activity (such as coordinating, arranging, and assigning functions or tasks in the most suitable way to achieve a specific purpose) is therefore at the center of every organizational problem. The capability to transform coordinated human activities is, in fact, a collective action (if there was only one individual, coordination would not be necessary).

2.4 Control management system and control cycle

The Management Control System (MCS) is a corporate governance system that acts before, during, and after the decision-making processes, in order to guide, structure, and translate them into effective and efficient actions. Such system is useful to establish goals that are consistent with the available resources and with the context (results' programming), to compare different tactics and plans of action (results' measuring), to check whether goals have been achieved and whether the adopted strategies have shown to be effective and efficient (deviations' analysis), and to introduce corrective actions.

The management control system can be represented as a process composed by four phases:

1. **goals' definition** – defining the objectives that need to be pursued through a specific system of action and based on a simplified model of reality;

2. **results' measurement** – measuring the outputs of the control system. They are influenced by two types of variables: Exogenous variables (that are non-directly controllable by the company); Endogenous and controllable variables (even though they are pre-identified, they may not always show as expected, since the forecasting model will always be a simplified model of the reality);

3. **Actual-to-forecast analysis** - analysing the causes that generated the deviations in order to distinguish the deviations that can be linked to exogenous factors from the ones that should be attributed to a mismatch between the forecast model and the actual reality;

4. **Setting up of corrective actions** - fixing the deviations highlighted by the actual-to-forecast analysis:

- if they are linked to endogenous causes, it will be necessary to work on the functioning of the system;
- if they are linked to exogenous causes, it will be necessary to adapt the objectives to the evolution of the external environment.

In addition to the **management control system (MCS)**, there are some other types of business management systems, such as:

1. **regulatory system** (it executes commands);
2. **cybernetic system** (it adapts its responses to the environmental changes);
3. **generative system** (it self-reorganizes through learning).

2.5 Regulatory Systems

For almost a century, from the second half of the 1800s to the first half of the 1900s, the Second Industrial Revolution has pushed scholars to develop theories regarding the so-called "scientific management" as a tool for profit's maximization. Among these, Taylor considered as a scientific progress the development of economies of scale implemented to satisfy the market demand. Scientific management (the organization of workers and machines through a scientific approach) started from the idea that an organizational system could be managed as a machine (that is created to achieve specific objectives and that operates rationally and efficiently) that could be systematically designed and implemented. Therefore, all the formal aspects (rules) of an organization became predominant in the scientific analysis elaborating the organizational and management theories.

The rules of each organizational system consist of explicit or implicit rules which structure the behavior of individuals and the interactions between them (instructions or program lines to be passively executed).

Rules can be seen as:

- Action Models – they guide the actions and the interests of a group of individuals working on a common goal in order to increase its efficiency. They are useful to solve conflicts between individuals with different goals within the same organization. Sometimes there can be contracts negotiated between groups with diverging interests.

- Bureaucracy – it becomes necessary when the size of the organization or the number of its control bodies increase. Bureaucracy also is needed when the variability of the external environment imposes new rules.
- Construction of meanings – they are representations of the organization. They indicate how things should happen within the organizational structure.
- History codifications – they are repositories of the organization's experience. Their collection represents a memory tool to ensure cooperation between actors that evolves over time through a chain of experience.

2.6 Cybernetic Systems

Unlike the regulatory system theories, by which human behavior must be subordinated to the needs of the organization, the cybernetic systems consider the individuals as the center of a network of fundamental relations and the company as an entity open to the changes of the environment.

The development of these systems is linked to the computer revolution; the answers to the questions posed by the management needs are processed by adapting the company to environmental changes, which are due to human factors, to the need for leadership, and to the networks of informal relationships.

Cybernetic theories (also called adaptive theories), placing human behavior at the center, highlight how jobs carried out in autonomy and with full responsibility provide satisfaction, improve productivity, and generate a greater involvement. Focusing on these issues allows us to leverage human needs, exploiting them as ”management levers”. Human resources of all categories agree in considering the transversal universal needs (growth, relationships, existence) as essential for their career. Among these are:

- needs for self-actualization;
- needs for esteem and self-esteem;
- belongingness and social needs;
- needs for safety and for an economic peace of mind;
- basic physiological needs.

In order to meet the needs of the human resources, the cybernetic model considers issues like: sharing corporate values (both by the entrepreneur and by the workers), professional growth (autonomy, responsibility, awards), social relations (team-working), safety (health, wages, career), and so on. Organizations that adopt such a strategy can be considered as “sociotechnical systems” that give value to the social and technical aspects of work, as well as to their interdependence.

In particular, a study on the quality of work by Herzberg in 1959 identified two factors that generate worker satisfaction or dissatisfaction:

- hygienic factors – that are linked to the condition of things;
- motivating factors – that are linked to the broadening of knowledge.

The **hygienic factors** concern the ”external” working conditions, such as the physical environment, the social environment, and the remuneration. The **motivational factors** are linked to the ”internal” sphere of a person, including the work tasks and the individual’s motivation in carrying them out. Some examples of moticationa, factors are:

- knowledge broadening,
- increase of relationships and creativity,
- autonomy (especially in conditions of uncertainty),
- real professional growth,
- principle of individuation.

2.7 Generative Systems

Under the pressure of globalization, the only possibility for a company to survive is to adopt a proactive attitude towards environmental changes by imposing its own (corporate) changes externally, so as not to have to passively adapt to the changes coming from the surrounding environment.

Generative systems are management systems enabling the development of collective actions that produce innovation. The enterprise must therefore produce innovation through a product/service differentiation. In order to produce innovation there is need for:

- new knowledge - the skills of the organizational actors are no longer executive (as with the regulatory model) but cognitive;
- collective memory - common and shared values, rules, procedures, and knowledge.

In the generative system, the discovery (and correction) of an error triggers a collective learning that becomes a heritage for the whole organization. For a generative organization, to find as much information as possible is vital in order to trigger the collective learning mechanism. Scholars like Walsh & Ungson (1991) propose to look for the information needed to such process within the individual and general conservation structures: the acquisition of individual information (conceptual maps, values, beliefs) and of a widespread cultural heritage (language, shared frameworks, symbols, stories) is followed by a transformation of such information into company data (procedures, rules, formal systems, technology) necessary to build the company's structures (roles, responsibilities) and ecosystem (physical settings); in this way an organization learns by building, experimenting, and reconstructing its own theory of action.

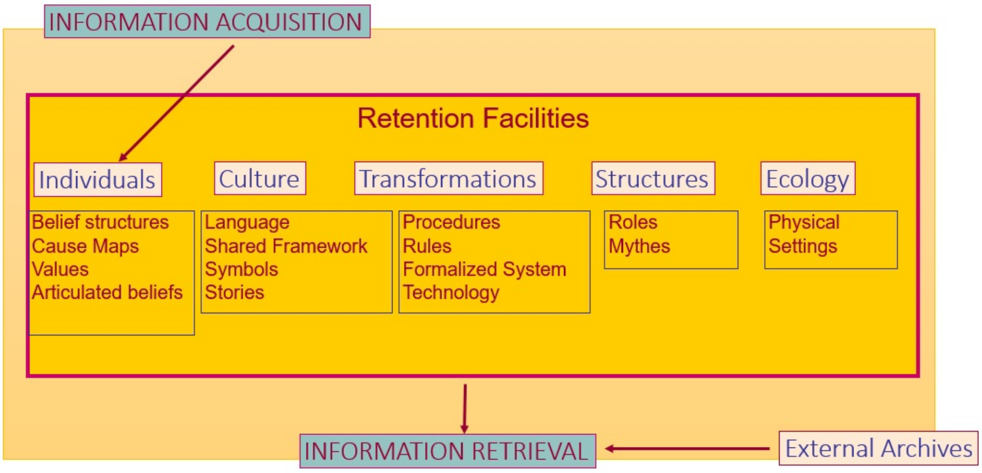


Figure 2.3: Walsh & Ungson Organizational Memory Process

The wealth of knowledge gained through correct management becomes skills. According to the generative system, competence can be conceived as the ability to assimilate rules, incorporating them into daily actions/activities. Four models are available for managing human resources' competencies:

1. **functional model** - competencies are composed of elementary units of capability that are specific to each corporate function. This model (which is typical of the organic approach) allows to build a catalogue of the required skills for each company function;

2. **Job model** - skills are the manners in which tasks are carried out and that lead an individual to achieve the expected results. This model adopts a ”by process” approach and suggests that competencies get generated by the tasks, the tasks' management, and the environment;

3. **Behavioral model** - competencies are individual and behavioral characteristics expressed while performing a job (individualistic approach);

4. **Action/reflexion model** - competences are ”knowledge in action” and ”knowl-

edge about action", used to reformulate the existing frameworks of action (analytical approach of action and reflection about action).

In conclusion, a human resource/individual is competent when:

- possesses the required skills at the needed degree;
- performs his/her duties as expected;
- possesses behaviors and personal characteristics that generate effective results;
- is capable of redefining his/her action in order to achieve higher performances.

To date, knowledge management is one of the fundamental elements for organizational competition. In the long run, an organization that does not manage Competence/Knowledge cannot maintain a position in the dynamics of a competitive market.

Chapter 3

Quality Systems

Since the 1980s we have been witnessing a process of continuous change of the concept of production activity. In fact, producers are increasingly improving the way in which production is carried out in order to achieve a higher level of satisfaction for themselves and for their users. This process is accompanied by increasingly strong markets' globalization that, while offering many opportunities for growth, it also forces many companies to choose between competition and marginalization (often experienced as an exclusion from markets or as a simple survival condition). Competing therefore means to offer products/services resulting from activities that meet high-demanding quality parameters.

In a globalized market, it becomes increasingly difficult to win over consumers/end-users and trigger a loyalty process towards the company or towards the corporate brand. The growing needs of the customers, that are becoming more and more evolved and attentive to the technical and qualitative attributes of the products, as well as to the service provided before, during, and after the sale, oblige the entrepreneur to implement new mechanisms for processes' measurement and control within the company. The new underlying philosophy imposes a way of thinking that is different from the traditional one and more aimed at achieving a "total quality" through continuous improvement.

3.1 The Quality Management Systems

The industrialized world has undergone a major change in recent years: the gradual transition from a production economy to a market economy, accompanied by a profound upturn in mentality and business culture that has led to great changes in the activities' internal organization, in the division of tasks, in the transmission of information, etc. Competition in the global markets imposes to adapt the offered products and services to the expectations of the customers, while also reducing costs. A continuous improvement of the products and services' quality provided to customers, as well as a reduction of costs, a rationalization of the structures, the elimination of waste, and the adoption of advanced technologies become of vital importance to compete in the global markets. Traditional systems are no longer useful to satisfy these new needs, nor to ensure an effective business integration because they provide a vision that disregards both the internal customer-supplier relationships and the influence that the outputs of a certain organizational unit have on the activities of other units (processes' logic). In a management system focused on Quality, instead, every member of the structure is directly involved: such as those who take part in the production process, those who guarantee the raw materials' procurement, those who issue the invoices, those who have direct contact with the customers, and so on. According to this model, all company functions are involved in a joint effort to achieve full (customer satisfaction).

Customer satisfaction, achievable through the product, is linked to the overall result of the production process. The quality approach must therefore invest in:

1. competitiveness
2. productivity
3. profit
4. product quality
5. quality of the organization

6. quality of relationships between people
7. quality of the work of each person in the company
8. quality of the company's image on the market
9. management results
10. customer service
11. safety at work
12. attention to the environment
13. attention to shareholders and other interested parties (stakeholders)

An organization that evaluates these elements of production as a whole tends to achieve the result of Total Quality. Such a result needs to go together with competitiveness, that is obtained through the following rules:

1. keep the cost-effective method (the best customer's satisfaction at the lowest possible cost);
2. listen to the requests for specific characteristics of the products/services (Required Quality);
3. check the output (the expected quality);
4. effectively interpret the latent quality (the customer can have unexpressed potential needs that the company must try to discover, sometimes even creating a new need);
5. control the processes (the planned quality + the obtained quality).

3.2 The modern root of quality

The attention to quality first started in the late second post-war period. Following the destruction caused by the war, the European market had a huge need for goods of all kinds and the USA had the necessary production capacity to put enough goods on the market. In this context, the quality of the product did not matter as much as its availability and quantity.

Later, the saturation of the markets and, in some sectors, the surplus of supply over demand triggered the quest for increasingly better and competitive products, leading to the ”end of a logic of quantity in favor of a logic of quality”. The main scholars researching on the transition from quantity to quality were:

- W. Shewhart (1945) - through the use of statistics, he laid the foundations of the modern concept of ’Quality Control’. By studying the defects it is possible to identify the causes and the areas of major criticality. This approach leads to the implementation of a control strategy with real Quality plans.
- W.E. Deming (1950) - he contributed to the spread of organizational quality criterias based on the experience developed in a period of exceptional request for military power, during which the United States produced big quantities of war supplies at low cost.

Deming Wheel (also known as PDCA) considers four cyclical phases that can be described as follows:

- **PLAN**:
  - observe and analyze;
  - isolate the real causes;
  - define corrective actions.
- **DO**:
  - prepare;
  - apply;
  - verify the application of the interventions.

- **CHECK**:
  - verify the results ;
  - compare with the planned goals.
- **ACT**:
  - standardization and consolidation;
  - preparation of the next planning phase.

Deming's theories (which are part of the Japanese models based on the concept of Total Quality and division of labor) consider the business management to be based on the following steps

a) use of the Planning-Production/Sales-Control-Replanning cycle to improve Quality (the so-called Plan – Do – Check – Act cycle better known as the Deming wheel);

b) control of the process using Shewhart's control charts [1] (a tool used for statistics aims to keep track of the parameters of a process);

c) engagement of the workers in the quest for production problems, the solution of which each time generates a small improvement (quality circles).

The transition for dealing with problems, from the search for a *culprit*to the search for the *causes* found fertile ground in Japan. In this country the quality matter was no longer seen as an exclusively technical issue but also involved the managerial aspects of the production choices (that were now based on quality). In addition, in order to

[1] **ISO 7870-1:2019**
Dr. Walter Shewhart in 1924 proposed the control chart as a graphic means to apply the statistical principles of significance to the control of a process. Control chart theory recognizes two types of variability. The first type is random variability due to "random causes" (also known as "common/natural/random/intrinsic/uncontrollable causes"). This is due to the wide variety of causes that are constantly present and not easily identifiable, each of which constitutes one very small component of the total variability but none of which contributes significantly. However, the sum of the contributions of all these unidentifiable random causes is measurable and is assumed to be inherent in the process. The elimination or the correction of common causes may require the decision to allocate resources to radically change the process and the system. The second type of variability represents a real change in the process. This change can be attributed to some causes identifiable that are not an intrinsic part of the process and that can, at least theoretically, be eliminated. These causes identifiable are referred to as "assignable causes" (also known as special/unnatural/systematic/controllable causes) of variation. They can be attributable to issues such as a lack of uniformity in the material, a broken tool, workmanship or procedures, erratic equipment performance, or environmental changes.

enable the operational staff to find solutions to problems, Japanese scholars developed a package of seven statistical tools ("7QC Tools" by Ishikawa) that are easy to apply:

- histograms;
- cause-and-effect diagram (or Ishikawa or fishbone diagram) [2];
- cata collection sheets [3];
- pareto diagram [4];
- stratification analysis [5];
- control charts [6];
- scatter diagrams [7].

These theories were then followed by those of:

- P. Crosby (1966), who introduced the concept of making the worker responsible for the quality of the activities entrusted to him;
- K. Ishikawa (1970-1980), who developed in Japan the concept of "Company Wide Quality Control" by which quality management is implemented by providing the entire company with quality control tools;
- W.E. Deming (1980-1990), who developed the concept of "Total Quality Management" introducing Quality Management also in the Western Countries (in lieu of Quality Control) and extending the objective of Quality as a *commitment* to all company sectors.

[2] Refer to the definition in the Glossary
[3] Refer to the definition in the Glossary
[4] Refer to the definition in the Glossary
[5] Refer to the definition in the Glossary
[6] Refer to the definition in the Glossary
[7] Refer to the definition in the Glossary

3.3 Deming Wheel

The "PDCA" (or Deming Wheel) represents a scientific approach with universal validity useful to solve business problems and to rigorously and systematically deal with any activity. This term is the acronym of the initials of the four phases into which the problem solving process can be divided:

- √ P = PLAN = plan before start:
  - the planning phase starts by identifying and describing a problem analyzing its main aspects. The activity of data collection through observation and analysis is preliminary for identification. Only verifiable and reliable data and facts on representative and significant samples of the problematic phenomenon must be used for the analysis, excluding any personal opinion. A sample is significant when all the collected data has numerical consistency. Among the most suitable tools for data collection and presentation respectively are brainstorming and flowchart;
  - the next step after collecting the data is to define the general objective in a clear, quantitative, and complete way, identifying both the tangible and intangible benefits that can be obtained with its achievement (economic effects). Timing, indicators, and control tools must also be identified;
  - the following step is to analyze the problem, identifying or studying its negative effects, and defining their importance as well as the priorities for action. Any improving action must be based on objective considerations, that is, on the results coming from the data-processing which reliably represent the phenomena being studied. Histograms are a good tool for data-analysis;
  - the analysis of the problem must investigate all the possible causes, developing a complete picture of all the causes of the problem and selecting the most probable causes (hypothesis) that will then have to be verified with the data-collections, the elaborations, the experimentations, etc. Finding a shared

theory about the causes of the problem and defining the ”law of priority” is fundamental. To do so, it is often useful to iterate the process until the real causes are identified. The cause and effect diagram (or Pareto diagram) is the most commonly used tool for identifying the causes of a problem;

  - finally, corrective actions must be designed, starting with the quest for and analysis of possible corrective actions, identifying the most effective ones, then designing the activities to be performed, together with the necessary methods and timings, and defining the criteria for evaluating the results.

- √ D = DO = do what has been decided:
  - prepare the interventions defining whatever is needed to their implementation;
  - in the “DO” phase it is first necessary to train the human resources in charge of implementing the corrective actions. In addition, corrective actions will have to be implemented together with the planned solutions. Finally, it is necessary to verify the correct application of those actions.

- √ C = CHECK = measure the results:
  - in this phase it is necessary to verify that the corrective actions are carried out on schedule, checking the results obtained with such actions. In addition, managers will have to compare the new collected data with the one available at the start of the process and the achieved results with the planned goals. If goals have been achieved, the Act phase can start; conversely if goals are not achieved, it will be necessary to start a new PDCA cycle on the same problem, critically analyzing the various phases of the previous cycle in order to identify the causes that lead to the non-achievement of the goal.

- √ A = ACT = standardize and create a procedure or repeat a new cycle:
  - in this phase managers will transform the new solution into a common practice in order to consolidate the corrective actions and make them irreversible.

It is also necessary to carry out a specific in-depth training for the workers and to schedule some tests to check the validity of the corrective actions, immediately establishing procedures and timings. Possibly, it is also useful to design a new PDCA on the same subject for further improvements (continuous-improvement activation).

The graphic representation of the P.D.C.A. is a moving circle called "Deming wheel", which shows the cyclicality and continuity of the application process.

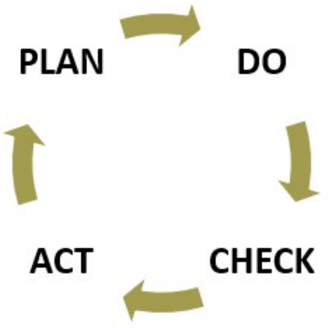


Figure 3.1: Deming cycle

3.4 Total Quality Management

The consistent monitoring of some intangible indicators, such as customer satisfaction, the ability to innovate, and the level of the intellectual capital (therefore not only the usual economic and financial data) allows the management to increase the competitive potential of the company. The main goal of the management is to get the most out of people. Managers have to focus their efforts on employees, customers, and suppliers in order to achieve success. The vitality of the company is, in fact, closely linked to all the factors that enable it to generate profits, such as:

a) the organization's capability to innovate;

b) the ability to satisfy its customers;

c) the level of energy and commitment of its staff.

The above-mentioned items are signs of vitality that can show whether a company is capable of generating new profits in the medium/long term. Looking at the economic and financial indicators only (profits, turnover, market shares, etc) is not sufficient to guarantee the company's ability to face future competitive scenarios. The imperative is not to stop and to constantly improve processes, working methods, customer services, and competitiveness.

The possibility to acquire competitive advantages is increasingly affected by the performances that the company is able to ensure in terms of quality. The advantage that arises from operating in a perspective of T.Q. (Total Quality) is well known, but the main problem lies in how to organize a company in such a way that quality's improvements become the corporate's life culture and a priority in the management orientation. The concepts at the basis of the T.Q. are:

1. competition on value/cost
2. quality reported to processes
3. supplier/customer relationship
4. teamwork
5. continuous improvement strategy
6. spreading of the attention to quality to all company's sectors, including the economic and the social ones
7. internal and external customer satisfaction[8];
8. improvement of internal (with the staff) and external (with the environment in which the company operates) relationships
9. lower costs
10. easier troubleshooting

[8] By 'internal' customer we mean those who are downstream of a process and who are part of the organization; while yes they define 'external' end customers who will use the product/service and who do not belong to the organization.

11. greater information and communication, with the involvement of all processes

A company that wants to operate with a T.Q. should provide a product/service that is able to respond to a diversified and constantly evolving demand. Such a goal can only be achieved with the involvement and the active participation of all the employees, as well as with a strong orientation to processes, implementing the following:

a) continuous improvement (inefficiencies' reduction, knowledge of the statistical tools);

b) data and facts (to be used for managerial decisions' support; ability to transfer values, objectives, and incentives to the coworkers);

c) leadership (to be implemented both at senior management and at all operational levels; human resources must be trained, developed, motivated, and made aware of the corporate objectives in order to create value);

d) involvement of human resources (set of interrelated activities characterized by input/output; processes management);

e) added value (the organization generates value through its processes);

f) systematic approach to management (suppliers are intended as an extension of the process - partnership);

g) collaborative relationships to create value (the customer is considered as the one who allows the existence of the organization).

3.5 Process management (BPR and BPI)

The evolution of the business management systems has generated a new need for the elaboration of plans (with a varying complexity) containing strategies, goals, and processes designed for their achievement. A "business process" can be defined as a structured set of logically related activities that are carried out by the resources of an organization and that transform inputs into outputs to be used by customers or recipients. The drafting of a business process is a task of the Process Management, which should elaborate a business strategy aimed at improving the business processes.

There are two different approaches to process management:

1. **business process reengineering (BPR)** - It is a redesign of existing business processes to achieve performance improvements within and between different organizations through the critical and radical analysis of the organization's business processes;

2. **business process improvement (BPI)** - It is an analysis of the business processes associated with incremental and continuous improvements over time.

A process management strategy that integrates the activities of both BPR and BPI allows for a substantial improvement of the performances by redesigning business processes. Such improvement gets preserved and increased through the adaptation of the processes to changes of the external environment.

3.6 Business Process Reengineering. Methodology and phases.

A Business Process Reengineering (BPR) is an organizational intervention consisting of a profound revision of the operational processes that is usually required when such processes are no longer adequate to the business needs.

The reengineering or redesigning of a process is a radical and comprehensive revision of the process. Simple adjustments, calibrations, or operational improvements by themselves are not enough to define a process reengineering.

The radical revision of a process usually requires a series of dedicated activities that should be implemented through a specific project, while small variations of the regular activities are not considered sufficient.

The phases of the BPR can be summarized as follows:

I. identification of the product/service:

   a) defining the mission;

   b) defining the owner of the product/service;

   c) identifying suppliers and inputs;

   d) identifying customers and outputs.

II. definition of the quality requirements:

   a) defining the performance goals;

   b) identifying the quality requirements;

   c) defining the quality indicators;

   d) identifying the performance objectives;

   e) defining the performance indicators.

III. reconstruction of the current process:

   a) use of flow charts;

   b) breakdown of macro-processes into processes;

   c) identification of the phases of the process to be reengineered;

   d) identification of the activities.

IV. identification of criticalities/causes:

   a) assessment of failure in achieving the quality requirements;

   b) assessment of failure in achieving the performance objectives.

V. V) Identification of the hypotheses for resolution:

   a) recomposition of fragmented activities;

   b) flows' categorization and differentiation;

   c) parallelization of the activities;

d) elimination of 'NON-VALUE' activities.

VI. process' redesigning.

3.7 Business Process Improvement. Methodology and phases.

A Business Process Improvement (BPI) plan is adopted with the aim of obtaining a maximization of the processes' functionality by improving the operational flows through incremental changes, starting from the operating procedures already in place.

The goal is to make the entire process effective and adequate, avoiding the error, often recurrent in complex organizations, of improving the single sub-processes with no benefits to be gained by the general process as a whole. In fact, companies often already have several working groups dedicated to the improvement of their sub-processes; the crucial problem is that they often do not know the effect their work has on the other activities of the process. As a consequence, the sub-processes are often optimized but the process as a whole has not improved. Unlike the TQM (Total Quality Management) which has a radical, bottom-up approach on the company's structure, the BPI plan immediately involves in medium and long-term activities both the management and the contact people, with the aim of improving the efficacy of the process in its entirety, achieving the required results, reducing the necessary resources (eliminating errors and reducing delays), and contributing to the enhancement of knowledge. The phases of a BPI can be summarized as follows:

I. organize to improve :

a) the top management selects the critical processes;

b) appointment of the Process Owners [9];

c) the Process Owners organize work groups or Process Improvement Teams (PIT);

d) the PITs develop the project plan;

[9] Refer to the definition in the Glossary

II. understand the process:

   a) the PITs develop a mapping of the current process;

   b) collection of all the available information;

   c) alignment of procedures with current activities;

III. "fluidization" of the process:

   a) simplification and removal of the activities that do not add value;

   b) process' improvement;

   c) identification of the phases of the process;

   d) identification of the activities;

IV. implementation, measures, and control:

   a) the 'new' process is 'put into action;

   b) measurement and control systems are established.

V. 5. Continuous improvement.

3.8 Business Process Management. Methodology and phases

The Process Management is a strategy aimed at improving the business processes. A "business process" is a structured set of logically related activities that are carried out by the resources of an organization and which transforms inputs into outputs used by customers or recipients. The evolution of this definition is traced back to scholar **Davenport** (1993) who defined the process as *"an organized set of activities aimed at achieving an output for a specific customer or market"*. Likewise for **Hammer e Champy** (1993): *"a process is a set of activities which, assuming inputs of various kinds, produce an output of value for the customer"*. Currently (2000) we define a process as a set of related or interacting activities that transform input elements (INPUT) into output elements (OUTPUT) UNI EN ISO 9000.

The Business Process Management (BPM) is a strategy that integrates the notions of BPR and BPI in order to both obtain:

- a substantial improvement in performance through the redesign of the business processes (peculiarity of the BPR);
- maintain and increase such improvement through continuous adjustment (peculiarity of the BPI).

The phases of the BPM are:

I. **planning**

   a) selection of the processes to be reengineered (macro-processes →processes →process activities);

   b) process Owners and working groups definition;

   c) definition of the goals;

II. **information gathering**

   a) data collection and documentation (organizational charts, IT procedures' manuals, work flows, internal documentation);

   b) users' interviews;

III. **mapping**

   a) process analysis:

      I. description of the basic information,

      II. applications' correlation (tools),

      III. correlation to documentation (manuals),

      IV. correlation to cost and performance information,

      V. definition of risks and controls;

   b) organizational structures' analysis (detection of the operating units and roles, allocation of resources, attribution of activities to structures);

c) process modeling;

IV. **sharing**

a) sharing of the obtained results;

b) process redesigning based on sharing;

V. **monitoring**

a) processes' monitoring based on cost, timings, and quality;

b) constant processes' updating;

c) KPIs definition.

The goals and methodologies referred to in the planning phase (point I, letter c) conceptually follow a preliminary drafting of a BPM for:

1. **workloads' sizing** - it allows to optimize the processes based on costs, timings, and volumes. It defines the correct sizing by highlighting the workloads assigned to the various structures involved in the process;

2. **compliance management** - it facilitates the correct compliance with regulations which require an updated mapping of the processes;

3. **skills management** - it supports the company on the management of the human resources particularly regarding the allocation of such resources in the right positions;

4. **software systems management** - it allows to define the effectiveness of the information systems in supporting the business processes;

5. **cost analysis** - it analyzes the costs associated with a certain operation, which helps identifying the overall cost of a product line;

6. **performance monitoring** - it monitors the performances by defining the performance indicators;

7. **business impact analysis** - it analyzes the impacts that disastrous events would have on business processes considered critical for the business;
8. **risk management and control** - it allows for the identification of the company's risks and the controls placed to oversee such risks.

Chapter 4

Project implementation methodologies

4.1 Definitions (Process - Organization - Project)

Some useful definitions are now provided, which are useful to introduce the following paragraphs:

- **Process** - a set of activities carried out continuously within the organization. Such activities consist of a sequence of known operations performed to achieve the organization's objectives.
- **Organization** - a public or private entity formed by groups of people who carry out activities of various nature, aimed at specific objectives.
- **Project** - a set of activities that generate changes within the organization in response to the changing conditions of the reference context (e.g. the market, civil society, the objectives of the organization, etc.). A project generates changes in an organization that may affect infrastructures, machineries, the internal organization, personnel, or any other aspect. These changes generate needs for revision of the internal processes of the organization indispensable for adapting operational processes to the new situations generated.

4.2 Project and Process

It is important to distinguish a business activity that forms a process from one that is instead part of a project. It could be argued that such matter has only a semantic importance and that labels like “project” and “process” are interchangable. While this is partly true when projects and processes overlap, there is a substantial difference between these two that concerns the way in which activities are managed.

A project is about creating something new or implementing a change. A process instead aims to create value through the repeated execution of an activity. In a project, objectives and plans can be changed by the stakeholders. Processes, on the other hand, are established procedures for work currently in progress and can generally be changed only with new planning and investments. A project is always temporary as it has a defined beginning and end, as well as a defined scope and resources. Furthermore, every project is unique in that it is not a routine operation, but a specific set of operations designed to achieve an individual goal.

In short, the borderline that distinguishes a project from a process can be marked by different elements, depending on each company:

- the frequency with which a certain activity is repeated in an organization and whether such activity is routine or not (the frequency with which the activity is repeated increases from project to process);
- the degree of risk associated with its implementation (the frequency with which the activity is repeated increases from project to process);
- other elements that vary depending on the specific company, for example the amount of the investment or the number of days required for the implementation of the project/process.

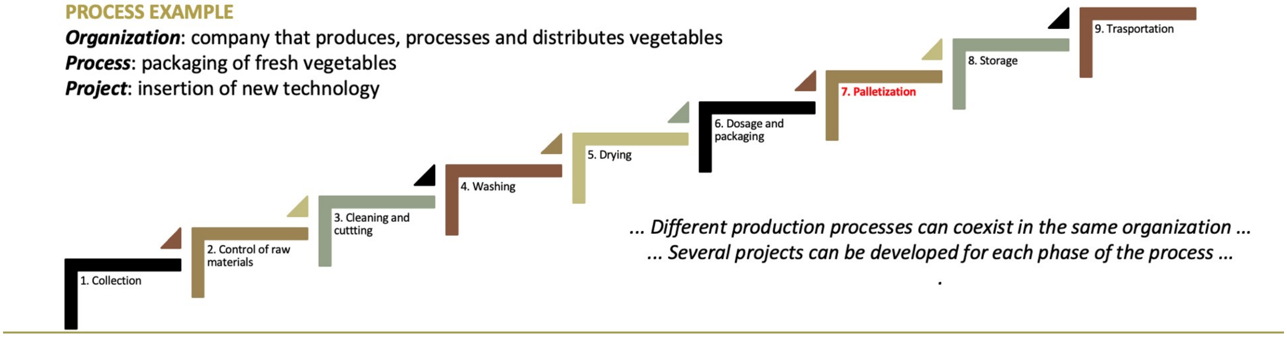

PROCESS EXAMPLE
Organization: company that produces, processes and distributes vegetables
Process: packaging of fresh vegetables
Project: insertion of new technology
1. Collection
2. Control of raw materials
3. Cleaning and cuttting
4. Washing
5. Drying
6. Dosage and packaging
7. Palletization
8. Storage
9. Trasportation
... Different production processes can coexist in the same organization ...
... Several projects can be developed for each phase of the process ...

Figure 4.1: Process - Project

The same activity can represent a process for a certain organization and a project for another organization. For example, the construction of a new building is a process for a large construction company while it represents a project for somebody who wants to build a new home for personal use or a new office for his/her business.

High-risk or change-making activities that can be defined as a *new project* require a dedicated management approach that differs from the day-by-day management (even though such activities are not always recognized and managed as projects).

It is important that appropriate management techniques and methods are applied for a correct implementation of the projects. Management failures often result in the failure of the entire initiative.

Finally, both processes and projects aim at achieving a goal through a sequence of planned activities. Both typically need teams and resources in order to be run.

4.3 Project and Project Management

The International Standards Organization (ISO) defines a project as: *“A set of coordinated activities, with a precise beginning and end, aimed at achieving a specific objective, based on time, cost, and resource constraints”* (ISO 9000).

In business organizations, several project management methods are used. Some of them follow some specific standards created ad hoc to meet the specific needs of the company.

Others, on the other hand, follow well-known consolidated methods that are widely used in various companies.

While some of these methods can be found standardized in project management certifications, others are still not very formalized. Regardless of the method chosen for managing a project, it is important to use a well documented process. The larger and more complex the project is, the greater is the care and the details needed in the processes. In other words, it is not essential to understand which method is best for managing a certain project but, rather, it is fundamental to have a clear idea of the life cycle of the project. Depending on the project management method in use, there may be differences in the nomenclature of some phases of the projects. Basically they belong to the same categories

4.4 Project life cycle

The Project life cycle describes the necessary phases (or steps) that need to be followed in order to successfully complete a project. They range from the initial inception [1] phase (which is included in the definition of Planning) to the final phase of Post-Deployment. In detail, the six phases are:

1. **planning**

    a) design and creation of the project through a series of meetings (brainstorming);

    b) examination of the project proposals that have passed a preliminary validation;

    c) definition and approval of the Project;

    d) definition of the Project Plan and of how the project should be implemented;

    e) selection of the specific approach to be used for the implementation;

[1] Concept phase of the product/project during which the necessary information is acquired to confirm feasibility or not of the project

f) allocation of the budget for the project;

2. **design**

   a) architecture's design;

   b) design of the product and its characteristics;

   c) creation of lists containing the details and the tasks of the project;

   d) creation of a Project Scheduling;

3. **development**

   a) writing of the sources and test codes;

   b) analysis and creation of reports with the statistics of the projects;

   c) meetings on the work's progress (Status meeting);

4. **integration**

   a) integration of the software components and end-to-end testing;

   b) optimization;

   c) codes' completion;

5. **deployment**

   a) completion of the operational documents;

   b) training;

   c) deployment on the production platform;

6. **post-deployment**

   a) controlling and monitoring on the released product;

   b) project's closure;

   a) celebration of the project's success with all the involved resources.

These six phases represent a necessary (but not always sufficient) condition for the success of a project. Any attempt to skip or reduce the time necessary for the correct execution of one of the phases generally turns out to be a fatal error, leading to unreliable results or sometimes even to the failure of the project. The most critical of these phases is the Post-Deployment, which is fundamental especially considering that any project mostly never generates an outcome that has no defects, nor imperfections. Therefore, at competition of every project, customers need subsequent technical and managerial support. In short, in order to ensure customer satisfaction it is always necessary to foresee and plan a post-deployment phase.

A good practice is to always document the life cycle of a project and the selected method even before the project starts. In many companies, the life cycle of a project often reflects the same trend of other projects that preceded it and, therefore, the documenting work becomes a real business process.

4.5 Components of the project

In order to document and successfully carry out any corporate project, it is necessary to define what is needed to be accomplished. The Purpose is an abstract fundamental component of the project (e.g. the goal or goals to be reached through the implementation of the project). Resources, on the other hand, are the material component that allows the realization of a project. In general, a project consists of:

I. **objective** - it is the goal (or goals) that has to be achieved through the implementation of the project;

II. **resources** - human, organizational, information, technical, and technological resources are the tools that allow the realization of the project;

III. **constraints** - such as the budget and time that are available for the project's implementation;

IV. **organization** - it is the system that (by assigning roles and responsibilities) facilitates a proper distribution of the available resources according to the given constraints;

V. **information** - it can be a resource, a potential project goal, or an output;

VI. **activities** - they are the elements into which a project can be broken down;

VII. **phases** - they are the homogeneous parts of the project and ensure that it is organized as an aggregation of similar activities;

VIII. **responsibilities** - definition and attribution of responsibilities to the human resources;

IX. **planning** - it defines the start/end dates of each activity, of the singular phases, and of the entire project. It also concerns the allocation of resources to the activities, the definition of the interdependencies between activities and phases of the project, as well as the milestones or the progresses of the project happening at the occurrence of a given event;

X. **management** - it consists in the coordination and development of the assigned resources. The result obtained through planning is a fundamental input for the management activity;

XI. **monitor and control** - they are the set of tools and procedures that certify the progress of the project. In particular, they are mainly based on complying with the time and cost constraints, as well as with partial objectives and milestones.

The above mentioned items are schematizations of a more complex reality. By simply recalling the importance in the projects' development of a tool such as the Deming cycle, it can be demonstrated, for example, how planning, management, and control are not as hermetic phases as it would seem from the above schematization.

4.6 The enhancement of the project

The valorisation of the project is greater if one invests in the initial phases of the Project life cycle, intervening mainly in the planning phase. In the table below, the guidelines to assist the Project Manager (PM) in identifying the key activities:

Component	**Notes/Definition**
Stakeholders' identification	Person, group, or organization actively involved in the project
Definition of roles and responsibilities within the team	Supplier, Customer, Project Manager, Team
Budget definition	Investment allocated for the purpose
Definition of the scope of action of the project	What needs to be implemented
Collection of the business requirements (requested by Stakeholders)	Condition/ability that has to be achieved or possessed by a system in order to satisfy a formally defined specification. Business requirements are divided into process and product requirements.
Creation of the Work Breakdown Structure (WBS)	Methodology for planning and hierarchical breaking down the planned work
Identification of the activities necessary for the production of the output	• Definition of the correct sequence of activities • Estimation of the costs required to procure the necessary resources • Time estimate for each activity
Comparison between expected and estimated deadlines	They are set based on the various commitments and become important moments of verification about the ability to deliver of the organization
Define costs considering the budget	Costs are forecasted based on the requirements
Plan for Quality	• Consistency and adequacy of outputs • Ability of the project to produce the milestones necessary to achieve the objectives
Carry out a Risk Assessment	It evaluates the effects, both positive and negative, that uncertainty has on the objectives. It is divided into product risk and process risk
Plan training/communication	Through brainstorming it is possible to receive ideas/requirements from stakeholders in a short time and at low costs

Component	Notes/Definition
Define the monitoring and evaluation indicators	It can be done using information-gathering techniques and through processes that allow the analysis of such information

Table 4.1: Core activities of a project

4.7 Planning Tools - Large projects

The basic operations of the PM, that are included in the previous table, aim to identify the fundamental elements of the project, such as: the activities to be carried out, the definition and allocation of resources, the time-planning, the control methods, and so on. Having all these elements clear is essential - among others - for the financial management of the project and for reporting purposes. For each of these elements of the project, tools have been identified that allow to schematically summarize the data of each activity so that it can be readily visualized, implemented, and corrected.

- Activities (what is planned to be done) →WBS
- Assignment of responsibilities (who is responsible for what) →OBS
- Resource definition (who will do it) →RAM
- Planning/scheduling (when it will be done) →GANTT
- Resource allocation (how much it will cost) →CBS
- Control method (what tests will be done) →Control plan

In the next paragraphs each of these tools are explained in detail.

4.7.1 WBS – Work Breakdown Structure

The WBS is a hierarchical breakdown of the project into its main phases or constituent elements and actions, with the aim to improve management and control. Such breakdown goes down to the elementary activities of the project, as each lower level represents an increasingly detailed definition of the project's components. The degree of detail of each branch of the WBS depends on the level of control that wants to be implemented.

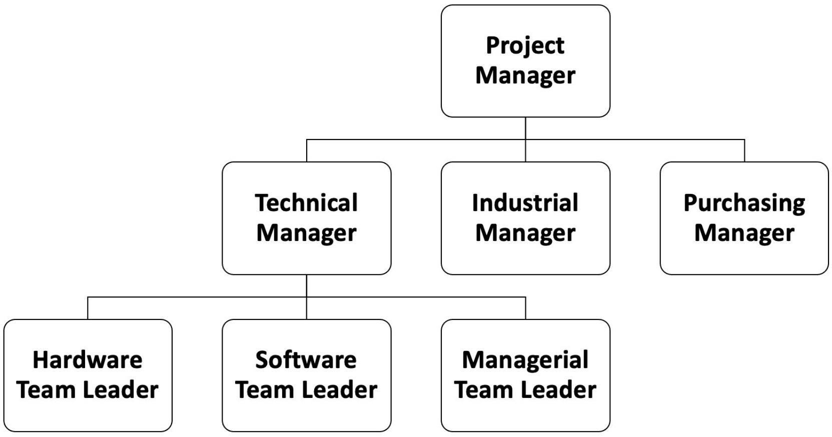


Figure 4.2: WBS

A WBS can be elaborated categorizing the activities in hierarchically-ordered levels:

- LEV.1 - project;
- LEV.2 - sub-projects or outputs into which the project is broken down;
- LEV.3 - work package [2] related to well-defined working and execution groups;
- LEV.4 - fundamental planning unit.

[2] Refer to the definition in the Glossary

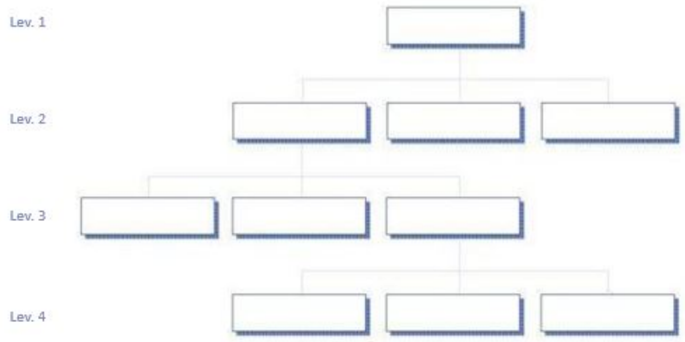


Figure 4.3: Levels of the WBS

The following image represents an example of WBS which shows the project activities to be carried out by a payment service provider, according to Regulation 847/2015 [3]:

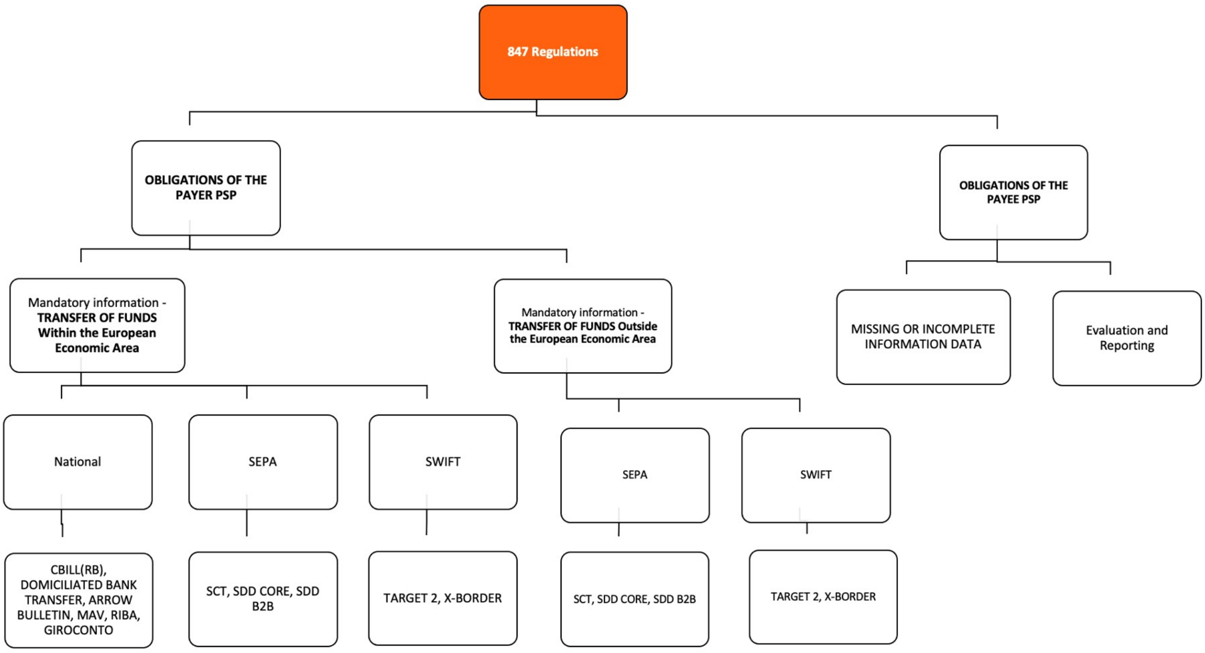

847 Regulations
OBLIGATIONS OF THE PAYER PSP
OBLIGATIONS OF THE PAYEE PSP
Mandatory information - TRANSFER OF FUNDS Within the European Economic Area
Mandatory information - TRANSFER OF FUNDS Outside the European Economic Area
MISSING OR INCOMPLETE INFORMATION DATA
Evaluation and Reporting
National
SEPA
SWIFT
SEPA
SWIFT
CBILL(RB), DOMICILIATED BANK TRANSFER, ARROW BULLETIN, MAV, RIBA, GIROCONTO
SCT, SDD CORE, SDD B2B
TARGET 2, X-BORDER
SCT, SDD CORE, SDD B2B
TARGET 2, X-BORDER

Figure 4.4: WBS Regulation 847/2015

[3] Regulation (EU) 2015/847 of the European Parliament and of the Council of 20 May 2015 concerning the information that accompanying transfers of funds and repealing Regulation (EC) no. 1781/2006

4.7.2 OBS - Organizational Breakdown Structure

After having identified the activities in which the project is divided, it is possible to move on to the phase of responsibilities' assignment. Responsibilities can be entrusted to people belonging to the organizational structure of the entity or to external consultants and may concern either the actual realization of one or more activities, or the coordination of an entire group of different activities. The result of this work is an organizational project structure that identifies functional responsibilities.

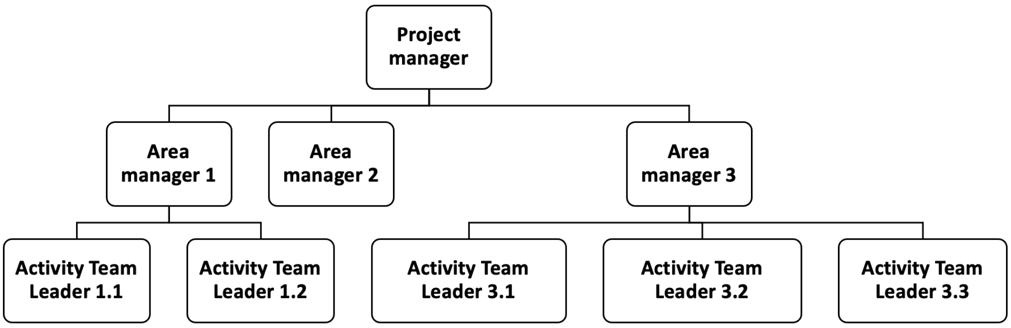


Figure 4.5: OBS

The OBS allows to:

- ripartire in maniera strutturata i livelli di responsabilità all'interno del progetto;
- stabilire il processo decisionale e le modalità con cui esso dovrà svolgersi all'interno del progetto, indicando come i vari ruoli dovranno integrarsi tra loro e quali sono i livelli di escalation a fronte di scelte e problemi operativi;
- integrare le informazioni tempi/costi secondo la struttura di responsabilità definita.

An OBS can be elaborated by hierarchically ordering the project managers:

- LEV.1 – Project Manager;

- LEV.2 – sub-projects manager;
- LEV.3 – work package/function manager;
- LEV.4 – simple resource.

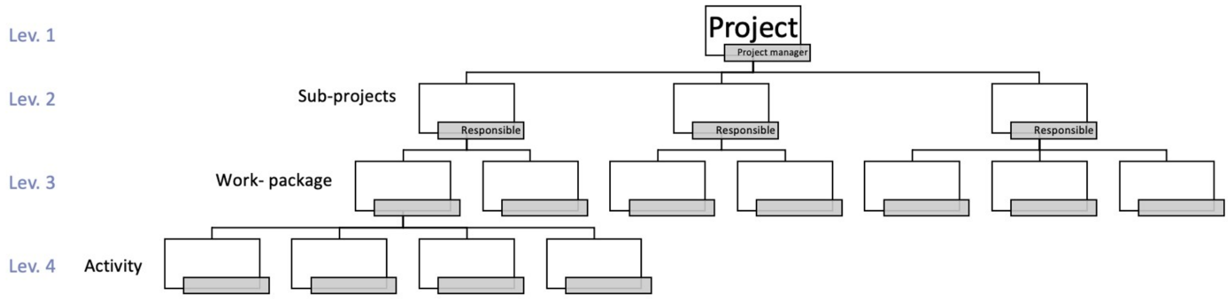


Figure 4.6: Levels of the OBS

The following image represents an example of OBS showing the project organization chart:

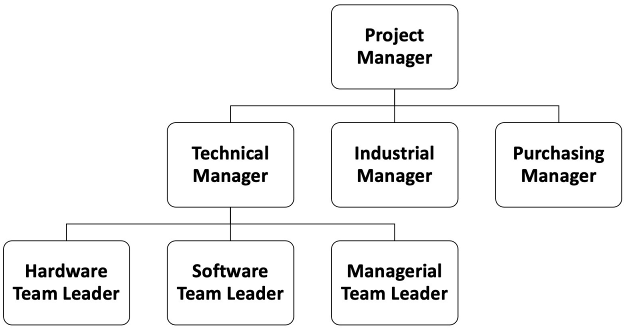


Figure 4.7: Example of the OBS

4.7.3 RAM – Responsability Assignments Matrix

The Responsibility Assignments Matrix (or Activity/Responsibility Matrix) is the graphic tool that allows to highlight the correlation between the elements of the WBS and those of the OBS.
The activity/responsibility matrix:

- relates the work packages defined by the WBS with the organizational units or the managers;
- helps identify the type/level of responsibility of people involved in the project based on their skills and their willingness to participate in the project.

Generally the matrix shows:

- in the columns, the components of the WBS (activities and tasks);
- in the rows, the human resources, the titles, or the functions (OBS);
- some symbols at the intersection of rows and columns that indicate ”who does what” and the level of responsibility.

The activity/responsibility matrix:

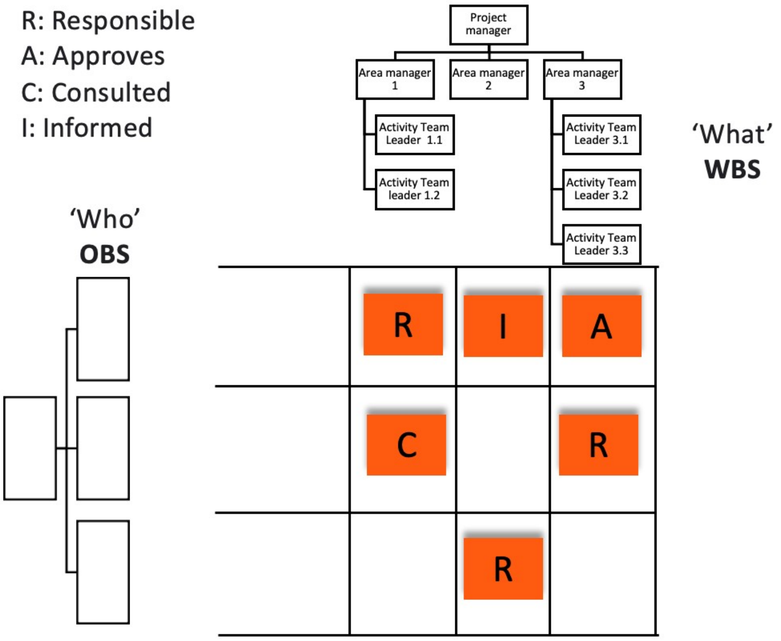


Figure 4.8: RACI Matrix

The following RAM represents the activities and the roles of responsibility for the project: Gala Dinner

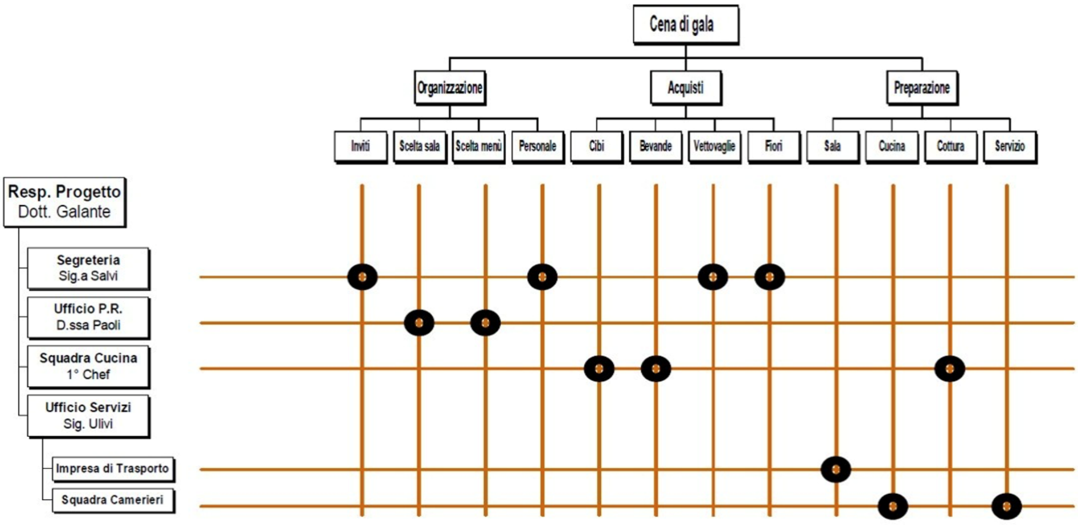


Figure 4.9: RAM (an italian Example)

4.7.4 GANTT – Henry Lawrence Gantt's chart

The Gantt chart is the official tool to manage the time planning of a project. It shows the timing of the activities planned within the project. The purpose of such representation is:

- to highlight the expected start and end dates of the project as well as its total duration;
- to highlight the activities that are part of the project and which of these activities are parallel to each other;
- to highlight the start and end dates of the activities, giving an estimate of the duration of each individual task;
- to highlight the critical path [4];

[4] In long and complex projects, in case of activities that have various types of constraints and that require continuous checks,

- to define a reference for checking the work progress;
- to have an indication about costs as a function of time.

In order to build of a GANTT chart, the following procedure is suggested [5]

I. **STEP 1** - list all the activities, showing the start and end dates on the left side of the graph and identifying the dependencies and the degree of priority of each activity;

II. **STEP 2** - at the top of the graph, create a project timeline as the sum of the time needed to perform each activity using a time scale suitable for the project;

III. **STEP 3** - since each activity is represented by a bar, the bars' positions and lengths reflect the start date, duration, and end date of each activity.

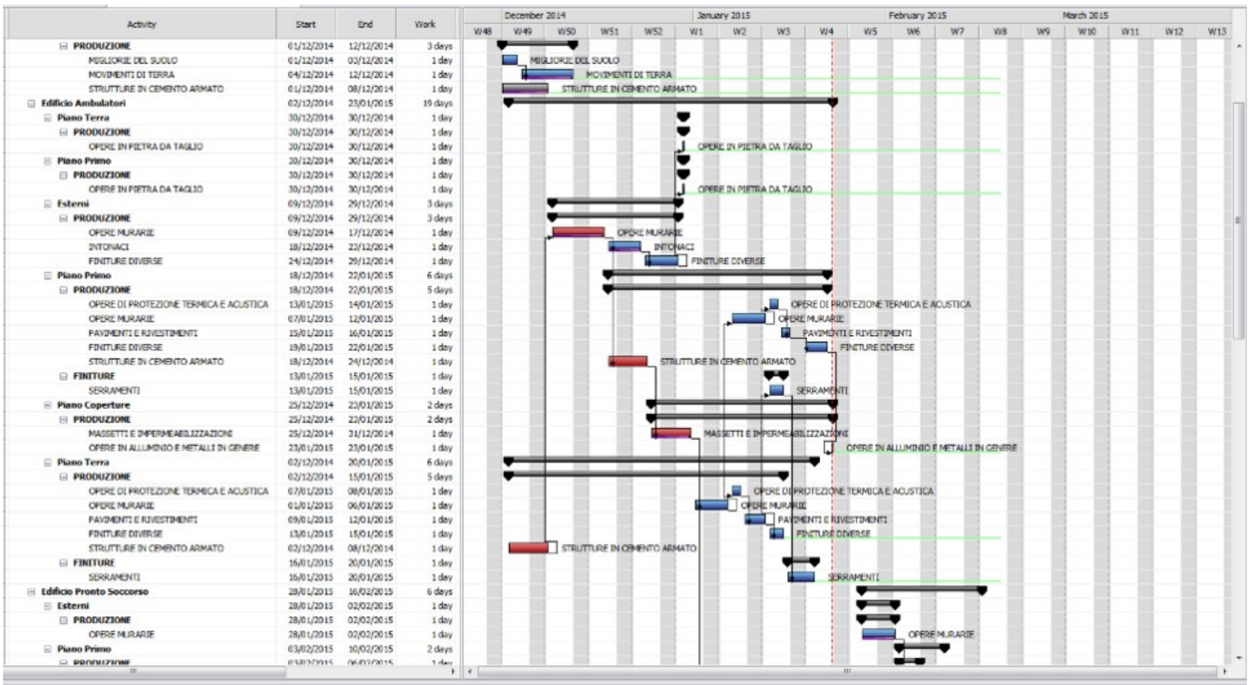
The following figure shows the activities, together with their overlaps and correlations, the timings, and the responsibilities for the project consisting in a hospital's construction:

the reticular planning is preferable, with a work timing indicated according to logical links of technical priority.

[5] **Reticular techniques:**

- **CPM** – Crititcal Path Method (metodo del cammino critico). È una tecnica di programmazione del progetto e viene utilizzato quando le attività che compongono il grafo sono già state svolte in altri progetti, quindi si conoscono le durate delle stesse. Permette di determinare il percorso critico mediante il quale si scoprono le attività in cui non devono assolutamente avvenire ritardi.
- **PERT** – Program Evaluation and Review Technique (utilizza una rappresentazione AOA - Activity on Arrow. In sostanza, mentre nel CPM le attività sono descritte in nodi rettangolari collegati da frecce che descrivono la durata, nel PERT le attività sono descritte da frecce che collegano nodi circolari che ne rappresentano la durata tecnica per valutare e controllare un progetto).

Note:the GANTT is also used in the control phase to distinguish the actual from the expected deviations, to evaluate their impact on the project, and to elaborate the corrective actions.



Activity	Start	End	Work
⊟ **PRODUZIONE**	01/12/2014	12/12/2014	3 days
MIGLIORIE DEL SUOLO	01/12/2014	03/12/2014	1 day
MOVIMENTI DI TERRA	04/12/2014	12/12/2014	1 day
STRUTTURE IN CEMENTO ARMATO	01/12/2014	08/12/2014	1 day
⊟ **Edificio Ambulatori**	02/12/2014	23/01/2015	19 days
⊟ **Piano Terra**	30/12/2014	30/12/2014	1 day
⊟ **PRODUZIONE**	30/12/2014	30/12/2014	1 day
OPERE IN PIETRA DA TAGLIO	30/12/2014	30/12/2014	1 day
⊟ **Piano Primo**	30/12/2014	30/12/2014	1 day
⊟ **PRODUZIONE**	30/12/2014	30/12/2014	1 day
OPERE IN PIETRA DA TAGLIO	30/12/2014	30/12/2014	1 day
⊟ **Esterni**	09/12/2014	29/12/2014	3 days
⊟ **PRODUZIONE**	09/12/2014	29/12/2014	3 days
OPERE MURARIE	09/12/2014	17/12/2014	1 day
INTONACI	18/12/2014	23/12/2014	1 day
FINITURE DIVERSE	24/12/2014	29/12/2014	1 day
⊟ **Piano Primo**	18/12/2014	22/01/2015	6 days
⊟ **PRODUZIONE**	18/12/2014	22/01/2015	5 days
OPERE DI PROTEZIONE TERMICA E ACUSTICA	13/01/2015	14/01/2015	1 day
OPERE MURARIE	07/01/2015	12/01/2015	1 day
PAVIMENTI E RIVESTIMENTI	15/01/2015	16/01/2015	1 day
FINITURE DIVERSE	19/01/2015	22/01/2015	1 day
STRUTTURE IN CEMENTO ARMATO	18/12/2014	24/12/2014	1 day
⊟ **FINITURE**	13/01/2015	15/01/2015	1 day
SERRAMENTI	13/01/2015	15/01/2015	1 day
⊟ **Piano Coperture**	25/12/2014	23/01/2015	2 days
⊟ **PRODUZIONE**	25/12/2014	23/01/2015	2 days
MASSETTI E IMPERMEABILIZZAZIONI	25/12/2014	31/12/2014	1 day
OPERE IN ALLUMINIO E METALLI IN GENERE	23/01/2015	23/01/2015	1 day
⊟ **Piano Terra**	02/12/2014	20/01/2015	6 days
⊟ **PRODUZIONE**	02/12/2014	15/01/2015	5 days
OPERE DI PROTEZIONE TERMICA E ACUSTICA	07/01/2015	08/01/2015	1 day
OPERE MURARIE	01/01/2015	06/01/2015	1 day
PAVIMENTI E RIVESTIMENTI	09/01/2015	12/01/2015	1 day
FINITURE DIVERSE	13/01/2015	15/01/2015	1 day
STRUTTURE IN CEMENTO ARMATO	02/12/2014	08/12/2014	1 day
⊟ **FINITURE**	16/01/2015	20/01/2015	1 day
SERRAMENTI	16/01/2015	20/01/2015	1 day
⊟ **Edificio Pronto Soccorso**	28/01/2015	16/02/2015	6 days
⊟ **Esterni**	28/01/2015	02/02/2015	1 day
⊟ **PRODUZIONE**	28/01/2015	02/02/2015	1 day
OPERE MURARIE	28/01/2015	02/02/2015	1 day
⊟ **Piano Primo**	03/02/2015	10/02/2015	2 days
⊟ **PRODUZIONE**	03/02/2015	06/02/2015	1 day

Figure 4.10: GANTT (an italian Example)

4.7.5 CBS - Cost Breakdown Structure

The CBS defines the project costs by assigning each unit cost to some work packages. It is useful to highlight the total cost of the project, which is given by the sum of the direct costs (sum of the costs of all project activities) and the indirect costs (sum of the costs that are 'transversal' to the activities, like those for secretariat, administration, headquarters, etc.). Aggregating the estimated costs for individual activities or for the work packages allows to define a baseline of authorized costs. The cost baseline is an authorized, timed budget that considers all the forecasted costs of the project with the sole exception of the managerial reserves. The cost baseline is an absolute reference used to measure, monitor, and control the project costs.

In the project budget, all estimated costs must be considered, with the exception of managerial reserves, which are different from the contingency reserves and which, for this reason, are often also referred to as extra budget resources. In fact:

- the contingency reserves are provisions necessary to deal with unplanned but essential changes and thus they are needed to mitigate the occurrence of one or more risks;
- the managerial reserves are extra resources reserved for unplanned changes within the scope of the project. Often the Project Manager cannot decide independently on the use of these resources and the approval from external actors is required.

It is possible to build a CBS by following the steps:

1. **STEP 1** – evaluate the cost of each individual activity;
2. **STEP 2** – re-aggregate the costs by going up through the WBS;
3. **STEP 3** – the sum of the total costs of the activities is the direct cost of the project to which indirect costs must be added.

The calculations of the cost of the resources already spent or still to be spent must take into account:

- the resources for use (e.g. labor and equipment that are used and reused during the same project). They can be calculated as follows: [(RESOURCE COST/UNIT of TIME) x (DURATION OF ACTIVITY)];
- the resources for consumption (the resources that cannot be used a second time, like printer paper, electricity, fuel for a company car, etc.) They can be calculated as follows: [(RESOURCE UNIT COST) x (NUMBER OF RESOURCES).

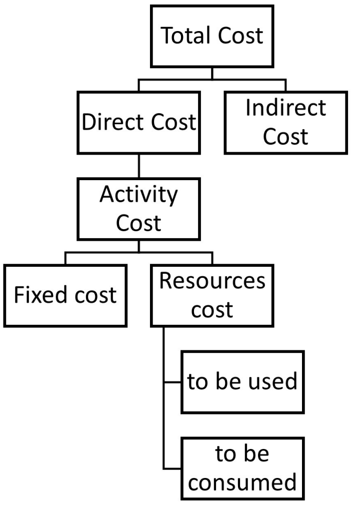


Figure 4.11: CBS

4.7.6 RBS - Resource Breakdown Structure

The RBS is a tool that identifies which and how many resources are needed for the project, according to the nature and the quantity of the product to be created. It identifies the personnel resources necessary for the successful completion of the project and all the other resources on which the project funds will be spent, including tools, machinery, materials, equipment, fees, and licenses.
The representation of the RBS diagram shows a hierarchical breakdown of the resources, ordered by category and type. It can be designed by the following steps:

1. **STEP 1** – assign the necessary resources to each WP (Work package):

   a. by type (what);

   b. by quantity (how many).

2. **STEP 2** – distinguish between human and material resources;

3. **STEP 3** – for human resources it is necessary:

   a. to identify the different types of needed skills (highlighted in the OBS in relation to the assigned responsibilities);

  b. to understand if they are available within the company;

  c. to estimate the amount of work (time/worker) needed.

4. **STEP 4** – for material resources it is necessary to distinguish between:

   a. logistical resources (offices, meeting rooms, conference rooms, etc.);

   b. equipment (PCs, workstations, projectors, etc.);

   c. materials (consumables, raw materials, etc.).

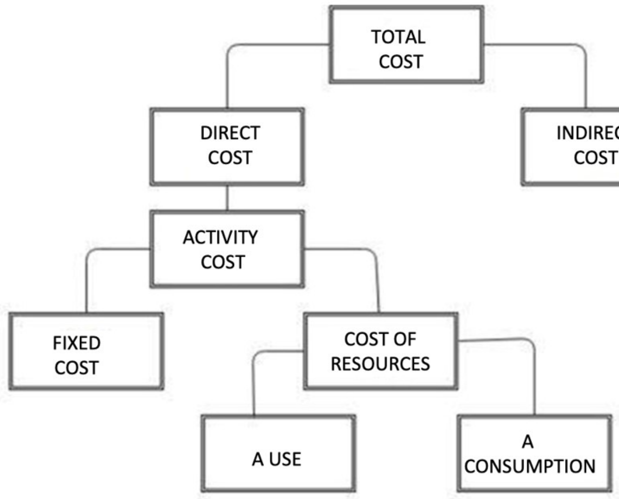


Figure 4.12: RBS

Part II

Section 2

Chapter 5

Project Management

In order to win nowadays' competitive challenges, new managerial skills and tools are needed to manage change: innovation should be handled within the company as a weapon for competition. Project management is a tool for managing change and innovation.

Project Management (PM) is defined as the process of managing a project from start to end, through the allocation, use, and monitoring of (limited) resources in order to meet the requirements (of the project) over a well-defined period of time. Project Management guides the behavior of all members of the organization; as the objectives of the project highlight the strategic trajectory of the company, and as such strategies indicate how to achieve those objectives.

The PM requires a mix of managerial and organizational skills directly proportional to the size and complexity of the project. Such skills, if being used through a formal and structured process, can lead to a successful finalization of the project. Project Management cannot and shall not deal with specific technical skills. Project Management's fundamental components are:

- problem solving orientation;
- measurability of interventions;
- methodological rigor;

- work in cross-functional groups;
- propensity for change;
- management of cross-functional processes.

5.1 Inter-functional communication

Inter-functional communication is seen as a team-working development lever: correct communication between the various functions and roles held by people in the company generates effective collaboration among the various company Actors.

Encouraging a team-working approach, as well as the birth and growth of new work teams in the organization can lead to better and more effective performances both within single projects and, more in general, within entire company functions.

All team members should be aware of the activities, tasks, products, and progress linked with their work, as well as to the objectives to which they actively contribute.

Project managers are at the heart of cross-functional communication and their ability to communicate efficiently is just as important as their technical competence.

To be effective, cross-functional communication has to include three axes:

- **top-down** → the project resources should always be involved and periodically motivated, through mailing-lists and messages that announce the work progress and the achieved goals. The resources should also be encouraged to immediately inform the project manager of any problems or inconveniences and any implementation ideas that may differ from the ones established in the project plan.
- **bottom-up** → to ensure the participation of all staff in the process of change embodied in every project, it is fundamental to communicate objectives and progress, as well as benefits that can be obtained over time.
- **middle-out** → it is important to provide support at every level, thus communication should be extended to all project resources.

Communication takes place in various ways: through closed meetings, group meetings, email, documents exchange, and so on. These communication methods are not interchangeable, as each of them has its own unique characteristics that can be more or less useful depending on the specific case.

5.2 Project Management Methods

The choice of the strategy, among the many possible (in some cases overlapping) approaches, that can be used for a successful management of the planning and implementation phases of a business project, is a consequence of the adoption of a specific project management method that should take into account the complexities of the project, the organizational priorities, and the risks.

The following paragraphs show the most common Project Management Methodos (PMM) and the criteria for evaluating the best solution for specific projects and companies. Once the choice has been made, the selection process can be documented and repeated also for future projects, allowing the company to spend less time discussing how to structure and manage a project and more time to focus on its objectives.

5.2.1 Waterfall

The Waterfall has been one of the main project management methods for several years. Its approach is sequential in nature and is used in many industries, most commonly for software development. It includes a series of static steps (requirements analysis, design, testing, implementation, and maintenance) to be performed in a specific order. By providing a very-detailed, formal planning phase, this method aims to identify all project requirements in advance and reduce the loss of key information. Although “Waterfall” allows for greater control during each phase, it can be too rigid a paradigm in the case of project’s variations.

The use of this method is desirable for projects that have clear requirements with no expected changes. On the other hand, it is not recommended when the requirements are not well defined and when there is a high risk of changes in progress.

The drafting phases of a Waterfall project can be listed as follows:

1. definition of the requirements;
2. design & development;
3. test;
4. implementation;
5. maintenance phase.

This project management methodology is used in many sectors, most commonly in software development.

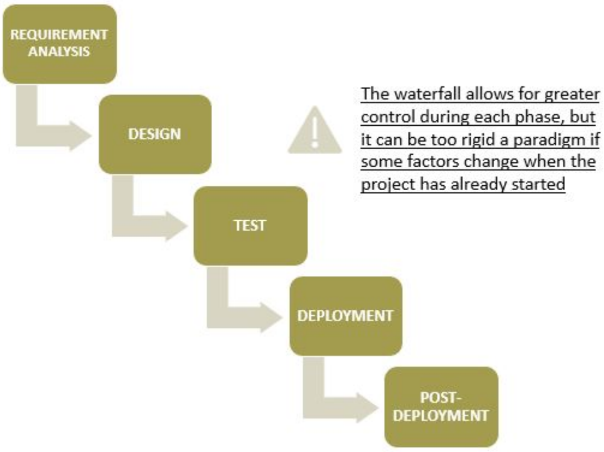

REQUIREMENT ANALYSIS
DESIGN
TEST
DEPLOYMENT
POST-DEPLOYMENT
The waterfall allows for greater control during each phase, but it can be too rigid a paradigm if some factors change when the project has already started

Figure 5.1: 'Waterfall' method

5.2.2 Agile

The Agile method takes a significantly different approach than the Waterfall method. Developed for projects that require considerable flexibility and speed, the Agile method consists of short delivery cycles, called "sprints".

Such a method facilitates quick changes during the course of a project: the team tries to do as much as possible in a given amount of time before moving on to the next stage of the project. Agile is a highly interactive project management method that allows for quick changes over the course of a project. It is commonly used in software development projects because it allows you to quickly identify problems and make changes early in the development process, rather than having to wait until the test is complete. This method offers repeatable processes, immediate feedback, and fast response time, while reducing risk and complexity.

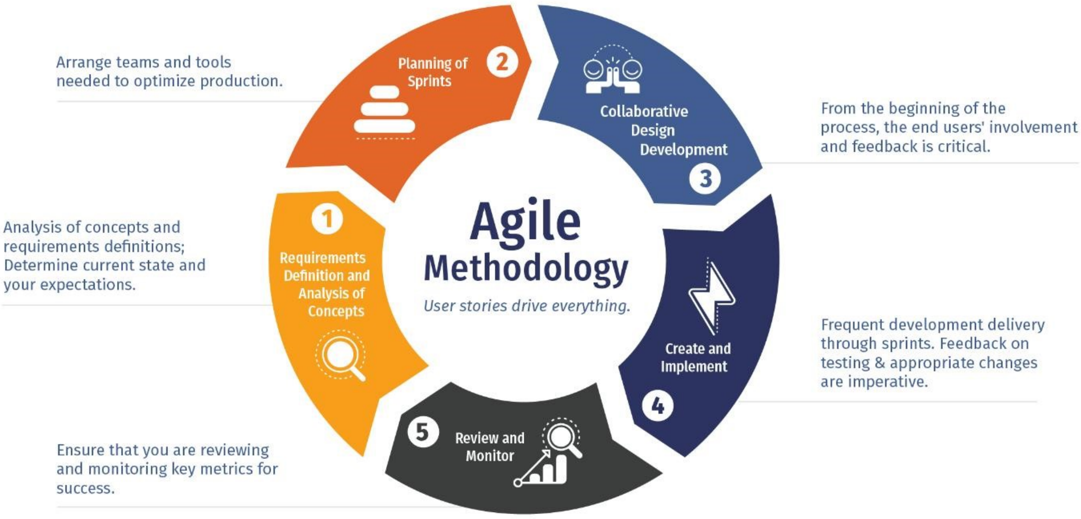

Agile
Methodology
User stories drive everything.
1 Requirements Definition and Analysis of Concepts
Analysis of concepts and requirements definitions; Determine current state and your expectations.
2 Planning of Sprints
Arrange teams and tools needed to optimize production.
3 Collaborative Design Development
From the beginning of the process, the end users' involvement and feedback is critical.
4 Create and Implement
Frequent development delivery through sprints. Feedback on testing & appropriate changes are imperative.
5 Review and Monitor
Ensure that you are reviewing and monitoring key metrics for success.

Figure 5.2: Agile method

The use of the Agile method is suitable for projects where rapid and iterative-construction goals are to be achieved. On the other hand, it is not suitable for large projects and for static/traditional environments

5.2.3 Scrum

”Scrum” is a term borrowed from the world of rugby. The Scrum method provides an interactive approach and is part of the Agile framework. “Scrum sessions” or “30-day

sprints" are used to determine the priority activities. The figure of the project manager is replaced by a supervisor who interacts with small teams that work independently on specific tasks.
The Scrum method has the following steps:

1. at the beginning of each Sprint, the team selects its **Tasks** from a list of activities scheduled by priority, with the aim of completing all the selected tasks by the end of the **Sprint**;

2. team members discuss on a regular basis during the **Daily Stand-up** or Daily Scrum Meeting (typically at the beginning of each working day);

3. the sprints are implemented as **development cycles**;

4. at the end of each Sprint, the Team releases the output that has been completed (**Done**), which should always meet the requirements.

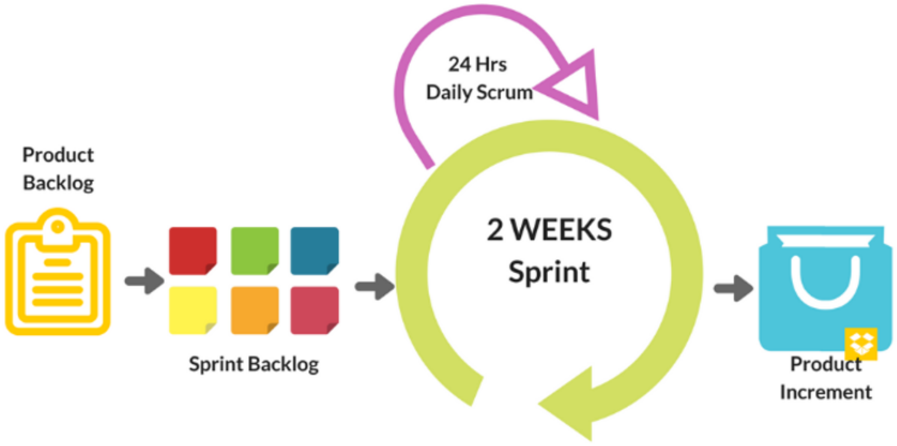


Figure 5.3: SCRUM method

This project management method is used in many industries, most commonly in product development. It is particularly useful in projects where continuous improvement of the performance or of the product characteristics is required; for this very reason it is not suitable for static projects where changes in requirements are not allowed.

5.2.4 Hybrid

The Hybrid project development method combines the advantages of the Waterfall and Agile approaches. With such a method, the planning and requirements-gathering phases are executed with the Waterfall approach, while the design, development, implementation, and evaluation phases use the Agile criterion.
The Hybrid method can be implemented through a simultaneous use of the Waterfall and Agile approaches or through a mix of the traditional and Agile methods, within the same project. In order to successfully use the "hybrid" approach, wide knowledge of both approaches is required, with awareness that an excessive mix of different methods may not automatically lead to an improvement in the project performance.

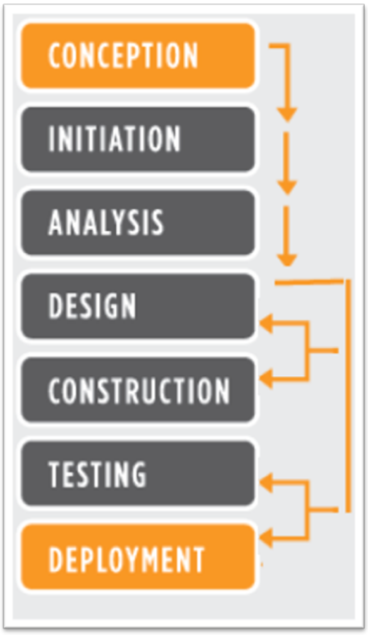


Figure 5.4: Hybrid method

5.2.5 Critical Path Method (CPM)

The Critical Path Method (CPM) is a step-by-step strategy used for projects with interdependent activities. It contains a list of tasks and it uses a work breakdown structure (WBS), a timeline to complete, dependencies, key points, and deliverables. It establishes which activities are critical and which are not by calculating the "longest" and "shortest" (float) time available on the critical path[1].

The CPM involves the following steps:

1. define the project WBS;
2. identify the project activities based on each work package;
3. use an estimation technique to evaluate the duration of each activity;
4. define the dependencies between the activities and the set of dependencies that will construct the logical network;
5. highlight the paths within the network from the beginning to the end of the project.

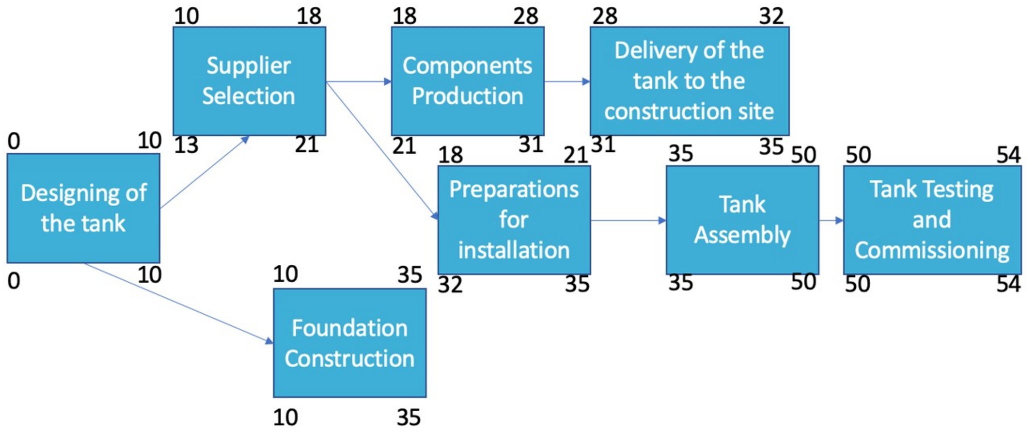


Figure 5.5: CPM method

[1] If the paths are few, it will be easy to identify the critical path (the one with the longest duration) otherwise it will be necessary to calculate the CPM.

5.2.6 Six Sigma

The Six Sigma method allows to remove the causes that generate defects of the process. This approach is aimed at customer satisfaction and is systematically based on the collection and analysis of data to reduce defects.

This project management method is particularly applicable to manufacturing companies and to the financial sector offering savings through reducing reworks and bottlenecks. Conversely, this approach may not always produce the desired result or it may lengthen production times.

Here are some techniques and tools used by this method to achieve the project objectives:

- DMAIC (for the continuous improvement of processes)

    1. define
    2. measure
    3. analyze
    4. improve
    5. control

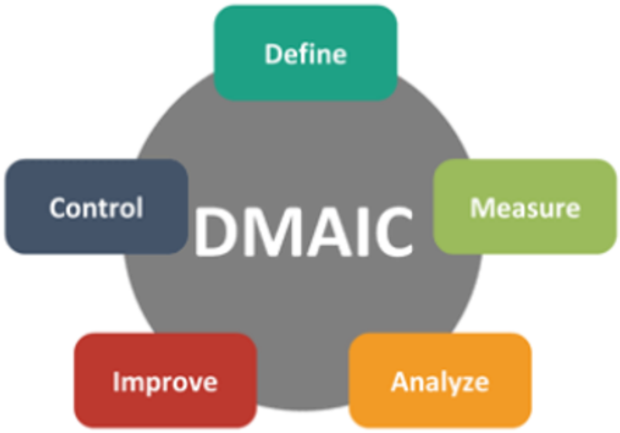


Figure 5.6: DMAIC method

- DMADV (for the redesign and optimization of processes)

  1. Define
  2. Measure
  3. Analyze
  4. Design
  5. Validate

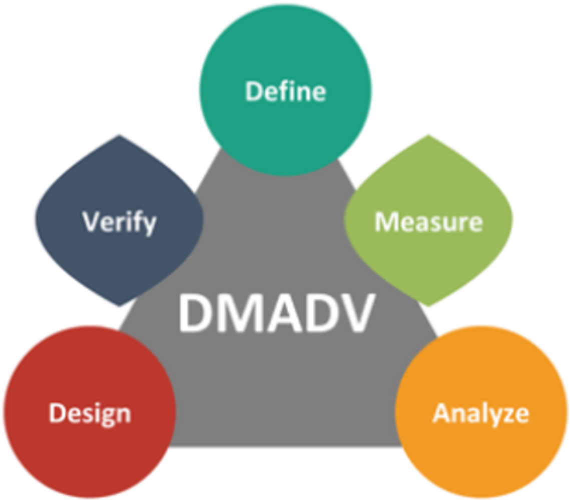


Figure 5.7: DMADV method

5.2.7 DevOps

DevOps can be considered as an innovation of the Agile method, generated over time by the close collaboration between two key company figures that are usually crucial for the project development:

- the developer - this company actor is mainly oriented to the creation of changes and to the addition or modification of functionalities;
- the operation - this type of worker pushes towards stability and improvement of services, checking the functioning and compliance of the new releases.

The DevOps method fosters the collaboration between these two areas in all phases of the product life cycle, aiming at generating value in terms of quality of the products and speed in their realization.

This method seeks to achieve business objectives through a set of practices, culture, and values that guide the change towards a more solid software/functionality capable of reaching the first phase of production with as few imperfections as possible.

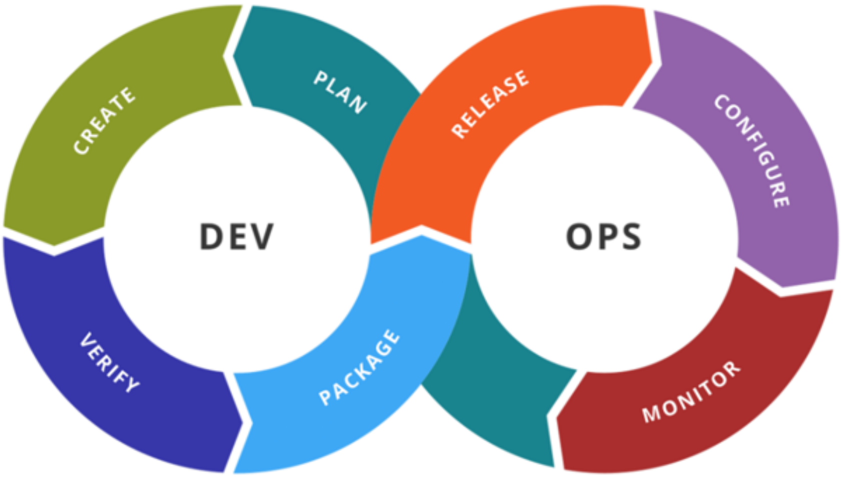


Figure 5.8: DevOps method

Chapter 6

Risk and probability of occurrence

By risk we mean a condition that leads the project towards a situation of crisis that may also be insurmountable. In general, risks should be identified and managed before they turn into a crisis.

As a rule, with the aim of minimizing the likelihood that a risk occurs and/or that the impact of the risk is decisive for the project, it may be necessary to modify the action plan. Risk management requires an organizational structure capable of carrying out systematic checks, needed to confirm the actual feasibility of the project objectives.

All projects have risks; however, the greater the innovative solutions included within the project, the greater the likelihood of the risk occurring.

The risks may be due to various causes:

- application of a new technology;
- lack of company support for the project;
- size and complexity of the project that may require greater coordination than expected;
- insufficient real market demand for the proposed products;

- market changes.

Some of the above risks are internal to the project, that is, they depend on the correct planning and implementation of the activities; other risks, on the other hand, are external to the project, that is, they depend on elements of the external context affecting the 'life of the organization' (new regulations, market changes, etc.).

The actions that can be implemented for the identification, definition, and management of risks are:

- structuring and planning of the project;
- good communication and interpersonal skills of the project manager;
- identification and classification of risks according to an order of priority.

The priorities of intervention depend on the severity associated with the risk, that is, the probability of occurrence of the risk and its impact on the outcome of the project. Risks with high probability and high impact must be addressed and resolved, while those with low probability or limited impact may be also accepted and suffered.

Over time, numerous risk management methods have been designed but none of them can guarantee certain results.

Chapter 7

Risk Management

Before addressing the subject of Risk Management, the reader should first be aware that projects imply innovation and that any innovation always implies a "dose" of risk. Managing project risk means identifying the causes and assessing the factors of uncertainty in a preventive manner.

Risk management can take place in five different ways:

- prevention (intervening to prevent an event from occurring);
- reduction (intervening to reduce the probability and/or severity of the risk);
- transfer (implementing measures that transfer the project risk to other subjects or situations);
- contingency (preparing plans to be implemented only in the presence of a risk);
- acceptance (deciding to accept and live with the risk without further intervention).

The choice of a certain course of action in the face of a particular risk depends on the relative balance between the economic weight of the risk and the cost of management intervention; in many cases it is also possible to initiate a combination of several types of intervention simultaneously.

Risk management is not a stand-alone exercise carried out in the planning phase and then forgotten, but it is an activity that has to be systematically implemented as new information on the progress of the work becomes available.

In the planning phase, the project team, or the PM individually, must consider risk management through the following phases:

- identify the possible project risks and evaluate the probability that each risk occurs;
- analyze every single risk, identifying its probable causes;
- estimate the period within which it is possible to intervene to mitigate the risk;
- determine the impact in case of occurrence of the risk;
- plan responses in case of negative impact, identifying types of intervention and containment actions;
- monitor and control.

To activate the process of identifying, evaluating, and prioritizing the causes (events or factors) that may adversely affect the achievement of the project objective, a distinction must be made between:

- **product risk** : it is a potential failure linked to the physical tools used to create an output, that decreases the quality of the product/service due to, for example:
  - softwares unable to perform the required functions within the established times,
  - inaccurate documentation,
  - poor data quality and integrity,
  - other;
- **project risk** : it is a potential failure linked to the 'cognitive abilities' used to achieve the project objectives, for example:

  - organizational factors (lack of skills),
  - technical factors (problems in defining the requirements),
  - factors related to the supplier (bankruptcy/contractual problems).

Once these two types of risk have been taken into account, risk management needs to be completed by planning the possible reactions and drafting a risk mitigation plan through the use of monitoring and control systems.

7.1 Risk management tools. The Fishbone Diagram

Among the tools used for developing project risk management plans, we find the Ishikawa Diagram (*cause/effect diagram or Fishbone Diagram*). This tool graphically traces the processes of a project according to an ideal flow, facilitating management and quality control (this scheme was created by chemical engineer Kaoru Ishikawa in 1943). The Fishbone Diagram is useful to:

- identify the possible causes of a problem;
- improve efficiency in detecting problems;
- improve the conditions for acting on problems.

Based on Ishikawa's theories, there are four standard causes that can negatively influence a project or its realization: Man, Machine, Method, and Material. To these four main variables (also called 4M), four more variables can be added (Measurement, Mother Nature, Management, Money) which are typical of the Six Sigma design processing method (the so-called 8M). In practice, the graph is built by drawing a central line (the bone of the fish) and writing, at the end of it, the effect to which you want to know the causes. Along the line indicate the 4M/8M (herringbones).

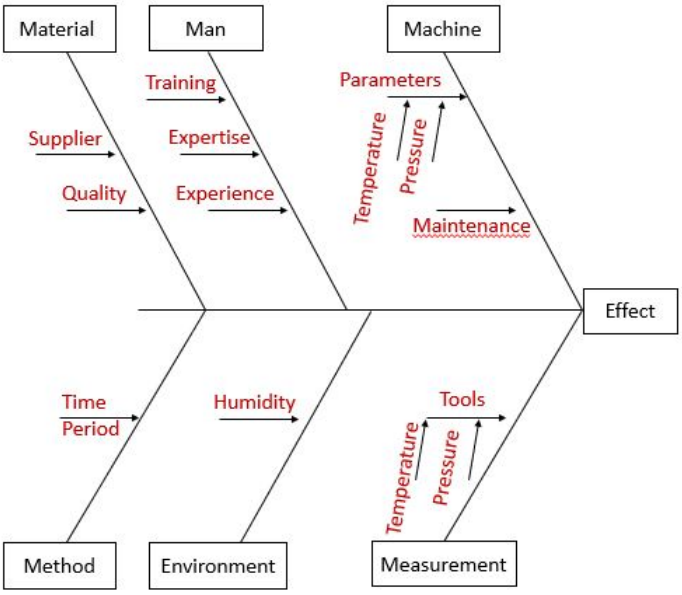


Figure 7.1: Fishbone diagram

After having collected the needed information and having placed each problem close to the "M" to which it belongs, it is then possible to trace the primary and secondary causes that generate the effect.

Chapter 8

The project roles

Although every organization uses different names and roles to indicate each of its professional figures and their respective competences, it is possible to elaborate a list of company people who cover specific roles within a project. In some cases, however, when the project is small in size some tasks/roles can be merged, while in more complex projects the same tasks are usually assigned between different people. In small projects a project manager is a multitasking figure, while in bigger projects he/she is often supported by a working group (PMO - Project Management Office), sometimes large in scale and with high skills and experience. In general, the following project roles can be identified:

- the sponsor;
- the program board;
- the project manager;
- the project support office;
- the quality manager;
- the head of administration;
- specialized external consultants;

- external suppliers;
- stakeholders;
- end users.

In the following paragraphs we will try to describe their activities and supporting tools.

8.1 The sponsor

The sponsor is usually a top manager, that is, an experienced manager who covers a role of responsibility within the company and who takes care of the project by focusing on the objectives from a business point of view. The sponsor is responsible for the project's outputs, but does not deal with the aspects related to the project management; in particular he/she deals with:

- maintaining relationships with suppliers and customers;
- ensuring that users' needs and expectations are met;
- ensuring that the risk is controlled;
- keeping the project in line with company policies and strategies;
- monitoring the relationship between costs and benefits of the project;
- evaluating possible variations of the project due to external factors;
- verifying compliance of the product quality with company standards.

The presence of a sponsor allows not to have to involve the entire management of the organization in every decision. Conceptually, the sponsor represents an intermediary subject who relates with the company on one side, guaranteeing the necessary funds and resources, and with the team on the other side, agreeing on the actual realization of the project. From an organizational point of view, the project manager relates to the project sponsor, who in turn relates to the company. Normally the contacts between project manager and sponsor are very limited and they occur at the conclusion of main events of the project or in risk situations that require decisions and interventions that go beyond the responsibility of the project manager. Ultimately, the sponsor represents a stakeholder who has the greatest interest in the planned initiative and who appoints the project manager that will carry out the work for the project's realization. The assignment of the role and the illustration of the project's contents take place through an assignment document, also called project charter.

The assignment document (project charter) is usually drafted by the sponsor, or by the PM based on the information provided by the sponsor. The document describes the goal to be achieved, the strategy to be implemented, and the responsibilities to be assigned. The drafting of the assignment document may not be instantaneous, as it is usually completed in all its sections throughout multiple meetings and analyzes, aimed at ensuring the best *commitment* of the parties involved.

The structure of the assignment document includes:

- problem statement;
- business case;
- goal statement;
- costs and benefits;
- temporal sequence;
- scope;
- team members.

8.2 The stakeholders

Stakeholders are any individuals or organizations actively involved in a project whose interests are, for various reasons, linked to the result of its execution.

The representative of the stakeholders (point of reference and coordination of the subjects directly or indirectly involved in a project or in the activity of a company) is responsible for his/her work, even before the project, towards the stakeholders. Stakeholders, depending on the type of project, can be internal or external to the company, and may include the following types of figures:

a) recipients of final outputs;

b) personnel belonging to other functions /divisions of the company whose daily work is affected by the project;

c) managers and team members of other projects depending on this project, from which they await outputs, or with which they compete for shared resources;

d) people or associations, unrelated to the company, who are interested in the objectives or consequences of the project; for example, if a new road is to be built, there may be owners of the land to be expropriated or inhabitants of nearby areas who may have special interests and oppose the project;

e) current customers who may react positively or negatively to the news regarding new developments;

f) suppliers and distributors who may fear economic losses due to the changes introduced by the project.

The role of these groups in the project needs to be addressed on a case-by-case basis. Projects can have effects outside their context and indirectly involve a wider audience than that of the final recipients. Sometimes project management overlooks these effects, only caring about the end-users and the organization of the project team, thus putting the success of the project at risk. Stakeholders' management is strongly linked to public relations' management. In the case of small projects it is usually not necessary to worry about external stakeholders; however it is good practice to remember that outside of the formal boundaries of the project there are many interested parties with whom it is important to communicate.

8.3 The Program Board

The program board is the management that represents the company and that has the ultimate power over the implementation and financing of the project. Most organizations, where multiple projects are running at the same time, have a dedicated board that meets regularly to oversee the projects portfolio. The program board has the following tasks:

- evaluate and approve project proposals;
- determinate the order of priority of the project proposals;
- allocate resources and authorize their use;
- monitor project risks and other risk situations;
- block the projects if needed, or promote and authorize corrective actions;
- optimize the use of resources between the various projects, identifying and possibly eliminating any overlaps.

8.4 The Project Manager – PM and the Project Initial Document (PID)

The PM has the task of planning, organizing, managing, monitoring, and reviewing all project activities. Also, in the case of minor projects he/she may carry out directly some of the activities. The project manager depends on the sponsor with whom he/she agrees to manage the project according to the procedures defined in the PID (Project Initial Document) and approved by the company management. In addition, he/she coordinates the team and organizes the work. The project manager generally has the authority to use and manage the budget, the human resources, and the materials within the limits established by the PID and therefore is usually an internal figure within the company. If the project fails to comply with the budget or time constraints, the project manager can request the company management (Programme Board) to increase the budget or the time available, justifying the needs. In these cases, only after the approval of the new fundings the revision of the plan can start and the project can proceed according to the new conditions. Finally, the PM has the task of drawing up project plans, identifying and managing risks, monitoring and documenting the progress of the project, and controlling the Change (CR - Change Request). The success of a project depends above all on the skills and work of the project manager. This figure, in fact, is the link between the sponsor and the project core-team (ideally placing itself between the funder and the executor), being able to simultaneously interact with both of them for the success of the project.

The Project Initial Document (commonly called PID) is a standard document needed for defining the project plan. It should always contain the essential information for the project, such as: the project budget, the realization timings, the details of human resources and structural resources necessary for the project, and so on. Usually the PID is the document that allows the company to approve, finance, and start the implementation of the project. It is structured as follows:

- project objectives;
- scope;
- organization of the project;
- business case;
- constraints;
- interested parties;
- estimated risks;
- project checks;
- reporting framework;

- PID sign off[1];
- summary.

The following figure shows a PID template:

STRUCTURE OF THE DOCUMENT
Objectives of the project
Scope
Organization of the project
Business case
Constraints
The interested parties
Risks
Project checks
Reporting framework
PID Sign Off
Summary

Table 8.1: Structure of a PID

8.5 The Project Management Office (PMO) and the Project Status Report (PSR)

In large companies, an office dedicated to project coordination and support (PMO) is often provided. This office carries out various activities, acting as a mediator between all the figures present within a project, such as:

a) secretarial and administrative activities, monitoring the use of company resources on the project in order to detect and compare the performed workload with the planned one;

[1] Approval process

b) technical activities such as collecting business requirements and informing users about ongoing initiatives as well as on any impacts of the project activities on their current work (through minutes and PSRs);

c) activities related to project needs, such as the recruitment of experts to assist on specific parts of the project;

d) several activities for the PM:

  I.) Spreading of the best Project Management practices within the company;

  II.) Organization of training courses; I

  III.) Definition of the standards for documentation.

e) various activities for the Program Board

  I.) production of Project Status Reports (PSR) on the progress of the project on behalf of project managers;

  II.) definition of a first order of priority of the project proposals presented by the various departments of the organization.

Among those listed, the most important activity is the drafting of the Project Status Reports (PSR). This document is a standard template used at the end of any project meeting (either formal meetings or more informal internal alignments meetings). It is essential to avoid the dispersion of important information that may be useful for the continuation of the project activities. The PSR should always be structured as follows:

I.) minutes title;

II.) summary table with the main information;

III.) table with approval status;

IV.) paragraphs:

  a) works' opening;

b) participants;

c) absent members;

d) agenda;

e) participants’ reports;

f) record of pending activities;

g) new activities resulting from the meeting.

8.6 The Quality Manager

The Quality Manager is the company figure that connects the management with all the other operational departments: he/she works on behalf of the sponsor and in parallel with the PM (without depending on the latter). This figure may not always be explicitly present within the projects, but he/she is the one who works to achieve corporate goals. In small companies, the duties of ”Quality system manager” and ”Quality control manager” are usually held by the same person. The Quality Manager should have a high analytical capability aimed at problem solving and at continuous improvement of processes, as well as proven ethical and reliability skills. He/she should also possess the following specific skills:

a) knowledge/experience of the context in which the organization operates, including the target markets;

b) educational background necessary for the job (weighted according to the context);

c) training and experience about certification standards that the company possesses or that aims to achieve.

The Quality Manager within small companies deals with:

1. project reviews;
2. the identification of risks and deviations between expected and obtained results;

3. the definition of the proper testing methods on products or services;
4. the traceability of projects through the use of checklists;
5. the identification of outputs' defects and non-conformities;
6. the construction of useful archives for future use;
7. the analysis of the results to identify areas for improvement.

In larger companies, the two figures of ”Quality system manager” and ”Quality control manager” deal with different tasks:

- **Quality system manager**
  - analyzes together with the management the context of the organization, its strengths, and weaknesses, as well as its corporate assets;
  - manages the design of the quality management system in the context in which the company operates, defining the objectives to achieve and the actions to implement, including their proper indicators and the budgets;
  - maps all business processes, defining and updating critical issues, mitigation actions, and the applicable legal and clients' requirements;
  - acts as a mediator between the company figures, participating to trainings and interviews in order to develop a culture for quality;
  - defines, as manager of the human resources, the minimum requirements for each role and the competences' improvement paths;
  - draws up all the documentation necessary for the correct functioning of the management system;
  - increases the level of internal communication to promote organizational efficiency in management of ordinary and/or emergency processes;
  - analyzes the level of customer satisfaction and works to increase it;

  - carries out internal checks on all aspects and applies corrective actions in the event of inconsistency using documental, behavioral, and procedural systems;
  - monitors the objectives and applies preventive and corrective actions in the event that the organization operates off-target;
  - plans and manages the several deadlines, as well as trainings, medical examinations, laboratory analyzes, calibrations, authorizations, etc .;
  - manages emergency situations and assists the management in cases of complaints, non-compliance, accidents, etc .;
  - takes care of carrying out the management review with the other relevant company figures;
  - manages relations with the quality offices of suppliers and customers, as well as with other external organizations and authorities.

- **Quality control manager**
  - is usually appointed by the upper management or the quality manager, monitors the process phases and verifies that customer specifications and requirements are met;
  - carries out tests to verify the specifications;
  - can monitor, in the operational phases, various requirements such as cleaning, maintenance, the presence of pests, accesses, compliance with procedures for food or health safety, work safety, etc.;
  - collects the data necessary for the correct progress of the processes and delivers it to the quality system manager.

8.6.1 Quality plan

The drafting of a Quality Plan is assigned to the Quality System Manager independently or upon specific request by a customer. This document can be created applying the requirements contained in UNI EN ISO 9001 and it should systematically collect the following project documentation:

I.) functional analyzes;

II.) processes & procedures;

III.) associated resources;

IV.) assessment of the specific risk associated with external suppliers.

The Quality Plan is a key document in the management of Quality Systems. It establishes the procedures, responsibilities, and standards for projects, orders, and other.
It should always include the following:

a) the reference of the project in relation to which Quality Plan is drawn up;

b) the quality objectives;

c) the sequence of activities and their time scheduling;

d) the requirements excluded and the necessary justifications;

e) the data and basic requirements used;

f) the expectations of the customer and a plan to meet them;

g) a map of the involved processes;

h) the organizational structure and a definition of the responsibilities of the corporate functions;

i) a list of documents, rules, procedures, operating instructions, and technical documentation necessary for the implementation of the project;

j) the risk analysis;

k) references to the management of any non-conformities (with respect to predetermined expectations/standards);

l) the product specifications;

m) the references for safety controls;

n) a list of the needed technical equipment, IT tools, and software;

o) the requirements for identification and traceability;

p) the controls on the products supplied to the customer.

The structure should have the following scheme:

- purpose of the Quality Plan
- quality objectives
- reference documents
- any exclusions
- acronyms and abbreviations
- responsibilities
  - dedicated organization chart
- methods
  - processes
  - tools
- project
  - list of activities and timings definition
- controls
  - procedures
  - tests

  - audit plan
  - acceptance criteria
  - roles and responsibilities.

8.7 Head of administration

The head of administration or project administration manager is responsible for ensuring the administration and accounting management of the project. He/she works under direct control of the project manager, of which he/she is one of the main collaborators. Because the administrative and accounting operations of the project are business operations, the project manager and, where present, his/her working team, work closely with the company administration. The main activities of the Head of administration are:

- preparation and management of administrative, accounting, and tax procedures;
- collection, entry, and management of accounting data;
- preparation of the project budget based of the budget approved in the planning phase;
- preparation, organization, and presentation of reports on the progress of project spending;
- compilation of interim financial statements to verify the presence of any management anomalies and forecasts' deviations;
- coordination and management of administrative relations with customers and suppliers;
- coordination and management of relations with banks;
- preparation and drafting of the project's final financial statements.

8.8 External consultants and suppliers

The specialized consultants are experts who are external to the stable work team of the project, as well as to the company in general, who can be used in order to increase the workforce or to provide specific competences. In defining the project team, it is not necessary to differentiate the tasks of the internal workers from the ones to be assigned to external experts, as in the first stages of the project it is not necessary to foresee the full amount of external consultants' work needed in support of the project team. However, in the initial planning phase of the project it is good practice to define a budget of time and money dedicated to the possible use of external consultants.

In any project it is almost always necessary to rely on some external suppliers for the realization of any project it is almost always necessary to rely on external suppliers for the realization of all or part of the basic products. The supplier can carry out all or part of a project (a sub-project), but the project manager always remains responsible for the final delivery to his own company and must treat the supplier with the same care and attention as internal resources.

The external supplier has to appoint its own project manager, who will represent its company in full, and who will follow the instructions provided by the internal project manager. Smart goals need to be set for external suppliers, which should provide timely and accurate progress reports, as any other team members should. During the negotiation phase, suppliers are often asked to provide their own executive project plans which need to be congruent with the approved general project plan. In the event that the supplier's plan guarantees improving changes to the general plan - for example with an offer that is cheaper in price or that provides an improved outcome - such changes can be directly inserted in the general project plan. However, in case of changes that negatively impact the project constraints in terms of quality, cost, or time, the new plan will need to be approved by the program board.

8.9 The Team

The Team is a group of company figures directly involved in the project who perform the tasks necessary for the realization of the deliverables, according to the technical specifications and the timing established by the project plan. These personnel are all employees of the PM: they are obliged to inform him/her about the progress of their activities and of the project in general, highlighting any possible situations of difficulty or risk.

The task of creating a project team is not concluded by choosing the resources to include in it. According to psychologist Bruce Tuckman (1965 and in 1977 in collaboration with Mary Ann Jensen and subsequent scholars), during the life cycle of a project the team goes through 5 phases:

1. **forming**;
2. **storming**;
3. **norming**;
4. **performing**;
5. **adjourning**.

Tuckman believes that these stages are necessary to the development of any project team, to its growth and evolution, to find solutions, to plan works, to face daily problems and challenges, and to achieve the desired results. The five phases are described in more detail in the following paragraphs.

Chapter 9

The phases of the project team

Over time, Tuckman's theory, originally on groups, has been transferred to work teams, which represent one of the main forms of social grouping aimed at the realization of one or more projects needed for work reasons. He argued that the Forming, Storming, Norming, Performing, and Adjourning phases are all necessary to grow the team, to face challenges and problems, to find solutions, to plan the work, and to achieve results.

9.1 Forming or phase of formation

The Forming phase starts during the first meeting of the project team members, which at this stage are still just a group of people and not a real team. Each participant in the group begins to get to know each other and to get a first impression on the others. The exchange of information on one's own background, skills, previous experiences, and personal traits takes place. In this phase each participant also shares his/her views on the project and the significant contribution that he/she believes to be able to give to the team, providing an overview of the roles that each team member can play within the work project.

In this phase the Project Manager and Group Leader has a fundamental role, as he/she has to be able to explain the project, answer the doubts or questions that arise, and manage the communications between all the members of the group. In addition, the PM needs to explain the goals of the project in a clear and precise way, also assigning the roles and activities that each member will have to carry out.

The Project Manager's involvement, support, motivation, availability, and transparency have a key role in this first phase, as these are some fundamental elements that should never be lacking during the entire duration of the project.

9.2 Storming or phase of conflict

In the second phase, after the needed presentations, the team is confronted for the first time. There begins to be an exchange of ideas on what and how something needs to be done. Often the technique of brainstorming is used which consists in the formulation of many thoughts and opinions in a free and spontaneous way. *Brainstorming* is a meeting technique where each person thinks aloud together with the others, with the aim of reaching a common result.

In this stage some first conflicts may arise, as each of the team members will try to have their own idea prevail, to prove that one is better than the others, showing their skills, as each member is in competition with the others. This confrontation should not be eliminated but has to be managed by the PM with skills and sensitivity. It can bring out new ideas, problems, and possible solutions that should not be underestimated and that can help avoid future conflicts right from the start.

The Project Manager, who is the reference point for the team, should always try to mitigate internal conflicts, listen, and make everyone listen to all points of view in a calm and constructive way.

He/she should intervene promptly to make sure that these disagreements are not carried along during the entire duration of the project and has to calm down the most aggressive team members, while involving the most assertive ones. When team members learn to collaborate peacefully, positively, and constructively, also becoming more independent, the storming phase ends and the group leader will be able to safely delegate some decision-making processes.

It may happen that some misunderstandings are not resolved even after the main causes have been identified, it will therefore be the responsibility of the Project Manager to decide whether to demand a final ”alignment” or to replace the resources due to incompatibility.

9.3 Norming or cohesion phase

During the Norming phase a real Team is formed, that is a cohesive group which respects and values the opinions and ideas of each one, where no one prevails over the other, and where harmony and team spirit reign, which are all essential to work in the best of way. Everyone has a clear common goal to be achieved together through rules of conduct and collaboration. Members know exactly what they need to do, how they should share information and knowledge, and how to resolve any problems. The role of the Group leader is now one of support and presence, as he/she facilitates the autonomous work of the group, the creation of mutual trust, and the normal evolution of internal relationships, while keeping the Team's motivation high.

9.4 Performing or stage of performance

Team members in the Performing phase are able to work effectively in a team, without the constant supervision of the Project Manager. They possess a strong awareness of the goals to be achieved and productivity is high.

The Team is now very close-knit, able to work independently, to resolve misunderstandings in a calm and rapid way, always sharing all the decisions and moments of difficulty without needing to have the Project Manager involved. The Group Leader will, in turn, continue to monitor the progress of the project, gratifying the team with the most appropriate means for having achieved even the smallest goals. This will help make everyone feel a fundamental part of the group and will increase their motivation and enthusiasm in reaching the final goal.

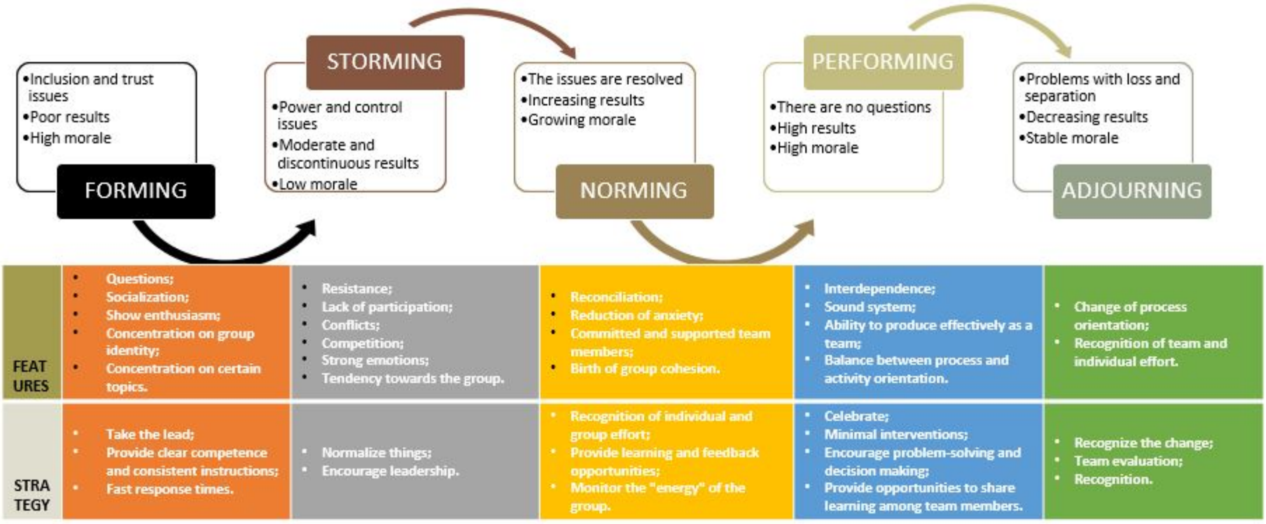
However, not all groups - and not always - manage to reach this stage (where the best performances of the team are recorded). Often some groups regress to the previous phases or stop at the Norming phase.

9.5 Adjourning or suspension phase

The project reaches its last stage, the so-called Adjourning phase, while approaching the conclusion. The project team, now consolidated, is completing the work and the last closing activities. In this final phase of teamwork, concentration is maximum and everyone is aimed at concluding their job in the best possible way, achieving the planned goal. Resources may already be assigned to new projects at the same time.

This stage is very important to get awareness of the achieved goals and of the moments of difficulty that have been overcome. Everyone should treasure the positive experience that comes to a conclusion, being able not to repeat any mistakes, and being aware of the concepts and new skills learned, thanks also to the contribution of the teammates, without whom the achievement of the result would have not been possible.

The following figure shows a summary of the five phases of a project team:



FEAT URES	• Questions; • Socialization; • Show enthusiasm; • Concentration on group identity; • Concentration on certain topics.	• Resistance; • Lack of participation; • Conflicts; • Competition; • Strong emotions; • Tendency towards the group.	• Reconciliation; • Reduction of anxiety; • Committed and supported team members; • Birth of group cohesion.	• Interdependence; • Sound system; • Ability to produce effectively as a team; • Balance between process and activity orientation.	• Change of process orientation; • Recognition of team and individual effort.
STRA TEGY	• Take the lead; • Provide clear competence and consistent instructions; • Fast response times.	• Normalize things; • Encourage leadership.	• Recognition of individual and group effort; • Provide learning and feedback opportunities; • Monitor the "energy" of the group.	• Celebrate; • Minimal interventions; • Encourage problem-solving and decision making; • Provide opportunities to share learning among team members.	• Recognize the change; • Team evaluation; • Recognition.

Figure 9.1: The phases of a project team

Chapter 10

ICT projects

The structures and categories exposed in the previous paragraphs regarding a work team and the phases that it goes through during the implementation of a project, are subject to some changes if placed within the ICT environment. In the area of Information and Communications Technology, the project team needs to be composed taking into account also the characteristics of the software to be developed. In addition, a software development project is conditioned by the life cycle of the software. The software life cycle can be represented as follows:

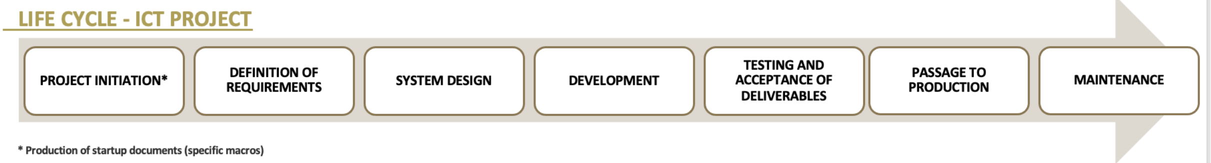

LIFE CYCLE - ICT PROJECT
PROJECT INITIATION*
DEFINITION OF REQUIREMENTS
SYSTEM DESIGN
DEVELOPMENT
TESTING AND ACCEPTANCE OF DELIVERABLES
PASSAGE TO PRODUCTION
MAINTENANCE
* Production of startup documents (specific macros)

Figure 10.1: The phases of a project ICT

The elements that mostly affect this type of project are:

1. the governance for the release of multiple versions and updates of the software;
2. the adoption of a project change management system;
3. the adoption of management systems for the problems that may possibly be encountered in a project and consequent commitment of the key stakeholders of the

project;

4. the adoption of flexible and iterative management systems, with a not excessively binding workflow.

Regarding point 1, one of the Project Management methods that have proved to be excellent for managing software-development cycles with predefined requirements is "Time-Boxing". On the other hand, for the iterative development cases (as in point 4), the Agile method is the most commonly used.

10.1 The method

The following paragraphs deal with the different ways of managing an ICT project, distinguishing between two possible strategies that can be adopted according to the different commercial scenarios.

As already mentioned in the previous paragraphs, when the Waterfall or traditional method gets chosen, the fixed parameters of the method represent the characteristics of the product as requested by the customer. In these cases, the times and costs incurred are precisely identified only after project development. Conversely, when the Agile method is preferred, costs and times get determined before anything else. In fact, these are the essential parameters on the basis of which the product starts to be implemented, ensuring the best obtainable functionalities with the given time and budget constraints. Thanks to the Agile method the end-customer is able to constantly interact with the development team (while with the traditional method this can happen only at the beginning of the order), getting to know in real time the characteristics of the product that is being produced.

With both methods the customer gets a performing product. What are the differences?

- with the Waterfall method it is not possible to predict or get to know the cost and time variables, already in the development and production phase;
- with the Agile method the desired product is created on the basis of fixed timing

and costs.

10.1.1 TimeBoxing and Agile method in ICT

TimeBoxing involves streamlining the software development process by assigning well-defined time slots to each activity that constitutes the project. Its application starts from the assumption that the project is bound to the following elements: Time, Cost, Scope, Quality, and Resources.

This method is also called "iterative" because, due to its path with well-defined time stages (TimeBoxing), it promotes a high commitment of the resources, ultimately resulting in an improved progress of the project.

The iterative approaches focus on rapid developments, trying to maximize the value of the deliverables at the contractualized timings and costs. The end-customer constantly interacts with the development team, getting to know perfectly the product that is going to be produced.

This technique involves the splitting of complex activities into tasks, that is into simpler, non-autonomous, units. A homogeneous set of tasks (or a project phase) in a certain time slot with fixed duration is called a Timebox. Each Timebox has specific goals, deadlines, and fixed budgets. Once the available time and resources have been fixed (and consequently the desired cost has been defined), products and their quality are developed as a result.

The life cycle of an ICT project from an Agile perspective is intended as a sequence of iterations. Each iteration includes analysis, design, implementation, and testing activities; during each phase it is possible to add parts to the product (Iterative Development) or refine it (Incremental Development). This approach is based on the feedback obtained on previous iterations to improve the subsequent ones.

The logic underneath it is to produce the deliverables as quickly as possible and then refine them through successive improvement cycles:

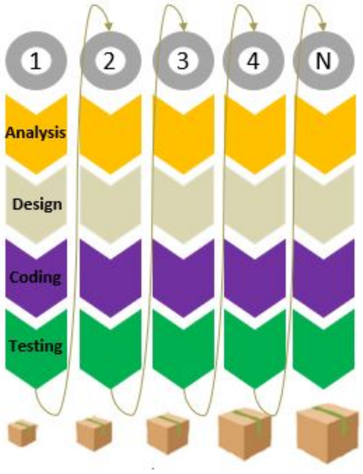


Figure 10.2: Phases of Agile

In order to be correctly finalized, this type of method requires cross-functional and self-organized teams, who know how to correctly use the necessary tools while maintaining appropriate management processes.
The high level of specialization that is usually requested needs to be contrasted with a widespread simplification of all operations and with an open approach to change. An important role is played by the testing activity, checking the correct functioning of the whole system.
One of the complexities that needs to be properly managed in this case, is the need for constant involvement of the customer in the design phases.

10.1.2 Waterfall or traditional method in ICT

The "Waterfall" approach is not suitable with changing needs and specifications (for these cases it is preferable to adopt the Agile approach).
In the Waterfall or traditional method, the fixed parameters of the method represent the characteristics of the product as requested by the customer. Precise timings and costs are identified only at the end of the project development.
The life cycle of an ICT project from a Waterfall perspective is intended as a sequence of phases:

- requirements analysis;
- drawing;
- implementation;
- test;
- installation;
- maintenance.

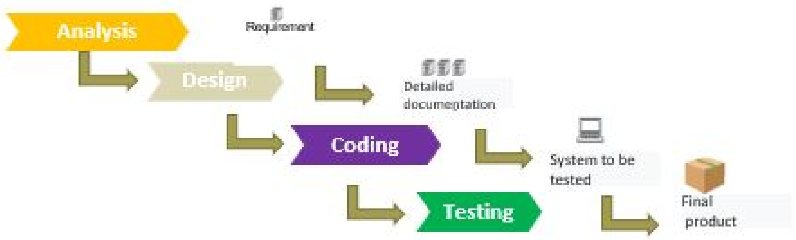


Figure 10.3: Life cycle of a project

From the presence of well-defined, agreed and formalized requirements, it follows that any changes can only be granted through escalation actions (formal process implemented to highlight any problem that has occurred to a higher authority). Therefore, the preliminary stages of the project play a fundamental role in intercepting any potential obstacles. The initial project documentation needs to be very detailed in order to document all the particular requirements that emerged during the work progress. The time constraints initially set for each phase and the release plan allow for easy monitoring and control. The customer views the deliverables only when the project is completed.

Chapter 11

The team of an ICT project

In an ICT project, the team members have specific skills and professionalism. The team is composed of people who can perform all the tasks necessary for the realization of the deliverables, according to the technical specifications and timings established by the project plan. The macro-activities carried out within an ICT project are:

a project management;

b preparation of the development and production environments;

c functional analysis of the systems that have to be implemented;

d prototypes;

e technical design (architecture, data, processes, and technological infrastructure);

f coding and testing;

g management of versions and releases;

h documentation management (product documentation and user documentation);

i change and configuration management;

j quality assurance and control.

This implies the presence of a support structure regarding:

1. governance and management policies for ICT projects;
2. metrics for projects administration and control;
3. trainings of managers and project teams;
4. procedures and standards to be applied in software development projects;
5. development environments, techniques, and tools;
6. documents and workflow management system.

An ICT project consists of various tasks for each of which a specific level of experience and qualification is required. For this reason, all participants in a project can be considered part of the project team. Within every project Team there are two types of components:

1. internal components (human resources allocated full time or at least at 60 % of their time);
2. added components (human resources who work in the project temporarily and often on specific issues, e.g. a project phase, creation of specific documents, etc.).

In the following paragraphs the role and tasks of the main company figures of an ICT project are analyzed, including:

i) team leader (or team manager);

ii) designer;

iii) analyst;

iv) specialist technician;

v) professional sub-levels.

11.1 The team leader

In addition to the project manager, there may also be other company figures who are responsible for certain activities or project lines. The team manager or team leader should have the ability to manage and develop an activity, promoting the correct adoption and application of the project management methods.

The team leader is a company figure who is not directly responsible for the budget but who has the authority to make decisions that affect the organization and development of the project activities. Therefore, knowledge and ability to use the main business systems' modeling methods are among the hard skills that a manager should always possess. Furthermore, a manager should have soft skills like strong communication and negotiation capabilities, which are vital to adequately involve the work group and relate to external interlocutors. He/She should develop leadership skills to effectively lead the team.

Team leaders are responsible of:

- defining the work plans of the group, together with the PM;
- allocating resources on activities;
- assigning tasks to team members;
- checking the progress of technical and non-technical activities, as well as of incurred costs, in constant contact with the PM;
- communicating any critical issues or slippings from the plan;
- interacting with each human resource under his/her control;
- attending alignment meetings.

11.2 The designer

The designer is responsible for the technical design and development of the applications, planning their overall architecture, coordinating the construction of their components, and controlling their quality. Unlike the manager, the designer needs to have a good technical knowledge of processes and products. Like a manager, on the other hand, he has a role of coordination and management and thus he/she should also have good communication skills needed to interact with users and upper management, to collect the project requirements, and to inform the working group on the detailed characteristics of the components being created.

11.3 The analyst

The analyst is a professional figure with in-depth knowledge of the technology used and the products defined in the project. He/She supports the designer by taking care of detailing and realizing all the methodological aspects necessary for the design, the implementation, and the documentation of the system. He/She also takes care of the analytical description of processes in a language understandable to specialist technicians, sharing its representation with the user. The analyst in the ICT field (Business Analyst-BA), or functional analyst, is one of the main players in the process and works as a liaison figure between two main areas of the company organization, namely:

- **business manager**, or more simply the end user, with whom the BA interacts to be informed of the macro-objectives. Through a series of targeted questions during ad-hoc meetings where all Stakeholders are present, the analyst is able to identify goals, requirements, taste of the user, and so on.

  Once the macro-objectives (requirements) have been defined, it is the BA's task to study the competition, evaluate the actual market demands, and analyze the considered target, trying to better identify the market evolutionary trend. The accuracy and scrupulousness with which all these elements are collected is crucial for the development of the project;

- **the developers, or IT area**, which is mainly composed of a team of programmers and experts in the field of UXD (User Experience Design), that is, those who define the usability of the future product.

The center of the work of a functional analyst in the IT field consists of:

1. specific expertise on the type of processes being handled →"speak the same language as the customer".
2. analysis drafting →identify user requirements (or business requirements) through special meetings and define how to synthesize and rationalize the processes that need to be implemented with the requested IT solution.
3. collaboration with IT architects →the analysis is transmitted to the development area where a new detailed technical analysis is conducted and a technical specification document is drawn up that highlights whatever is needed to start the developments.
4. dialogue with development →the functional analyst needs to be able to talk to the IT people and to give the right support both in the joint analysis phases and in the test phases.
5. testing →in the IT field, tests have extraordinary importance and are a task of the functional analyst. Since developers have necessarily a partial vision of the topic, it is the responsibility of the analyst to design and conduct the tests in the most possible articulated way. The functional analyst takes care of creating the various scenarios to test the solution, first in a purely design sphere and then, when the software is ready, together with the customer.

11.4 Technical specialist or technical analyst

The Technical specialist is the team member that deals with the development of the components necessary for the realization of the project outputs. He/she has skills specifically tailored for the product being created (eg: web developer, systems engineer, database ...). He/She is responsible for carrying out unit tests of the released modules and for producing technical documentation. Some of the required skills of this company figure depend on the needs of the project (e.g. consolidated programming experience, knowledge of the development environment and database model, knowledge of protocols, communication devices, emerging technologies, etc.), while others are transversal to all projects (such as the capability to adopt the analysis documentation and to independently manage the creation and production of supporting documentation, etc.). In short, the Technical specialist or technical analyst deals with:

- the development of systems, that have been studied and written by the functional analyst, capable of performing the functions requested by the customer for the realization of tailor-made products;
- the selection of company softwares and optimal-customized technical solutions;
- carrying out unit tests of the issued modules;
- producing technical documentation.

11.5 The Tester

A Tester is the professional figure who works in the field of Quality Assurance (a process used to verify the correctness, completeness, and quality of the developed software. It includes a series of activities conducted with the intent of finding errors in the software so that they can be corrected before the product is released to end-users). He/She independently executes test cases, also creating the test designs. Below are listed the figures related to the scope of QA:

- **QA manager**: the QA manager has the same role as a Governance Test manager but he/she also manages the activities of a business stream.

- **Test manager (role of governance)**: a Test manager deals with the same activities as a Test leader but on projects that handle bigger quantities of resources dedicated to the tests.

- **Test leader (role operation)**: he/she is responsible for monitoring and coordinating the testing activities on individual project streams.

- **Tester specialist**: his/her role is to automate the test activities and manage the supporting tools.

- **Tester Analyst**: it's a company figure who reads the project documentation and who deals with the design and drafting of test cases, executing them in direct contact with the stakeholders.

- **Tester (simple tester)**: it's a company figure who simply runs test cases of the test plans.

11.6 Professional sub-levels

Each of the previous company profiles can be further divided into sublevels based on the level of experience already acquired. Usually the sublevels are:
Junior →an individual with basic skills or limited previous experience;
Senior →with a minimum of 2 years of experience and with proven autonomy in carrying out the assigned tasks. Senior resources need skills to manage the junior resources assigned to them.

Chapter 12

Functional analysis

12.1 Objective of the functional analysis document

The objective of functional analysis and the resulting documents is to clarify, detail, and document the functions, services, and performance that have to be offered by a software system. All the information collected in the analysis phase represents the starting point for designing, realizing, validating, and maintaining a software product.

12.2 Analysis activities

The steps/elements that compose a functional analysis are:

I the definition of the problem. This activity consists in fully understanding the problem that the system being developed is called to solve;

II the drafting of the feasibility analysis. It aims to establishing whether the objectives that have been set for the development of the system are reasonable and achievable;

III the preparation of a cost and benefit analysis. It first assesses the economic convenience of developing the software system, taking into account the expected costs and benefits that it will be able to provide;

IV the performance of a domain analysis. It is capable of thoroughly examining the context or application domain in which the system will have to act (for example, in a bank: Customer, Account, Operations, Mortgage, etc.);

V the drafting of the requirements analysis. It has to specify in detail the services, functions, and possibly the performances required by the system.

12.3 The minutes

During the very first phases of the project, at the end of any meeting in which one has taken part, whether it is a formal meeting or simply an internal alignment, it is necessary that the BA be able to draw up a correct and complete report of what has been discussed. This is essential to avoid the dispersion of important information that may be useful for the proper continuation of the project activities. In large projects, the minutes' drafting is delegated to the PMOs or to specific company figures who are responsible for dictating the timings of the project. On the analysis side, the preparation of the minutes serves the BA to better manage the requirements in order to create a **Feasibility Document**(feasibility study) that will include all the requests of the stakeholders.

A detailed minutes should specify the following points:

- title of the received Invitation (usually convened by email);
- summary table of the main information such as version, date, author, status, subject who verifies and approves, brief recap of what happened, etc.;
- detailed table about the state of approval;
- compilation of the paragraphs with the details of the meeting, the data of the participants, the points of the agenda discussed, and the description of the activities.

12.4 Recap

In summary, writing the minutes of a company meeting means to write in a simple and targeted way. In order to attend the meeting with an appropriate standing it is sufficient to follow three basic steps:

1. prepare before the meeting takes place. Try to collect all the necessary information relating to the topics that will be discussed, writing them down in a ”pre-meeting” reminder table that will help to better focus the attention during the meeting only on the key topics:

    a) reason for the meeting;

    b) the topics that will be discussed;

    c) first name, last name, and role of the participants [1];

    d) where and when;

    N.B. The same standard model should be used to draft the minutes of any meeting.

2. have an active role in the meeting by communicating to the participantd that you are taking notes to draft the minutes and that you will be responsible for reporting the decisions that will be taken during the meeting;

3. report any feedback received after the meeting and (maximum a couple of days later) send the minutes by e-mail to all the participants.

[1] It is also a good idea to know in advance the possibility that there will be absent people because they can- it would be necessary to subsequently communicate the results of the meeting that was held. This is depends on the role covered by the absent profiles and the organizational uses of the company.

12.5 Requirements engineering

12.5.1 Requirements engineering objective

In this phase, the analyst has the goal of defining the features that the new product (or the new modified product) has to offer, or the requirements that the developed software has to meet. This phase ends with the drafting of a detailed requirements specification document that accurately describes the functionalities of the new software. Through a series of questions, the analyst stimulates the client to reasoning by trying to deduce hidden (or unexplained) information and generate new ideas.
The analysis of the requirements normally takes place as a negotiation between different subjects:

- business contacts who describe what they expect from the functionalities to be developed;
- functional analysts who interpret the "desired" outcome and describe the possible solution;
- technical analysts who develop/implement what has been described.

The dialogue is not always simple and for this reason, in a subsequent intermediate phase between the collection of requirements and the development of the software, prototypes or previews (partial releases) are often developed.
The main document produced by the requirements analysis is the requirements specification document.

12.5.2 Definition of requirement

The term requirement derives from the Latin “requisitus” (participle of “requiro”) and has the meaning of:

I) search, research, go in search of ...

II) ask to obtain or to know, inquire about ...

III) needing, demanding ...

The requirements represent the characteristics of a product/service. They are usually collected, using various methods, within a structured document containing the information collection to establish the architectural, content, graphic, and functional needs of the entire project. This information represents the needs of the stakeholders and will be the subject of analysis for the feasibility of the project. This phase requires collaboration between several groups of different actors and resources with diversified knowledge: users and developers have to collaborate to write the **requirements specification document**, using a simple language that will be subsequently formalized and structured (in Unified Modeling Language - UML[2] or other) during the analysis phase. Mistakes made during this phase are difficult to correct and are often detected in the delivery phase. The collection of requirements is divided into three main phases:

1. exploration: (interview with stakeholders, cataloging of customer/business requests, competition analysis);
2. organization: (drafting of the requirements). In this phase, the focus is on the user's point of view and attempts are made to trace the boundaries of the system to be developed. In particular, the following get specified:

   a system functionalities;

   b user-system interaction;

   c errors that the system has to manage;

   d constraints and conditions of use.
3. review and approval: (preliminary to the technical design and development phase).

[2] Refer to the definition in the Glossary

12.5.3 Classification, characteristics, and collection of requirements

The requirements differ mainly in:

- **functional requirements** that describe the interactions between the system and the external environment (users and external systems) regardless of its implementation.
- **non-functional requirements** that describe aspects of the system that are not directly related to the functionalities.

The characteristics attributable to the requirements are:

(a) **completeness** (all possible scenarios should be considered, including exceptional behavior);

(b) **consistency** (they should not be in contradiction);

(c) **un-ambiguity** (a single system needs be defined, while interpreting the specifications in different ways should not be possible);

(d) **correctness** (the requirements should show with accuracy the system that the customer needs and should be as closely as possible to reality);

(e) **concreteness** (the system can be implemented in a reasonable time);

(f) **verifiability** (the system needs to be verifiable at the start of the project and every time that it is implemented with tests);

(g) **traceability** (each requirement can be mapped with a corresponding application functionality).

The process of gathering the requirements can be described with the following steps:

1. **identify the actors** - an actor is an external entity that communicates with the system and can be an user, an external system, or a physical environment.

2. **identify scenarios** -a scenario is an informal, concrete, and focused description of a single characteristic of a system and describes what people do and experience as they try to use computing systems and applications.

3. **identify use cases** - a use case describes a series of interactions that occur after an initialization made by an actor and specifies all possible scenarios for a given functionality (in other words, a scenario is an instance of a use case).

4. **refine use cases** - the elements that are manipulated by the system get detailed.

5. **identify the relationships between actors and use cases** - There are various types of relationships between actors and use cases: Communication; Extend; Includes (the definition of which is explained in the following paragraphs).

6. **identify the participants** - users and developers should create a glossary of terms for use cases. Starting from the terminology that users have, they then negotiate changes. The glossary that gets created is then included in the end-user manual.

7. **identify non-functional requests** - see definition in the Glossary of non-functional requirements.

12.6 Requirements analysis

A Requirements analysis is necessary to produce a correct, complete, consistent, and unambiguous model for the system. The difference between the collection and the analysis of the requirements is that, with the latter, developers deal with structuring and formalizing the requirements received from the user, looking for any errors made in the previous phase (the former, the requirements collection phase). By formalizing the requirements, the analysis forces developers to identify and implement difficult system functionalities already at this early stage. The overall analysis model consists of three individual models:

- functional (represented by use cases and scenarios);
- object-oriented (represented by class diagrams and object diagrams [3]);
- dynamic (represented by state diagrams and sequence diagrams[4]).

The activities to be planned based on the requirements are:

i drafting of the specifications document;

ii estimation of implementations;

iii evaluation of implementations;

iv confirmation or replacement with alternative implementations;

v validation of the analysis document.

12.7 The tools of the functional analyst

There is no precise method on how to carry out a functional analysis that can scientifically answer questions like: what contents are essential? What documents are needed? The perimeter of an analyst's activities and skills can vary (sometimes he/she also deals with organizational analysis, information design, and so on). The main fundamental components of the analyst's office are:

- the functional analysis document
- the minutes
- requirements engineering
- flow charts and UML diagrams
- Use Case Specification (UCS)

[3] Description of the tools in the following paragraph

[4] Description of the tools in the following paragraph

- CR
- test phase
- User Manual

12.8 Feasibility Analysis and the Feasibility Specification Document

A ”feasibility study” includes all the activities aimed at determining the convenience of carrying out an intervention and at providing the project team with information related to the investment necessary for the achievement of its purpose. A feasibility study has to:

- highlight the context of interest of the project;
- reach a level of detail that allows to verify the feasibility and evaluate costs, risks, and benefits.

In order to achieve these objectives, the feasibility study needs to:

1. identify architectural solutions;
2. identify organizational solutions (for the system and for its implementation);
3. define the timelines/deadlines needed to draft a ”preliminary project”.

The feasibility study has to provide all the information necessary to draw up the related documents:

1. **current situation →** context, problem, analysis and diagnosis, constraints, and definition of the goals.
2. **“Preliminary design” of the solution →** requirements (of the process and of the system), specifications, and methods of implementation.

3. **risk analysis →** risk factors, analysis, and management methods.

4. **methods of implementation of the project →** segmentation, global specifications, planned acquisitions and accomplishments, a draft plan.

5. **timing analysis →** evaluation of the timings of the project.

6. **cost-benefit impact analysis →** evaluation of benefits, cost estimation, and investment analysis.

7. **change management →** strategy, tools, and actions.

8. **recommendations for the implementation phases →** for procurement (form of acquisition), for project management, and for the drafting of the contract.

In principle, the above information should be included in the sections of the feasibility study also called "Feasibility Specification Document", which is divided into the following paragraphs:

- **introduction** it contains the references used for the drafting of the document and the main definitions, acronyms, and abbreviations;

- **project objectives** a paragraph dedicated to the needs and, in general, to what it is hoped to obtain by achieving the objectives;

- **needs** within which the wishes of the project stakeholders are listed;

- **TO BE process** paragraph that summarizes the planned changes with respect to the current characteristics possessed by the functionality (normally indicated: 'with respect to the AS IS');

- **functionalities** it includes the specific characteristics of the project. This paragraph needs a division into sub-paragraphs containing the work breakdown structure (WBS), the project assumptions, the functionalities foreseen in the project budget (in-scope), the functionalities outside of the project (out-of-scope), as well as a description and analysis of each single functionality.

12.8.1 Business processes

In order to understand the company and the infrastructure to which the functional analyst has to provide advice, it is important to be able to read the graphs that depict the business processes. A business process is a set of activities performed to produce a product or service; it describes:

a) the activities performed during the process;

b) the organizational units that participate in the execution of the process;

c) the input and output data being used;

d) the IT systems involved;

e) the events and risks that can impact the execution of the process.

The process organization can be represented with its activity elements, data elements, and organizational elements. The sequence of activities in terms of a business process is represented using process chains. The start and end events for each activity in the process should be modeled. Events can both trigger activities and be results of the activities. The start and end elements of a business process are always events. In fact events define the state or condition that causes a task to start, as well as the state or condition that causes a task to end. An event can be the source of several simultaneous activities; on the other hand, an activity can cause various events.

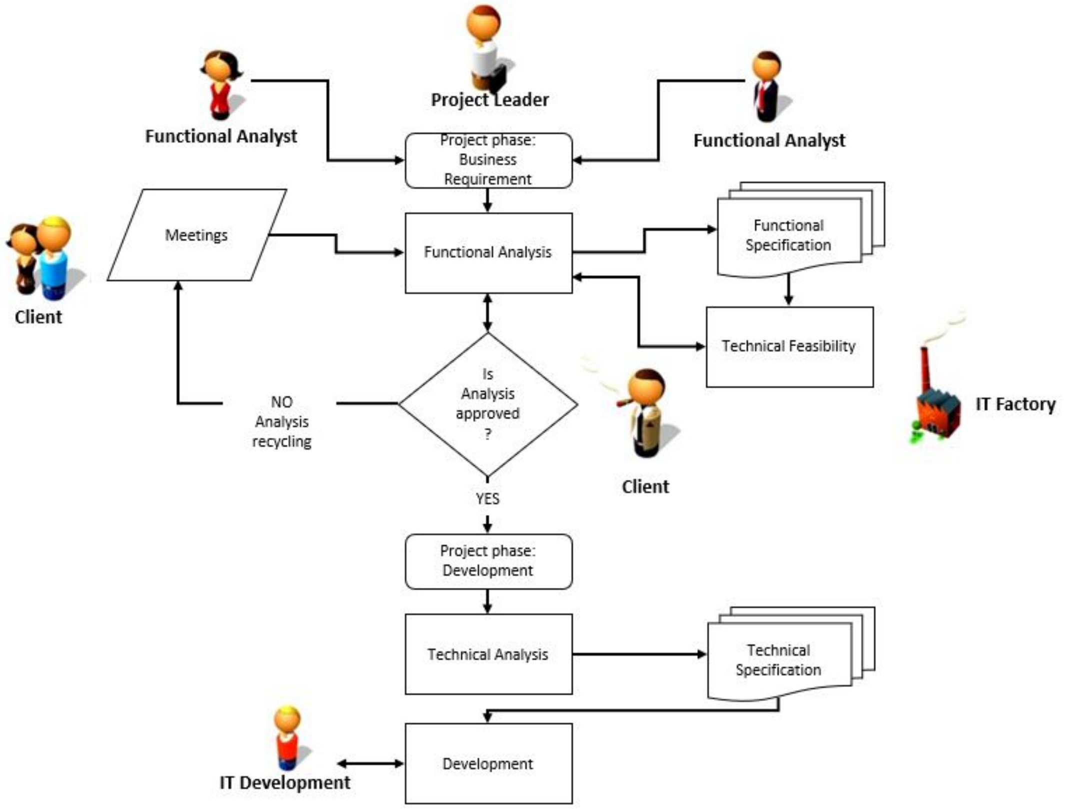


Figure 12.1: Process of carrying out a company project

12.8.2 BPMN- Business Process Modeling Notation

The Business Process Modeling Notation (BPMN) is an effective representation standard easy to use and understand by business user interested in modeling, designing, and computerizing the business processes.

Many business figures such as process analysts (who build the initial drafts of organizational processes), programmers, and developers of IT applications, as well as managers and executives responsible for managing and monitoring the processes, need a graphic representation of the processes in order to be able to properly work on them.

The BPMN is an evolution of the flow charts with some additions and modifications that allow it to overcome some limits in the modeling of business processes.

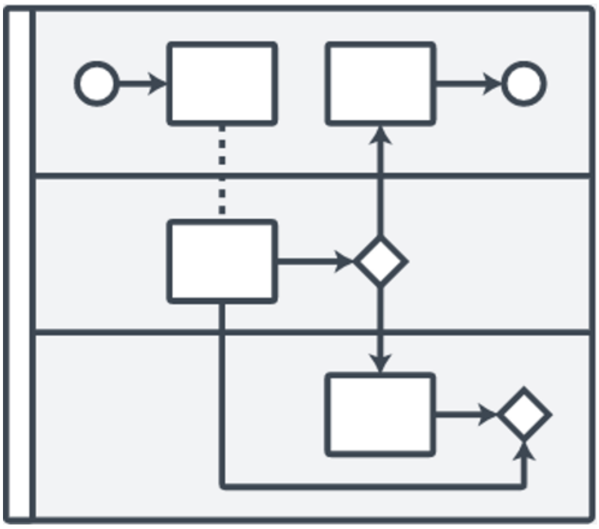


Figure 12.2: BPMN Graphic Representation

Through the BPMN, process diagrams (BPD - Business Process Diagram) are built with graphs or networks made up of ”objects” that graphically represent process activities, connected by control flows that define the logical relationships, dependencies, and order executions of the activities.
There are two different situations in which BPMN is used:

- in the case of processes that take place entirely within the company. The flows of activities essentially remain within the process lane, never crossing the boundaries of the organizational unit involved (”internal” processes);
- in the case of “collaborative” processes between two or more business entities (companies, organizations, units, etc.). The diagrams take the general view of a third party as an ”outside observer”. The moments of interaction are those that are ”visible from the outside” or even without the internal details of the activities of each process being known.

12.8.3 BPD - Business Process Diagram

A BPD is built with graphic elements whose main characteristic is their simplicity of interpretation for business and organizational process analysts. Its graphic representation recalls the classic symbols of a flowchart, with activities typically represented by rectangles and decisions illustrated by rhombuses. The BPMN standard defines some basic graphic elements that are generally sufficient to model a wide range of processes, and to which integrations and more elements are added on a BPD to give greater representative effectiveness for complex processes, without changing the basic setting of the notation. The four basic categories of graphics of BPD are listed as follows:

- flow object;
- connecting object;
- swimlane;
- artifact.

12.8.4 The standard graphic elements of a BPD

Event: it is indicated by a circle and represents something that ”happens” during a process. Events can have a ”cause” that triggers them and a possible ”outcome” (result). There are three types of events depending on their location within the flow of a process: the start, the intermediate event, and the end.

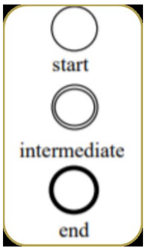


Figure 12.3: Graphic representation of events

Activity: it is identified by a rectangle with rounded corners and generically indicates a task or operation carried out within the process considered. An activity can represent an elementary and "atomic" task that is not further decomposable, or a sub-process that can be further broken down into its elementary tasks. In this second case, the "+" symbol is used in the lower central part of the rectangle.

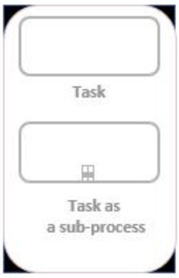


Figure 12.4: Graphic representation of activities

Gateway: symbolized with a rhombus, it defines the point in the process where the flows of activities diverge or converge. It is used to represent traditional decision points as in classic flowcharts, but also simple bifurcations of the flow of activities into parallel activities, or vice versa the reunification of parallel activities in a single flow.

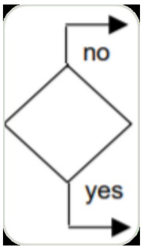


Figure 12.5: Graphic representation of a Gateway

Connectors: logical links between flow elements. There are 3 types of connectors:

- **Sequential**: it is used to indicate the logical-sequential order between activities or events of a process;
- **Message flow**: it symbolizes the fact that a message is exchanged between two

different activities or entities participating to the process, one that transmits and the other that receives the message;

- **Association**: it is used to indicate a simple link between data, text, and other objects. It is also used to indicate the inputs and outputs of activities.

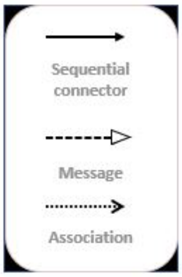


Figure 12.6: Graphic representation of Connectors

Lanes: they are used to illustrate the organization of activities and flows in homogeneous groups (grouped by organizational functions involved, operational and management responsibilities, etc). The BPMN has two types of lane: **Pool (lane of an organizational unit)**, it is a well-defined organizational entity that carries out its own process, possibly interacting with other organizational units; **Lane (or sub-lane)**, it is a subdivision of the lane of an organizational unit into partitions that cover the entire length of the lane, vertically or horizontally.

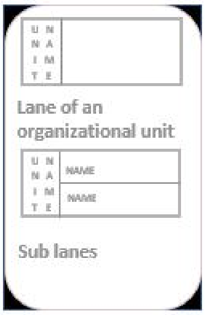


Figure 12.7: Graphic representation of lanes

Artifacts: they constitute additional symbols that can be added to specify additional objects or entities.

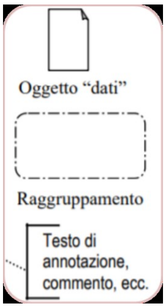


Figure 12.8: Graphic representation of artifacts

Data objects: they are used to describe the types of data necessary to or produced by an activity. They are connected to the activities through the "association" connectors.
Groups: they serve for purposes of specification, analysis, and documentation.
Annotation: they allow the modeller to add a clarification text in order to provide further information to those who have to interpret the BPD.

12.9 Graphs and diagrams

12.9.1 Flow Chart

The flowcharts can represent the functioning of a service, an interaction, or a process. The flowchart is an analytical system that follows, step by step, all the tasks necessary to perform any activities in the operational phase, that is, a tool that examines the set of tasks related to a process in any function or area of responsibility.
The flowcharts show the sequence of events that appear in a process: they start with the inputs, show the transformations that take place, and end with the outputs. They are carried out through interviews and facilitate the simplification and optimization of the process.
Illustrating the steps for completing a task, the flow charts have a well-defined structure:

- rhombuses represent decisions;

- rectangles represent the states of the system;
- the connectors give orders of execution of decisions and states.

In addition, a flowchart can contain notes that provide additional details.

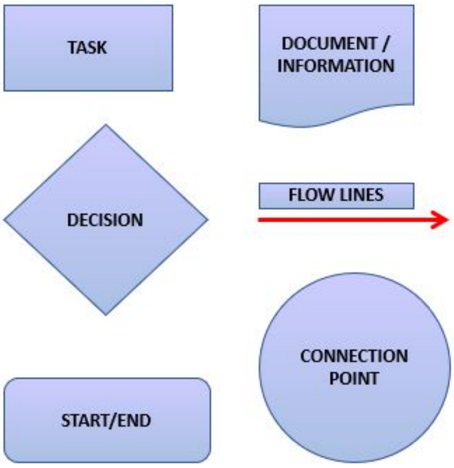


Figure 12.9: Graphic representation of the characteristic elements of the flow chart

12.9.2 UML Diagram

UML (Unified Model Language) is a set of languages that, used together, enables to describe/model all (or almost all) the relevant aspects of a system, according to an Object-Oriented approach. The use of this standardized method of representation entails significant benefits:

- it allows to have software systems professionally designed and documented even before the related code is written by the developers;
- it allows coding to be easier and more efficient;
- it has remarkably low development costs;
- the representation of the entire flow helps predict and anticipate any ”holes” in the system, providing a clear idea of the entire system to anyone involved in the development;

- it allows for easier future changes to the code which, on their side, reduce the costs for maintaining the system;
- it facilitates a more efficient and direct communication and interaction between the human resources that take part in the development of the system.

UML diagrams can have two generic functions:

a. they can be used to visualize, specify, construct, and document the static aspects of a system. Among these we find:

   i. class diagrams;
   ii. object diagrams;
   iii. component diagrams;
   iv. composite structure diagrams;
   v. deployment diagrams;
   vi. package diagrams;

b. they can be used to visualize, specify, build, and document the dynamic aspects of a system, highlighting the flow of messages over time. In this case, they are:

   i. use-case diagrams;
   i. state charts;
   iii. activity diagrams;
   iv. sequence diagrams;
   v. communication diagrams;
   vi. timing diagrams;
   vii. interaction summary diagram;
   viii. interaction overview.

Depending on the functions described above, they can identify nine basic diagrams. However, it should be borne in mind that it is possible to build and add diagrams different from the standard (hybrid) ones compared to those defined by the language. The nine basic UML diagrams are:

- class diagram:
  - the following is an example to get an immediate idea of what a class is. All objects or living beings can be traced back to certain categories (e.g. computers, cars, plants, animals, etc.). These categories constitute the classes. A class is a category or group of objects (for convenience, with this term we also include living beings) that have similar attributes and similar behaviors. Class diagrams provide the representations that are mainly used by developers.
- object diagram:
  - an object is an instance of a class[5] - that is, something specific, with certain values, attributes, and behaviors.
- use Case diagram:
  - a Use Case is a description of a particular behavior of a system from the user's point of view. For developers, use-case diagrams represent a remarkable tool: in fact, through such diagrams, they can easily get a clear idea of the system requirements from the user point of view and thus write the code without fear of misunderstanding the final purpose.

    The Actor is an entity that interacts with a use case by starting the sequence of actions described by it and, possibly, receiving precise answers from the system.

    It can be one person or even another system.

[5] An instance is a particular object of a given class. Each instance is separate from the others,but it shares its general characteristics with the other objects of the same class, through the commune description of fields and methods.

- activity diagram:
  - the activities that occur within use cases or within the behavior of an object typically occur in a well-defined sequence. These sequences are represented with the activity diagrams.
- state diagram:
  - at a given instant, during the system's operation, an object is in a particular state. State diagrams represent these states and their changes over time. Each State Diagram begins with a symbol that identifies the initial state (Start State) and ends with another symbol that represents the final state (End State). For example, each person can be identified by the following states: newborn, infant, child, adolescent, adult, elderly.
- sequence diagram:
  - class diagrams and object diagrams represent static information. In a functioning system, however, objects interact one another and these interactions occur in relation to the flow of time. The sequence diagram shows time-based dynamics of the various interactions between objects.
- collaboration diagram:
  - the elements of a system work together to realize and satisfy the needs of the system. A modeling language needs to have a way to represent such cooperation. The Collaboration Diagram is used for this very reason.
- component diagram:
  - nowadays, in software engineering, an organization model is increasingly used, according to which everyone in the work team is involved in working on a different component. The component diagram describes this important feature.
- deployment diagram:

- the deployment Diagram shows the architecture of a system from the physical and logistical point of view. This diagram can describe and show the computers and the various devices, together with their connections and the softwares installed on each machine[6].

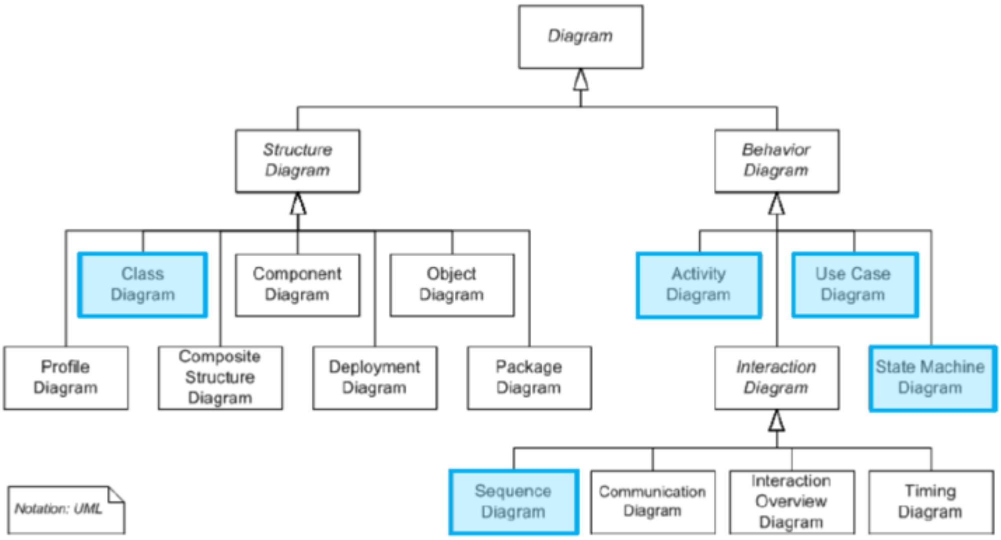


Figure 12.10: UML Diagram Hierarchy

12.9.3 Class Diagram

The Class Diagrams provide a static view of the classes involved in the creation of a System. This type of diagram is mostly used during the requirements analysis and system design phases to represent the structure of a system. In a class diagram:

- each class is graphically represented by a rectangle;
- the name of the class, by convention, is a word with a capital letter and appears near the top of the rectangle (e.g. Home). If the name of the defined class consists of a compound word, then a notation is used according to which the initials of each word are written in uppercase (e.g. HomePage);

[6] A description of the collaboration is not made within the following paragraphs of the object, components and deployment diagrams, as these are useful representations for the 'technicians' of the project

- each class can have zero or more properties called Attributes. An attribute is a set of values associated with the objects of a particular class. An attribute, whose name consists of only one word, is always written in lowercase characters.
  If, on the other hand, it consists of several words, the name of the attribute is written as a combination of all the words that compose the name with the first word written in lowercase and the following ones with their first letter in uppercase (e.g. Attribute1: type1);

- the list of attributes of a class is graphically separated by a horizontal line from the name of the class to which it belongs;

- to each attribute it is possible to associate a type of relation, a data format (string, float, int, bool, etc.), as well as a specific default value;

- Multiple attributes can be "active", that is, can be put in relation to one another through operations (or methods). The operation or method is an action that the objects of a class can perform within the system. The name of an operation follows the same rules used for attributes. The operation list is graphically represented below the attribute list, separated by a horizontal line. In the brackets after the name of an operation, it is possible to show the parameters and types necessary for the method. Finally, if the method represents a function, the returning type of paramenter needs to be specified too.

- Multiplicity is a type of relationship/association in which the number of objects belonging to a class that interacts with the number of objects of the associated class is shown:

  – one to one (1-1);

  – one to many (1-n);

  – one to one or more (1-1,2,3 ...);

  – one to zero or one (1-1,0);

- – one at a limited interval (e.g. 1 to 2-20)
- – one to an exact number ”n”;
- – one to a set of choices (eg: 1 to 5 or 8);
- – use (*) to represent ”many” and ”more” options.

The following figure shows an example of a Class Diagram depicting the operation of a hospital.

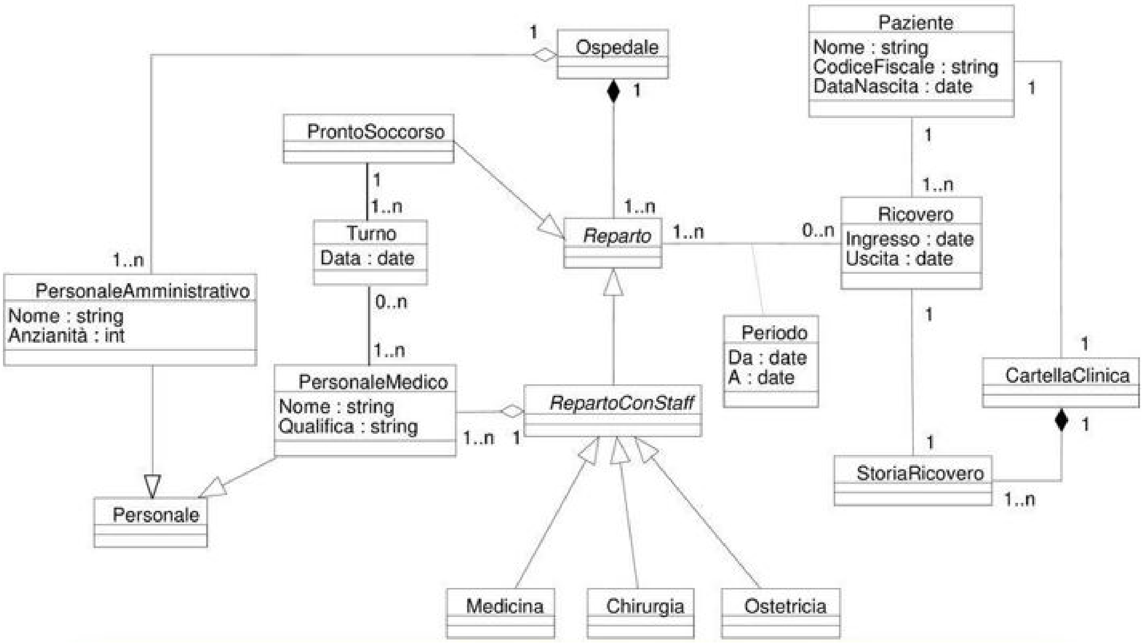


Figure 12.11: Class Diagram Example - (Real italian diagram case)

12.9.4 Use case diagram

The Use Case Diagram (UCD) offers a dynamic view capable of showing how the system and its classes change over time.

The static view of the system helps analysts communicate with the clients. The dynamic view, on the other hand, provides analysts with a valid means to establish a clear dialogue with the team of developers, helping them create the programs.

The Use Case is an excellent tool to stimulate potential users to express their criticisms and opinions about the functionalities of the system. Through interviews with end-users, the definition of high-level Actors and Use Cases can be obtained which describe the functional requirements in general terms. This information also helps trace the limits and purposes of the system under development.

One **use case** includes:

1. name and a brief description
    - the objective/purpose of the use case in a few lines (one or two periods, possibly mentioning the actors);
2. detailed flow of events:
    - the description of the primary and alternative flows of events that follow the start up of the use case;
    - initial focus on a ”happy path”;
    - use verbs in the present tense in active form (Qualify: AAA actor, SSS system);
    - indicate starting and termination conditions;
3. special requirements: all non-functional requirements related to the specific UC;
4. preconditions: conditions that bind the start of the use case;
5. post conditions: they define the state of the system when the use case ends, especially if it ends with anomalies. They may be different for each alternate flow, but few are recommended.

An **actor** is defined with:

- a brief description (a few lines only);
- what or who it represents;
- why there is a need for the actor;

- what interests the actor has in the system.

The Use Cases are collections of scenarios describing a sequence of events concerning the use of the system. The sequence of events described by a Use Case can be initiated by a person, by another system, by a piece of hardware, or by the flow of time.

The relationship between the use case and the actor can be of three types:

- **inclusion**: allows the reuse of a use case within another use case. To represent it graphically, a dashed line is used that connects the CU with an arrow pointing to the CU on which the other CU depends. Just under this line, the word << includes >> is added enclosed, as you can see, between double brackets formed by the symbols ”<<” and ”>>”

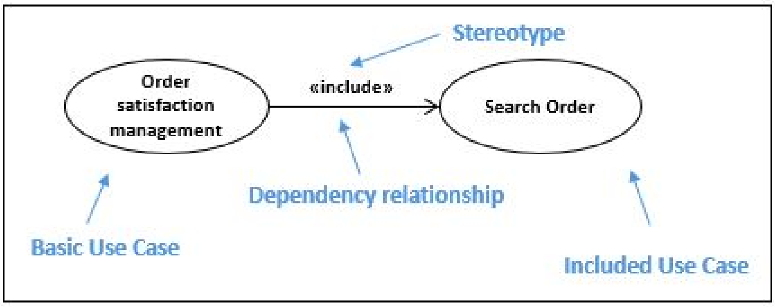


Figure 12.12: Example inclusion

- **extension**: allows you to create a new use case by adding more steps to an existing use case.
  - Here too a dashed line with a trailing arrow is used for represent the extension, together with a stereotype that shows the word <<extend>> in parentheses. Within the use case base, the point of extension appears under the name of the use case itself.

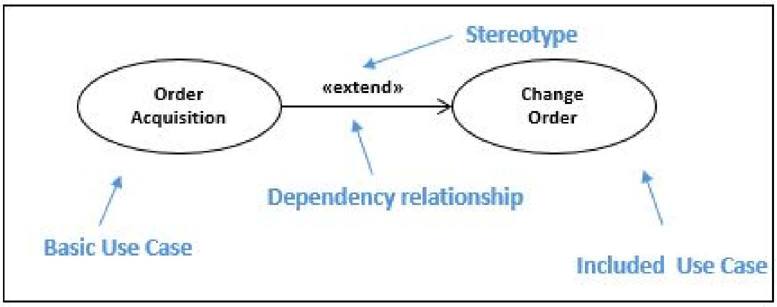


Figure 12.13: Example extension

- **generalization**: Use cases can inherit the characteristics of another Use father houses. In inheritance between use cases, the child use case inherits the behavior and the meaning from the father and in addition it adds the specific characteristics of her. Therefore it will be possible to apply the child use case, where it is possible to apply the parent. Generalization is represented with a solid line that has a triangle open pointing to the father. The relation of Generalization can also exist between Actors.

12.9.5 Scenario

A scenario is an instance of a use case or more simply a particular "execution" of a use case. A scenario explicitly identifies and describes the system actions and events for a particular use case.

Each use case is characterized by a basic scenario (typical sequence of steps) and by an unspecified number of variants that can be added also subsequently.

Each use case should be accompanied by a set of scenarios:

a main scenarios, within which everything works correctly;

b secondary scenarios, within which exceptions are described with any problems or malfunctions.

During the analysis as many scenarios as are necessary to understand the correct functioning of the system should be defined, as well as the exceptions that are considered

significant. Each scenario can have a series of conditions necessary for the start of the use case and a number of final conditions for the exit. Both have to be observable by the user and are intended to clarify the flow as much as possible.
Within each use case and/or scenario it is possible to:

- identify a normal or basic sequence of transitions.
  To identify transitions means to give an answer to the following questions:
  - what can go wrong?
  - what cannot happen?
  - what resources can be blocked?
  - number cases e.g. such as A1, A2, A3 ...
  - they can be described in detail, but it is usually enough to identify them is usually enough.
- identify various alternative sequences/variants/optional flows that structure the basic flow.
  To structure a flow means:
  - to define where an alternative flow begins;
  - to define the condition for this alternative [trigger];
  - to describe the behavior of the system during the alternative flow;
  - to describe where the basic flow resumes;
  - to create a section in the basic flow to be used as a reference in each alternative flow (optional).

12.9.6 Activity Diagram

The UML Activity Diagram is very similar to a flowchart. In fact, it shows the steps (called activities), the decision points, and the branches that intervene in the flow of a program. This diagram is useful for showing what happens in a business process or in an operation and is considered as a part of the system analysis.

An Activity Diagram is designed to be a simplified view of what happens during an operation or process. The Activity Diagram also highlights such activities.

Each activity is represented by a rectangle with rounded corners (it is narrower and more oval than the rectangle icon of the state diagram). The processing of an activity leads to its completion and then an automatic transmission to the next activity takes place. An arrow represents the transition from one activity to the next. The Activity Diagram also has a starting point represented by a filled circle and an ending point represented by a bull's-eye.

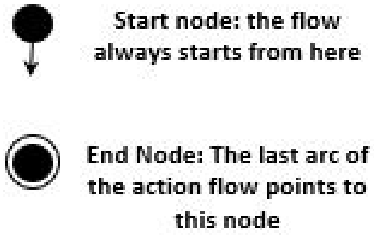

Start node: the flow always starts from here
End Node: The last arc of the action flow points to this node

Figure 12.14: Graphic representation of the starting/ending point of the process

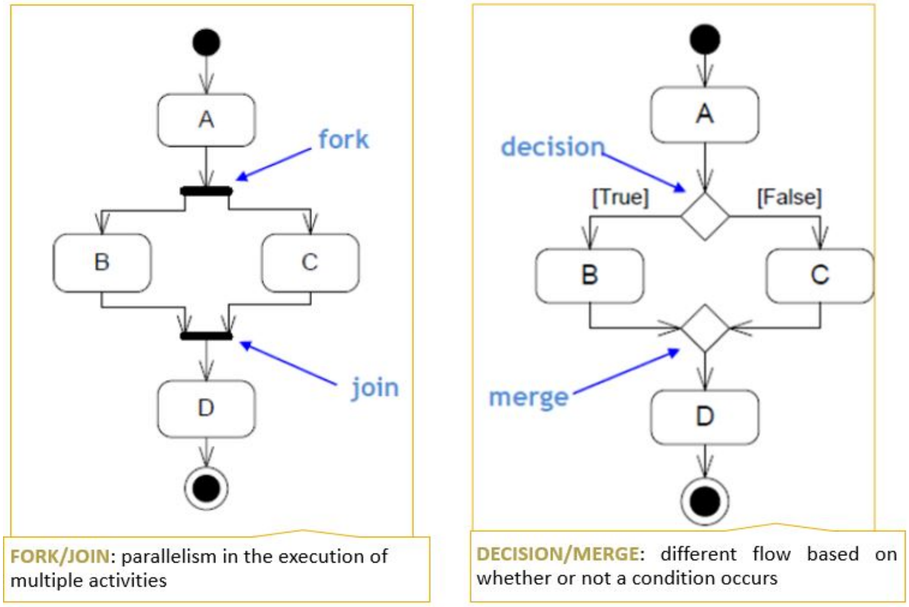
Following the various components of an Activity Diagram are described in detail, together with their representation:

- decisions;
- competing paths;
- signals;
- swimlanes.

12.9.7 Decisions

A decision point can be represented in three ways; the choice of the method to be used is entirely entrusted to the analyst.

The first way to represent a decision is to show the possible paths that derive directly from an activity. The second way consists in representing the transition of an activity with a small rhombus followed by all the possible paths of the flow. The last way to represent a decision is to indicate the condition with an instruction enclosed in braces next to the appropriate path.



FORK/JOIN: parallelism in the execution of multiple activities

DECISION/MERGE: different flow based on whether or not a condition occurs

Figure 12.15: Decision - Merge

The figure shows the different flow followed according to whether a condition occurs or not. Imagine you have to go to work. The steps that describe this process will be: open the car, insert the keys in the dashboard and then choose between the following two situations:

A the car starts up regularly

B the car does not start

These two possible cases will produce two different possible activities:

I drive a car

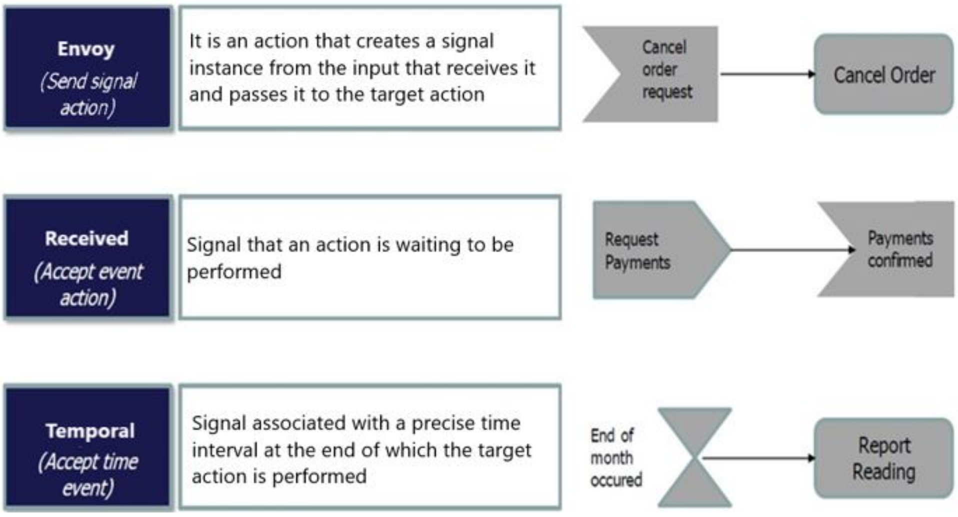
II go to the bus stop (or take a taxi, bicycle, walk, etc.).

12.9.8 Competing Paths

During the modeling of activities, there is often a need to separate the transitions into two paths that are executed at the same time (*concurrently*).
Such a subdivision is represented by a solid bold line, perpendicular to the transition that shows the paths that occur outside of the line. To represent a subsequent ”merge” of the two activities, the paths converge towards another bold line.

12.9.9 Signals

During a task sequence, a signal can be sent. When received, the signal causes the creation of an activity. The symbol for sending a signal is a convex pentagon and the symbol for representing the received signal is a concave polygon.



Envoy *(Send signal action)*	It is an action that creates a signal instance from the input that receives it and passes it to the target action
Received *(Accept event action)*	Signal that an action is waiting to be performed
Temporal *(Accept time event)*	Signal associated with a precise time interval at the end of which the target action is performed

Figure 12.16: Type of signals

12.9.10 Swimlanes

The Activity Diagram also defines the ability to view roles. To do this, the diagram is separated into parallel segments, called swimlanes. Each swimlane displays the name of a particular role at the top and shows the activities of that role. Transitions can also involve two swimlanes. It is possible to combine the Activity Diagram with the symbols of other diagrams to produce a hybrid diagram.

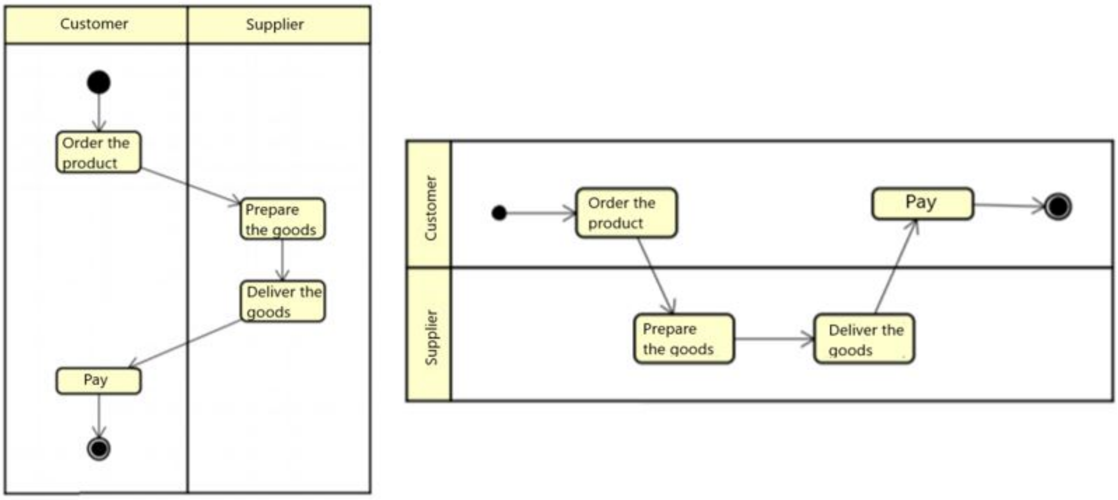


Figure 12.17: Horizontal/vertical swimlanes

Sequence diagram

The UML uses the Sequence Diagram to represent the communication between the objects of a process. The Sequence diagram enables the modeling of the communication between objects, in relation to the time and following a precise order. The Sequence diagram consists of objects represented such as rectangles bearing a name (with the name underlined) messages represented by solid lines with an arrow at their end, and time represented as a vertical progression.

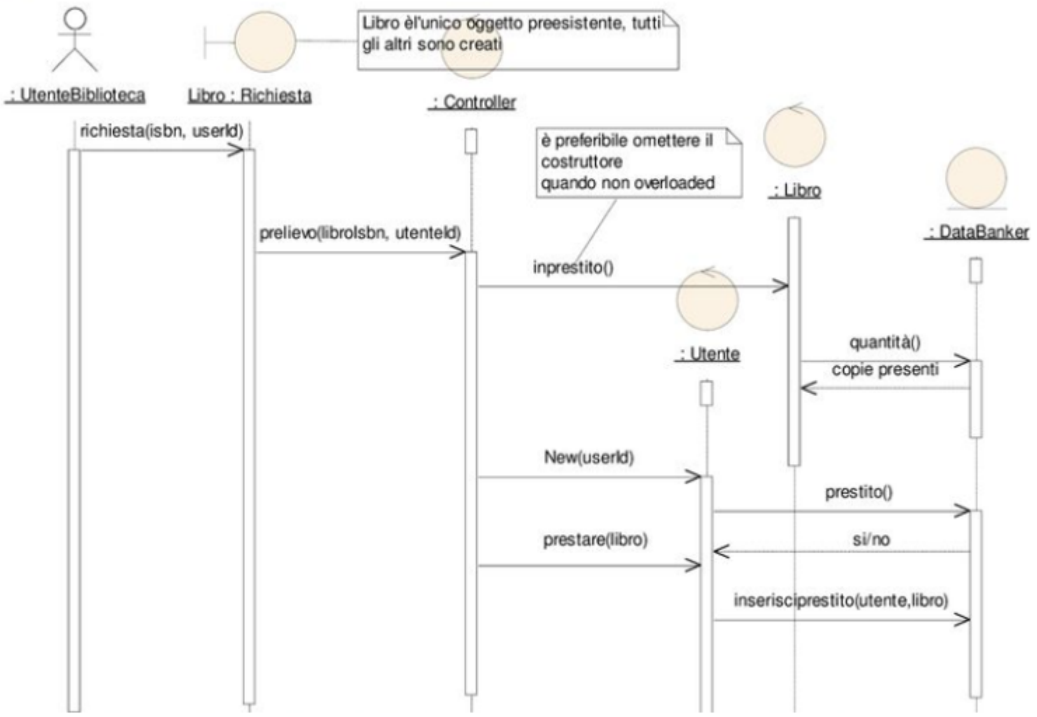
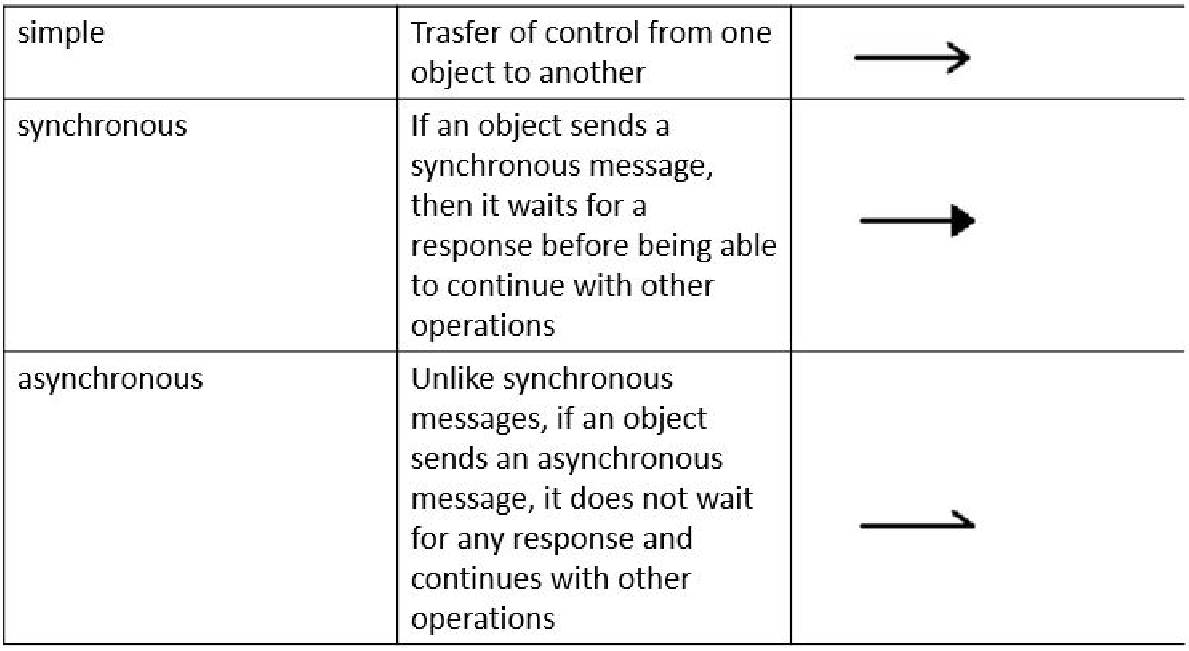

: UtenteBiblioteca
Libro : Richiesta
Libro èl'unico oggetto preesistente, tutti gli altri sono creati
: Controller
richiesta(isbn, userId)
è preferibile omettere il costruttore quando non overloaded
: Libro
: DataBanker
prelievo(libroIsbn, utenteId)
inprestito()
: Utente
quantità()
copie presenti
New(userId)
prestito()
prestare(libro)
si/no
inserisciprestito(utente,libro)

Figure 12.18: Sequence Diagram Example - (Real italian diagram case)

Following are the components of a sequence diagram:

- **objects**: the objects are drawn near the top of the diagram and are arranged sequentially from left to right. They can be placed in any order that makes the diagram easier to understand. From each rectangle departs a dashed line going downwards, called the “lifeline”. Along the lifeline is a small rectangle called “activation”. The activations represent the execution of an operation for which the object is in charge. The length of the rectangle, on the other hand, represents the duration of the activation.

- **messages**: a message that travels from one object to another is drawn starting from the lifeline of the object from which the message starts and arriving on the lifeline of the object to which the message is directed. Sometimes an object can send a message even to itself; this is a message that starts from its lifeline and arrives at the same lifeline. The following table identifies all the possible messages defined in UML for sequence diagrams, each of which has a different graphic representation.



simple	Trasfer of control from one object to another	→
synchronous	If an object sends a synchronous message, then it waits for a response before being able to continue with other operations	➞
asynchronous	Unlike synchronous messages, if an object sends an asynchronous message, it does not wait for any response and continues with other operations	⇀

Figure 12.19: UML messages

- **time**: when viewed in a vertical direction, the sequence diagram represents the flow of time. Time graphically starts at the base of each object and continues down to the bottom. For example, a message that is closer to an object than to another (with respect to the vertical direction), occurs earlier in time.
- **frame and operators**: frames group a set of messages together. Each frame is characterized by an operator that defines the type of interaction to be applied to the messages.

  a **Loop** - the messages enclosed in the frame are executed several times within a loop;

  b **Alt** - some groups of messages contained in the frame are performed alternatively to each other;

c **Opt** - the messages enclosed in the frame are executed optionally;

d **Par** - the groups of messages enclosed in the frame can be executed in parallel with each other, as long as the precedence between them in each compartment is respected.

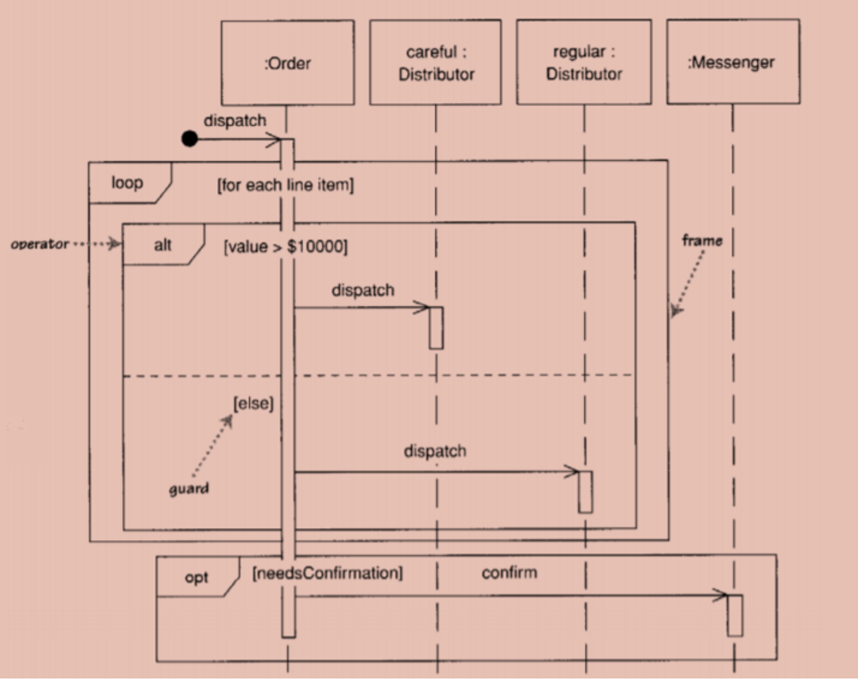


Figure 12.20: Frame & Operators

State diagram

A State Diagram shows the states of a single object. Since a system interacts with users and, possibly, with other systems, the objects that compose the system go through changes that help foster these interactions.
For example, when a switch is pressed, a lamp changes its state from off to on (or vice versa). State Diagrams display the states that an object can go through and the transient steps between these states. In addition, these Diagrams show the starting point and the ending point of a sequence of state changes.

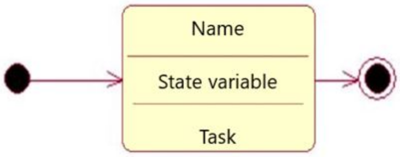


Figure 12.21: State diagram

Graphically the states of a State Diagram are represented with a rectangle with rounded corners. The symbol showing the transition from one state to another is drawn using a solid line ending with an arrow.
A solid black circle represents the starting point of a sequence of states. The end point is represented by two concentric circles in which the innermost circle is filled with black (such a figure is also called "bull's eye").
The previous figure showed an example of the described representations. As shown in the figure, more details can be added by dividing the rectangle into three separate areas. The top area contains the name of the status, the central area contains the status variables (timers, counters, dates, etc.) and the bottom area represents the activities (i.e. what happens when the system enters a certain state and what happens, on the other hand, when the system leaves that state.

Entity-Relationship Diagram (E-R Diagram)

During the analysis phase, the functional analyst (or the technical analyst, when this role is required by the project) often carries out a modeling of the available data, as a first step towards satisfying the business requirements necessary for the creation of the product/service. The most commonly used graphic model for this aim is the E-R (Entity-Relationship) Diagram. Invented in 1976 (Chen), it is the most used tool to build an application-independent conceptual model of data.
Modeling data means building a simplified representation of the observed reality, identifying the characterizing elements and the links between them.
The design of a data model takes place on 3 levels:

I conceptual →E-R model[7];

II logical →Tables (SQL);

III phisical →Database.

The advantages of such representation are the following:

- it graphically models the real world using only entities and relationships;
- it can be easily used to design relational databases;
- it is simple and easy to understand;
- it can be used as a work plan for database developers.

The elements that characterize an entity-relationship model are:

(a) **entity** - Classes of objects that have common properties and autonomous existence.

(b) **relationships** - A link between two or more entities. The level of the link is also defined as **relationship cardinality"** (the amount of instances of the arriving entity that are associated with the starting instance. There are three types of association):

  i one-to-one, or bijective (**1:1**);

  ii one to many, or simple (**1:N**);

  iii many to many, or complex (**N:N**).

(c) **attributes** - The properties of entities and relationships. The main property of an attribute is called Primary key (it is the set of one or more attributes useful to distinguish one instance from another for the same entity).

To sum up:

[7] concept detailed in the following paragraphs

- **E-R model** is one of the most solid approaches for modeling application domains in the IT environment.

- **entities** represent classes of objects that have common properties and an autonomous existence. Instance of an entity is an object of the class.

- **attribute** describes the elementary properties of entities or relationships. The domain of an attribute is the set of admissible values, that is, the values that the attribute can get.

- **relationships** represent the link between two or more entities.

- **cardinality** is specified for each entity participating in a relationship and describes the minimum and maximum number of times an entity can participate in the relationship.

- **cardinality of an attribute** can be specified for some attributes of an entity or relationship and describes the minimum and maximum number of values of an attribute associated with an instance of an entity [if the cardinality of an attribute is (1,1), it is omitted; the attribute can also be optional, in which case a zero must be specified in the cardinality: (0, N)].

- **identifier or primary key** needs to be specified for each entity of the conceptual scheme (or model) and describes the concepts that allow an instance of the entity to be identified without any ambiguity. In some cases, an identifier is made of one or more attributes of the entity.

- **external identifier or foreign key** sometimes the entity's attributes are not sufficient to uniquely recognize an instance and therefore other entities need to be involved in the identification.

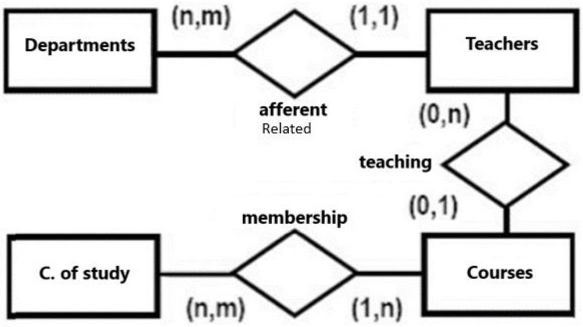


Figure 12.22: Example model E-R

Use Case Specification

A Use Case Specification (also called UCS, or more simply Use case) is a set of interactions between the system and one or more external actors, aimed at achieving a certain goal. An actor in an entity that is external to the system and that interacts with it: it can be a class of users, a role that users can play, a part of the system that has not yet been developed, a trigger, etc.

A use case is initiated by an actor, with a particular goal, and successfully ends when that goal is achieved. It describes the sequence of interactions between the system and the actors necessary to achieve the service embedded in the objective. A complete list of use cases:

I it specifies all the different ways of using the system;

II defines all the required system behavior;

III delimits the system scope.

In order for the Business Analyst to carry out a complete analysis of the project to be carried out, the following is required:

A use case diagrams →Overview, relationships between actors and use cases, structuring of use cases.

B use case templates/Specification →Textual specification of individual use cases;

C Mock-ups →Information and methods of interaction on the graphical user interface.

The use case specification identifies and describes the elementary scenarios of use of the system by the actors who will interface with it. It is necessary to evaluate each requirement by focusing on the actors who interact with the system, considering three different levels of interactions:

1. high level - representation of the E2E (end to end) process relating to the use case;
2. medium level - representation of the Use Cases through the UML methodology;
3. low level - representation of the basic rules necessary for the implementation of the process activities associated with the actors involved.

The drafting of this document is the responsibility of the BA and partly prepared with the help of Business (figures who within the project provide the necessary requirements for the creation of the functionality) and IT (technical figures who concretely implement the functionality through a software development). It is a technical document developed after the requirements collection, structured as follows:

1. Initial sections	Keep track of comments and review suggestions, update the document with a new version at every significant change
2. Definitions, Acronyms and Abbreviations	Report descriptions of abbreviations and acronyms used within the document
3. References	Report the documents quoted within the analysis and any file recovery paths
4. Short description	Expose the contents of the UCS in a concise but complete manner
5. Actors and objectives	Describe the resources involved in the process and their objectives

6. Flow of events	Describe the combination of activities that may occur during the execution of the functionality; in particular, the activity diagram, the preconditions, the basic flow, the alternative flow, and the post-conditions
7. Other details	Report any rules, logic, and data of the flows. For the specific subdivision of the sub-paragraphs see Annex ”X”

Table 12.1: UCS structure

On the content side, a UCS should be accompanied by:

1. **UML diagrams and Flow Chart** → Graphical representations that allow to have a graphic view of all the functions that each individual actor can perform (UML diagrams) and to detail the logical processes to be undertaken according to a specific activity that an actor chooses to carry out (Flow Chart).

2. **preconditions** → Constraints/bases necessary to ensure that the use case can be triggered.

3. **basic flow e alternative flow** → List of interactions that take place between the actors and the system.

4. **post conditions and supplementary specifications** → The former are created following the execution of the use case. The latter are the purely technical specifications that refer to the use cases and detail certain technological peculiarities to the IT department.

They are used to:

- discuss and define functional requirements;

- define the interfaces of a (sub) system;

- segregate and structure the functionalities;

- plan and monitor development;
- plan tests from a functional perspective;
- support a usability engineering plan (improve the use of the product/service);
- specify workload and workaround conditions.

They are not used to:

- describe procedures in the information processing flow;
- segregate the information processing flow within the system;
- design.

UCS are easy to maintain. However it can be difficult for the analyst, over time, to keep the document structure 'clean', due to the continuous introduction of improvements to the product/service (Change Request).

The mockup

The mockup is a document that shows:

a. the prototype of the graphical interfaces exposed to the user;

b. the navigation structure of the pages;

c. the layout of the individual pages.

The BA and the User Experience Analysts are accountable for the creation of this document. It is a technical document developed after or simultaneously with the preparation of the UCS, in order to:

1. encourage the involvement of users and domain experts;
2. guarantee concreteness of the representation;
3. obtain a low cost of evolution;

4. stimulate criticism and discussion.

During the life cycle of the project, the mockup is modified passing through various finalization phases:

I early prototype (low-fidelity);

II validation of requirements (high fidelity).

Change Request (CR)

During the implementation or release phases of a project, it may happen that the customer makes unexpected requests. In these cases, with a positive mindset of constant acceptance of client's requests, the project manager is in a position to evaluate whether it is possible to satisfy them.

Whatever the evaluation of the project manager is, the optimal solution is always achieved through an appropriate management of change requests. For this reason it is important to ensure a proper control flow of these requests.

Change requests may involve modifying the processes described in the UCS or not.

In the first case, it is appropriate to write new use cases; in the second case, instead, it is necessary to write a new Change Request specification (CRS) that reflects the document structure of the Feasibility Specification document (FSD), but in a scaled down way. A typical change request management flow is based on the following activities:

- submission of the request to the project staff;
- verification from the staff and request for clarifications to the customer;
- in-depth analysis of the change request in order to identify costs, effort, and implementation risks;
- sharing of the analysis between client and project manager in order to decide on the approval of the change request;
- implementation of the request.

Testing phase

After all the phases of the functional analysis have been carried out (collection of requirements, analysis, development, etc.), it is necessary to test the implemented functionalities.

12.9.11 Testing principles

Practice has identified seven key principles for the testing activity:

principle 1 – Testing shows the presence of defects

Testing can show the presence of defects, but cannot prove their absence. Testing reduces but cannot eliminate the likelihood of undiscovered defects in the software.

principle 2 – Exhaustive testing is impossible

Exhaustive testing (i.e. the testing of all combinations of inputs and preconditions) is not feasible, except for extremely simple cases. So instead of focusing on exhaustive testing, the BA should focus on priority analyses and major risks testing.

principle 3 – Anticipate testing as soon as possible

In order to detect defects promptly, testing activities need to start as early as possible during the software or system life cycle and should be focused on well-defined objectives.

principle 4 – "Clustering" of defects

The testing effort should be directed according to the density of defects (expected, or measured in retrospect) of each module. A small number of modules usually contains most of the defects discovered during pre-release testing, or is responsible for many negative operational outcomes (this is because the defects tend to form agglomerates, called "clusters", concentrating in a few modules, especially in the first development phases).

principle 5 – "Pesticide paradox"

If the same test cases are repeated over and over, presumably no new defects will be found. In order to overcome this so-called "pesticide paradox" and potentially find more flaws, test cases need to be revised and corrected, as well as new tests should be written to stimulate different parts of the software and system.

principle 6 – Testing is context-dependent

Testing is performed differently in different contexts. For example, a safety-critical software is tested differently from an e-commerce website.

principle 7 –The cliché of the absence of errors

Finding and correcting defects does not help if the developed system is unusable and does not meet the user's needs and expectations. The absence of errors is a false belief. According to Principle 1, a tester can only say, at best, that he/she cannot find any other error in a software, but can never say that the software does not contain any errors.

12.9.12 Main types of tests

Tests are normally distinguished between four main types:

- usability tests;
- safety tests;
- functional tests;
- performance tests.

Usability Tests – a set of techniques that allow to study the behavior of real users dealing with real products or their prototypes, in order to:

- correct critical issues related to the graphic interface of the software during the design phase;
- in the design phase, consider any reasons for difficulty shown by the users, after studying the use of the tools and understanding how users move and think in front of the application.

Safety Tests – in close connection with usability tests, they consist in an evaluation of the product conducted by some specialists or more appropriately by a sample of users, targeting categories of users who already use or could potentially use the product. The methods for verifying a product's safety of use are based on collecting information relating to the possible modalities of human-product interactions within a specific context of use. Safety Tests allow to identify and analyze users' behaviors and needs, as well as the type and frequency of potential errors that might happen when performing the required tasks.

Functional Tests - they represent the most common type of tests, as they tend to investigate any defects or regressions of the application in real conditions of use. In this scope, a further distinction is often made between two different types of functional tests:

- black-box →performed by accessing the software only via the user interface or via communication interfaces between processes. Often the "black box" testing is executed by people external to the development group or, in other cases, it can be conducted in an automatic mode, with undoubted benefits on costs and quality.
- white-box →carried out directly on the code. In order to perform a "white box" test, the tester needs the source code.

Regardless of whether it is automatic or manual, a functional test should always have an increasing gradation: it should first individually examine all the software modules, the routines, or the limited sets of routines (preferably using a white-box test) and then move on to testing the progressive aggregation of the modules (preferably using a black-box test), until the whole system is tested.

Performance tests - their aim is to verify whether the application meets certain performance requirements. The performance test category includes four different types of tests:

- **performance tests** - they check the execution timings in different conditions;
- **load tests** - they measure the reactions of the system to an increase of the load, highlighting possible regressions that could occur under particular circumstances;
- **durability tests** - they verify the robustness of the system over time;
- **stress tests** - they verify the behavior of the system in the breaking phase.

12.9.13 Test Process

During a project, in addition to focusing on the analysis and development of the required functionalities, it is advisable to test them, ensuring the correspondence between what is expected and what is obtained.

The test execution is the most visible part of a testing process. However, in order to be effective and efficient, test plans have to include the amount of time to be spent in planning, in designing test cases, in preparing for execution, and in evaluating the results.

The foundations of the testing process consist in the following main activities:

Planning and control:

- it includes the act of defining the testing objectives and the activities specifications that enable to achieve the "mission" of the project;
- it consists in a phase of continuous comparison between planning and work progress, documenting the current state and highlighting deviations from what was planned;
- it includes the implementation of countermeasures to enforce the "mission" and objectives of the project: controlling implies that the testing process has to be conducted throughout the whole project. The test planning takes into account the results of the monitoring and control activities.

Analysis and design:

- Activity that transforms the objectives into appropriate test conditions and test cases.
- Main tasks:
  - review of the basis of the test (such as requirements, architecture, design, interfaces);
  - assessment of the testability of the test bases and test objects;

  - identification and prioritization of the test conditions based on the analysis of the test units, specifications, behavior, and structure of the software;
  - design and prioritization of the test cases;
  - identification of test data needed to support the test conditions and test cases;
  - design of the test environment and identification of each element of the required infrastructure and of the activity support tools.

- This activity can be carried out by the functional analyst, by the technical analyst, or by the Quality Assurance, if the project requires the presence of such figures.

Implementation and execution:

- Implementation and execution are activities during which the test procedures (and any scripts) are implemented, by combining the test cases in a particular order and including any other information necessary for the execution: in this way the test environment is set up and the test cases can run. The implementation and execution consists of the following main tasks:
  - development, implementation, and prioritization of the test cases;
  - development and prioritization of the test procedures, creating test data, preparing (if possible) the necessary kits (test harness), and writing automated test scripts;
  - creation of test suites from the test procedures for efficient execution;
  - check whether the test environment has been prepared and configured correctly;
  - execution of the test procedures either manually or using execution tools, in accordance with the planned sequence;
  - saving of the outcome of the test cases' execution in appropriate logs and recording of the identifiers and versions of the software being tested, as well as of the test tools and testwares;

  - comparison of actual results with expected results;
  - indication of discrepancies in the form of incidents and their analysis in order to establish their cause (for example, a defect in the code, in the test data, in the test document, or an error in the way the test was runned, etc.) .
  - repetition of the test activities as a result of the actions taken for each discrepancy.

Results report. Test conclusion activities The test conclusion activities are aimed at collecting the data produced by the completed test to consolidate the user experience, the testwares, the facts, and the numbers. These activities should be carried out when a software system is released, a test project is completed (or canceled), a milestone has been reached, or a maintenance release has been completed. The test conclusion activities consist of the following main tasks:

- check of which planned deliveries have been completed;
- closure of incident reports and opening of change reports for each incident that remains open;
- preparation of a system acceptance documentation;
- finalization and archiving of the testwares[8], test environment, and test infrastructure for subsequent reuse;
- delivery of the testwares to the maintenance organization;
- analysis of the lessons learned, to be used for future releases and projects;
- use of the collected information to improve the testing maturity.

Although these activities are logically sequential, they could overlap or take place simultaneously in the process.

[8] Refer to the definition in the Glossary

12.9.14 Test plan

An effective test plan should indicate:

1.Test case ID	the unique code of each test case
2. Path	the path to be validated
3. Method	the name of the method being validated
4. Description	the test case and the specific behavior subject to validation
5. Dependencies	the dependencies on other test cases
6. System Status	the system status needed to run the test case
7. Input	the value to be provided to the system, needed to carry out the test case
8. Notes	any additional information useful for understanding or executing the test case
9. Procedure Execution	the specific procedure to perform
10. Expected value	the value that is expected after the execution of the test case
11. Obtained value	the value obtained at the end of the test
12. Impacts	the impacts that the positive result of the test case may have on other tests

Table 12.2: Elements of the test plan

The following image shows the Testing process performed in the event that the figure of the Tester (QA column) differs from the BA.

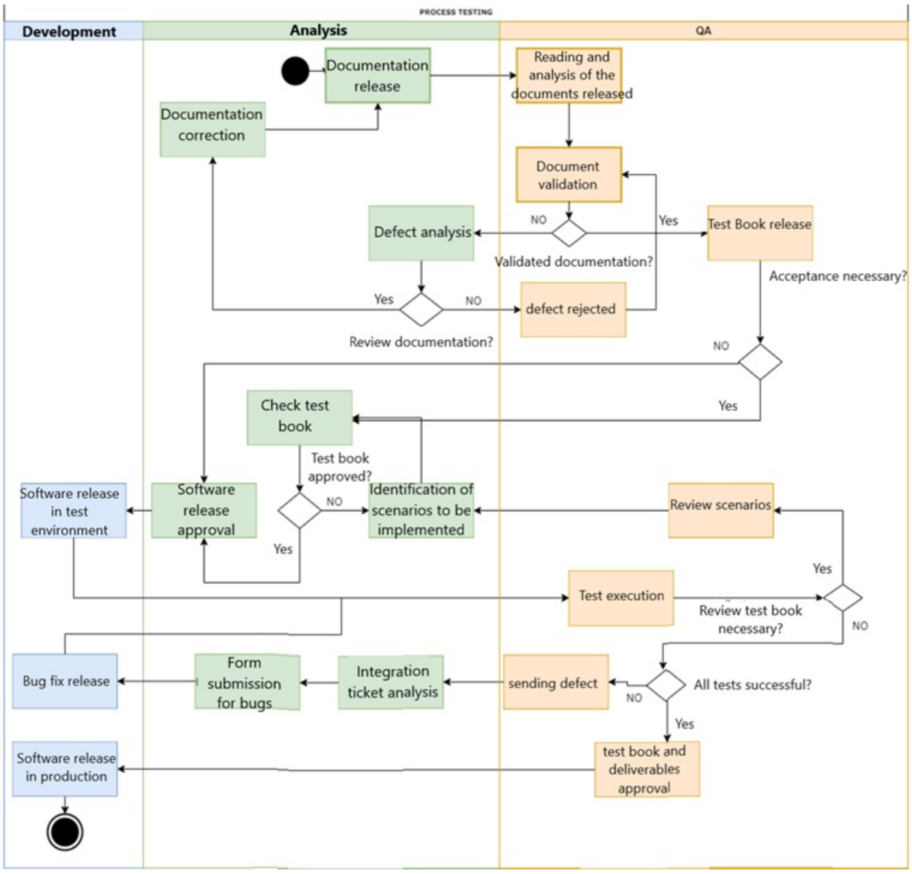


Figure 12.23: Workflow testing

User manual or online help

Another document of fundamental importance and within the competence of the business analist (BA) is the User Manual or online help. The aim of this document is to provide the end-user with the necessary assistance until he/she reaches a complete autonomy of use. In fact, the manual allows the user to solve minor problems in a coherent and simple way.

The User Manual is a document especially used in ICT projects dedicated to the Operations area. Some other times it is made for customer support functions (e.g. Customer Care). In large projects, in addition to the drafting of the user manual, one or more training days are often provided to the project Actors interested in knowing how the product/service works.

To achieve its aim effectively, the user manual should:

- have a **well-structured index** to easily find what the user is looking for;
- be **rich in images** but with **essential contents**;
- be constantly updated in order not to create any confusion.

In some cases, when the application is also intended for an audience of technicians, the user manual needs to be integrated with the application management manual:

- **User Manual** provides a general description of the application and an operational guide to the use of the product's individual functions;
- **Application Management Manual** is a tool useful for the installation and utilization of the application. It is a manual aimed at technical personnel.

Chapter 13

The functional analyst in the ICT field - Conclusions

All team members, including the functional analysts with technical skills, can be classified into professional sub-levels: Junior and Senior. With no regards of the levels of experience, the functional analyst can be defined as the person who studies and analyzes the characteristics of the clients, their added value, their pre-established macro-objectives, translating these into ad-hoc projects equipped with the required functionalities, also studying and analyzing the competition to verify available spaces or margins for the products/services development and evolution.

The functional analyst comes into play whenever the customer needs a new software application or wants to make changes to an already existing/started one. He/she carries out all the necessary analyses to verify the feasibility of the project and to finally give all the appropriate directives to the technical analysts that will take care of the implementation process.

The figure of the functional analyst needs numerous skills, including:

- interpersonal skills
- inclination to work in a team

- organizational capabilities
- creativity
- availability
- fast learning

It is very important to emphasize that a person who works in this role does not typically work alone, but gets in contact with numerous subjects covering various positions within the company, above all the customer and the technical analysts.

13.1 Phases

The typical phases of a functional analysis are:

1. definition of the problem, which consists in fully understanding the problem that the system is called to solve;
2. feasibility analysis, that aims to establish whether the objectives set for the development of the system are reasonable and achievable;
3. cost and benefit analysis, which preliminarily assesses the economic convenience of developing the software system, taking into account the expected costs and benefits that it can provide;
4. domain analysis, that consists in fully understanding the context or application domain in which the system will have to act;
5. requirements analysis, which consists in specifying in details the services, functions, and performances required for the system.

Through the requirements analysis, the functional analyst uses various documents that can help (both him and the client) to better understand the feasibility (FSD) and use cases (UCS) of a certain application. By elaborating clear graphs and diagrams, he/she provides the tools to the customer and/or the programmers to easily understand how the software works (by elaborating a user manual at the end of the developments) and how it should be created. Being aware of the entire process, the functional analyst is also able to test the application.

13.2 The tools

REQUIREMENTS ANALYSIS:

reports the information collection and analysis phases, providing the basis to discuss the actual feasibility of the project and whether the concrete realization, validation, or maintenance of a certain product/service is possible. The requirements analysis normally takes place as a negotiation between different subjects:

- business representatives - describe what they expect from the functionality to be developed;
- functional analysts - interpret the ”desired” and describe the solution;
- technical analysts - develop/implement the described solution.

In concrete terms, the functional analyst behaves like an investigator, asking questions to the client to stimulate his/her reasoning, trying to better understand his/her idea and, if necessary, to propose a new one. In order to do this, the analyst draws up a checklist of questions, assumptions, or statements that will have to be answered with the help of the other parties involved (business contacts or technical analysts).

DIAGRAMS:

The most used tools are indicated as follows.

- Flowchart: they represent how a process operates step by step, highlighting the sequence of actions needed to complete the process and the steps to be taken in case of error.
- UML: they model the external behavior of a system in terms of available functionalities that the actors (users, other software systems, etc.) can use.

USE CASE SPECIFICATION:

An analysis document with which it is possible to study/highlight every single use case associated with a specific application. Like for the requirements analysis, a standard can be followed to draw up this document in a complete and exhaustive way. Clearly, such a standard may be subject to changes depending on the project, the specific case, and the functional analyst's preferences.

USER MANUAL:

A manual that describes the use of the software, specifying all its features. A good user documentation should:

- provide the user with assistance until the complete resolution of a problem in a consistent and simple manner;
- not be a source of confusion and therefore be constantly updated;
- have a well-structured index in order to easily access the desired contents;
- be rich in images, but at the same time with essential content.

TESTING:

A process that consists in the execution of the software by the tester, in combination with other service softwares or alone, in order to assess whether the behavior of the software complies with the requirements provided by the client and agreed with the working group (WG) and with all the stakeholders. The functional analyst takes care of creating the various scenarios (UCS) and testing the solution, first in a purely design-field and then, when ready, also with the customer. In software testing, the functional analyst mainly deals with planning the tests that will be carried out, deciding how many and what types of tests to perform, as well as what steps each test requires.

Part III

Section 3

This section will highlight the steps followed by a Business Analyst for the creation of the main functional documents within a project. As a rule, the projects analyzed below were carried out in different work fields. In particular, for each example document associated with a project X, the activities carried out by the functional analyst will be described with presentation of the steps followed for the realization of the required documentation.

Chapter 14

Project 1

The purpose of the following sample project is the creation of a intranet, which is useful both to achieve the company's objectives and for the growth of the human resources. In particular, the Business Analyst (BA) is provided with the mockup or online help that allow to analyze the **AS IS** features available at the time of the analysis, which are indicated below:

- learning;
- dashboard →used to quickly share 'Information' with colleagues;
- recreational.

The business requires these features to be integrated with some additional tasks, indicated as **TO BE** features, required by the project. Below is the list of resources involved and assigned functions:

Resources	**Assigned functionalities**
XXX Developer	1. News section on the home page; 2. Social space for employees
XXX Developer	1. Interactive animation for Company history; 2. Electronic library for document sharing (no customer documents)
XXX Developer	Recreational section (interactive games, quiz for employees)
XXX Developer	Professional profile

Resources	**Assigned functionalities**
XXX Developer	Career section of the logged-in employee (current level, next levels to be reached, career/assistant names, etc.
XXX Developer	Learning/Academy - Automatization of the work done by the training manager
XXX Developer	1. Dashboard with centralized mini applications/functions; 2. Career levels section (general)
XXX Designer	Mockup - Considerer all the functionalities assigned to the developers
XXX Designer	Mockup - Considerer all the functionalities assigned to the developers

Table 14.1: List of TO BE functions required in the "Intranet" project

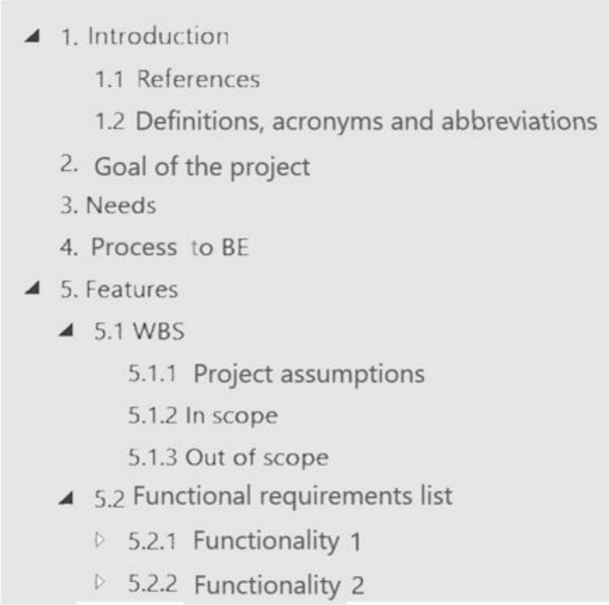
The BA is then asked to produce functional documentation associated with three distinct phases of the project:

1. **AS IS feasibility**:drafting of the feasibility analysis with the list of all the requirements extracted from the screens already created. Alternatively, description through the visualization of the Mockup, of the functionalities implemented to date and drafting of an FSD post;

2. **TO BE feasibility**: revision of point 1 of the document adding the requirements related to the new tasks. Alternatively, on the basis of the AS IS feasibility, a Change Request is made and its feasibility analysis is requested;

3. **drafting of the UCS**: the CR provides for the drafting of as many UCSs as there are specific use cases for each required TO BE functionality.

Starting from the generic template associated with the FSD, the functional analyst has to structure his/her document in relation to the project to be implemented.
The structure of the Feasibility Specification Document includes the sections shown in the following figure:



◢ 1. Introduction
1.1 References
1.2 Definitions, acronyms and abbreviations
2. Goal of the project
3. Needs
4. Process to BE
◢ 5. Features
◢ 5.1 WBS
5.1.1 Project assumptions
5.1.2 In scope
5.1.3 Out of scope
◢ 5.2 Functional requirements list
▷ 5.2.1 Functionality 1
▷ 5.2.2 Functionality 2

Figure 14.1: Generic paragraphs of a feasibility analysis

Based on the generic structure of an FSD, after the analysis of the provided documentation necessary for the contextualization of the project, the BA has to implement the following 'activities':

1st STEP – insert an introduction to briefly explain the purpose of the document, collecting, analyzing, and generally defining the needs of the stakeholders. The document should specify the goal that the stakeholders want to achieve, the target to whom it is addressed, and the reason why the need exists. Also, if the functionality requires it, in this paragraph it is possible to attach an English version of the document.

1.1 STEP – report any documentation from which the analysis started. In this example, the mockup of the AS IS functionalities was made available to the BA in order to understand the User Experience. Through direct meetings with the developers and designers, the BA has the possibility to define the technical requirements that are necessary for the realization of the project and its feasibility analysis.

1.2 STEP – list the references used in the preparation of the document (in this specific case: online help and interviews with the developers/designers in charge of the realization of the AS IS tasks).

2nd STEP - briefly describe the project, also referring to the context in which the product needs to be created;

2. Goal of the project

The Digitai Workplace,by aggregating information and services in response to the needs of employees, makes it possible to provide functions that make it possibleto:

- carry out everyday activities;
- satisfyingthe needs of employees and collaboration between workers;
- improve communication and organizational processes, thereby increasing productivity
- corporate and promoting business;
- sharing of the corporate culture at all hierarchica l levels.

3rd STEP - highlight the needs of the stakeholders and the needs that persuaded the program board to accept the project;

3. Needs

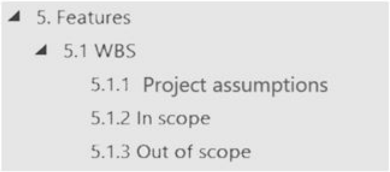
The needs of our stakeholders are:

- need for fast, practica land immediate communication;
- insta nt sharing of corporate documents;
- one-click training;
- and knowledge of the company organizat ion chart;
- digita ica lendar to improve the management of specific operations;
- online play spaces for amusement;

- organizat ion ofcorporate tournaments, to strengthen team spirit through sport.

4th STEP – the ‘TO BE Process’ part is usually filled in by the BA with a brief description of the TO BE functionalities requested by the stakeholders, that are usually not enough to meet the needs. In fact, these tasks will not be implemented during the drafting of the AS IS FSD. In this specific case, the paragraph may refer to the CR linked to the TO BE tasks, with consequent drafting of an ad-hoc document.

5th STEP – the Features include the following paragraphs:



◢ 5. Features
◢ 5.1 WBS
5.1.1 Project assumptions
5.1.2 In scope
5.1.3 Out of scope

Figure 14.2: Features paragraph list

a a. the WBS shows the project activities directly related to the obligations. In this specific case, the project activities directly related to the functionalities requested for the Intranet are reported;

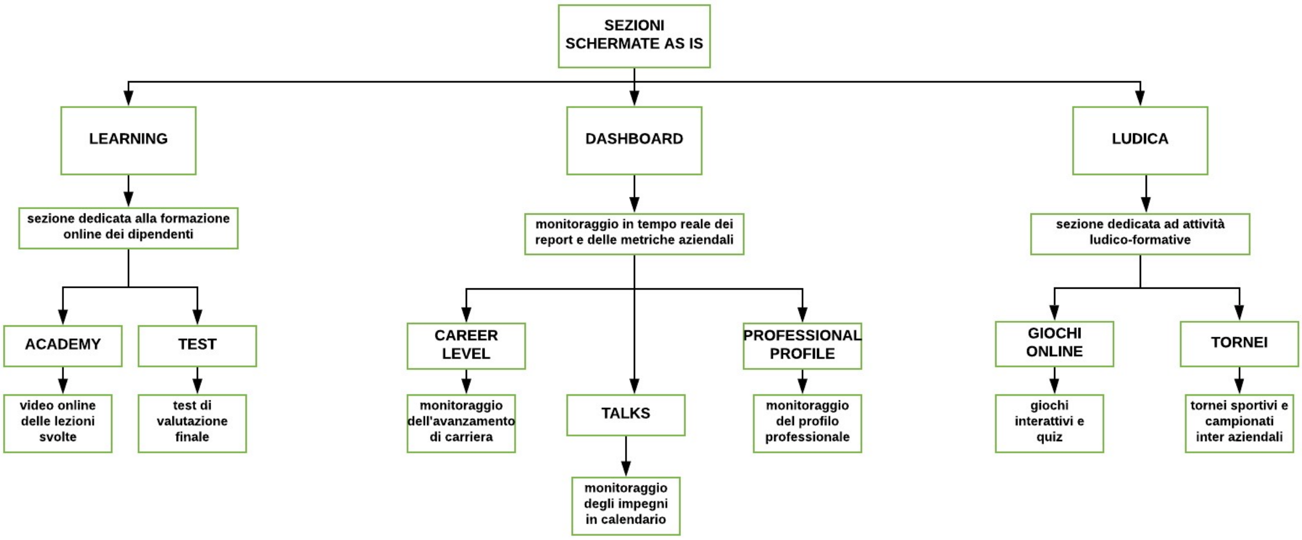

SEZIONI SCHERMATE AS IS
LEARNING
DASHBOARD
LUDICA
sezione dedicata alla formazione online dei dipendenti
monitoraggio in tempo reale dei report e delle metriche aziendali
sezione dedicata ad attività ludico-formative
ACADEMY
TEST
CAREER LEVEL
PROFESSIONAL PROFILE
GIOCHI ONLINE
TORNEI
video online delle lezioni svolte
test di valutazione finale
monitoraggio dell'avanzamento di carriera
TALKS
monitoraggio del profilo professionale
giochi interattivi e quiz
tornei sportivi e campionati inter aziendali
monitoraggio degli impegni in calendario

Figure 14.3: WBS AS IS functionality

b in the paragraph ”project assumptions”, the assumptions and constraints associated with the project are listed. In particular, the assumptions are the list of premises or preconditions that, if verified, enable the start of the project. The constraints, on the other hand, are a series of conditions that constrain the execution phase of a project;

c the paragraph “In scope” lists all the AS IS functionalities created for the project;

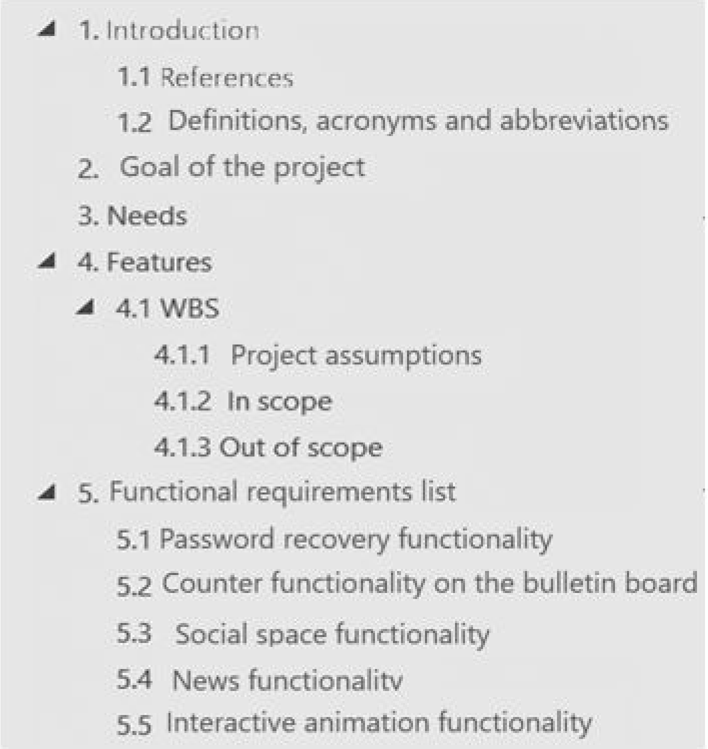
d the ”Out of scope” paragraph lists all the TO BE functionalities. In this specific case they are still considered ’Out’ of the software development process but they will be implemented later and described in the Change Request document;

6th STEP – finally, the BA has to list in detail the expected impacts for each functionality. In the specific case, the AS IS functionalities requested by the stakeholders and necessary to satisfy the Needs are listed.

FUNCTIONALITY	FUNCTIONAL REQUIREMENT
Sign In	Allow user access to the intranet
Change Password/Username	Allow the user to change the security login credentials
Edit Image	Allow the user to change their profile image
My profile	Section in the Hamburger Button, that allows the user to view their profile and associated information, visible to the public
Social Wall	Section containing attendance/absence/employee counters
Career Level	Allows you to visualize the sequence of career levels that the company provides and describes in terms of roles, competences and skills
Professional Profile	View the job profile of actual employees
Applications	Section that allows the management of HR applications
Talks	Allows you to consult/organize the calendar of activities
Lessons	It allows you to follow the lessons held during the training and download their contents
Messages	Chat between trainees and the trainee
Statistics	It allows you to develop statistics in relation to user groups and areas of interest
Publishing	Section dedicated to posts. in edit mode for the manager role, while in read-only mode for employees
Tournaments	It allows you to organize tournaments dedicated to employees with relative rankings
Log Out	Allows the employee to log out of the intranet

Table 14.2: Functional requirements of AS IS tasks

Within each project, a second analysis phase is usually planned, consisting in the integration of the AS IS FSD (creation of the TO BE FSD by filling in the ’TO BE Process’ paragraph) with the TO BE tasks described in the Change Request Document. This document (CRD) should have the same paragraphs of the AS IS FSD, with the sole exclusion of the ’TO BE Process’ paragraph and with the description of the new tasks planned within the CR.



◢ 1. Introduction

1.1 References

1.2 Definitions, acronyms and abbreviations

2. Goal of the project

3. Needs

◢ 4. Features

◢ 4.1 WBS

4.1.1 Project assumptions

4.1.2 In scope

4.1.3 Out of scope

◢ 5. Functional requirements list

5.1 Password recovery functionality

5.2 Counter functionality on the bulletin board

5.3 Social space functionality

5.4 News functionalitv

5.5 Interactive animation functionality

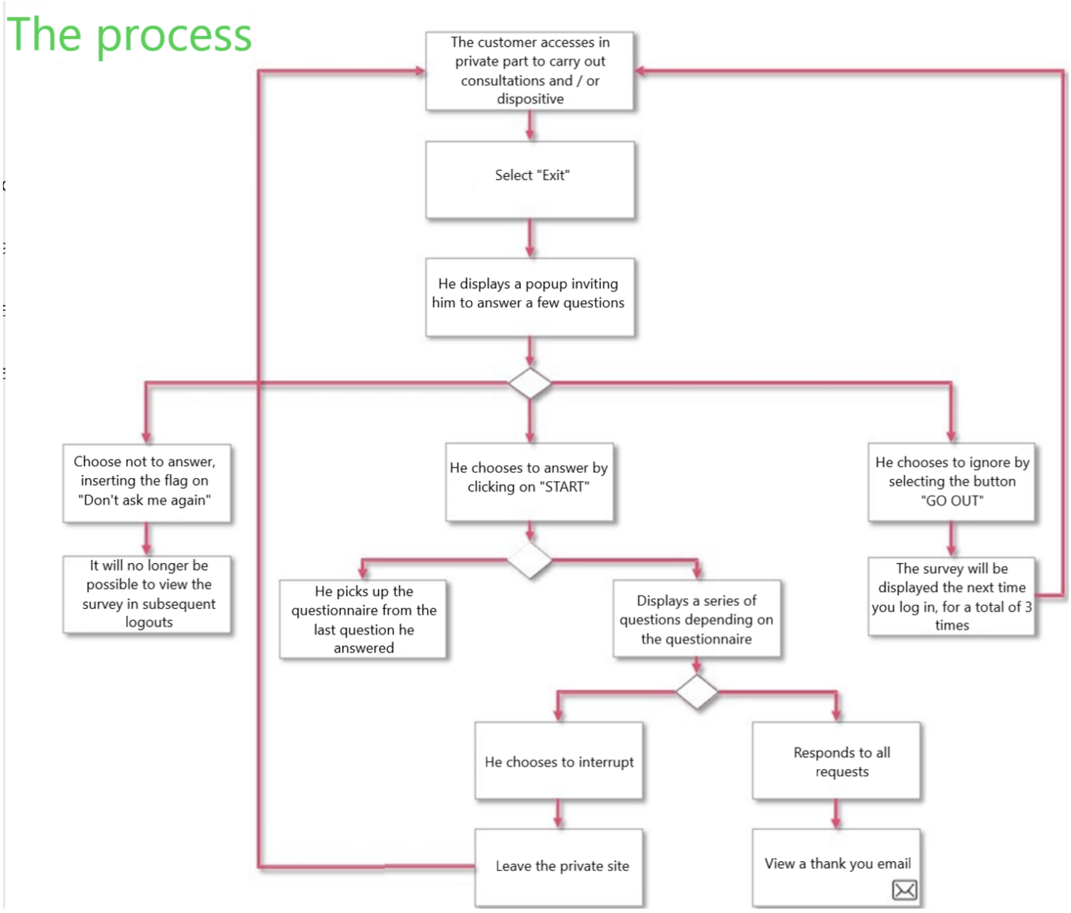
Figure 14.4: CRD paragraph with TO BE task list

In the last step of the analysis the Business Analyst creates as many UCSs as the functional requirements listed in the CRD. In this specific case of the INTRANET project, he/she has to develop as many UCSs as the number of new functions requested by the stakeholders with the aim of integrating the Digital Workplace with new functionalities.

Chapter 15

Project 2

The purpose of the third sample project called ”Survey” is to provide, on the private website of the XXX insurance institution, a series of surveys with closed-ended questions, aimed at loyal customers. This case focuses on the testing phase, one of the final stages of the analysis, within which the BA is actively involved, especially in iterative projects. Based on the process agreed with the project team and on the analysis carried out, the BA is asked to fill in the associated test plan.



The process
The customer accesses in private part to carry out consultations and / or dispositive
Select "Exit"
He displays a popup inviting him to answer a few questions
Choose not to answer, inserting the flag on "Don't ask me again"
He chooses to answer by clicking on "START"
He chooses to ignore by selecting the button "GO OUT"
It will no longer be possible to view the survey in subsequent logouts
He picks up the questionnaire from the last question he answered
Displays a series of questions depending on the questionnaire
The survey will be displayed the next time you log in, for a total of 3 times
He chooses to interrupt
Responds to all requests
Leave the private site
View a thank you email

Figure 15.1: Process linked to the ”Survey” project

The survey needs to be aimed only at some target customers, identified by:

- the level of seniority of their relationship with the company;
- the number of accesses made on the several channels (private website and app);
- the frequency of login in the last 6 months (private website and app).

Different types of surveys are proposed for each target:

- **survey 1** addressed to all customers who have made at least 3 accesses to the private website of the XXX insurance company and who have been customers for a minimum of 6 months and a maximum of 1 year;

- **survey 2** aimed at all clients with apps, who have made at least 3 accesses on the private website of the XXX insurance company and who have been customers for at least 1 year.

In order to profile the survey for each type of required target, the following matchers have been created that contain the necessary requirements to access one of the two surveys:

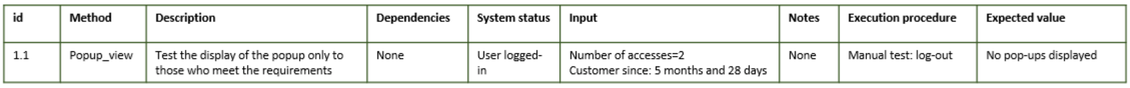
- FIRST TARGET →Customers who have been active with more than 2 logins or who have logged in for the first time and have been active for at least 6 months but no more than 1 year.
- SECOND TARGET →Customers who have been active for more than 1 year or who have logged in more than twice from the App in the last 6 months.

The test case report to be structured by the BA requires the completion of a similar table with the following fields:

- **id. test case** - it contains the code that uniquely identifies each test case;
- **path** - it indicates the linearly independent path subject to validation; it is used only for white-box tests;
- **method** - it contains the name of the method being validated; it is necessary to explicitly indicate the complete path leading to the definition of the method;
- **description** - it contains the textual description of the test case and precisely it indicates which specific behavior, or test case, is subject to validation;
- **dependencies** - it contains the list of test cases, the results of which affect the current test case, that is, the test cases referred to during the execution of the test case subject to validation;
- **system status** - it specifies the state in which the system needs to be in order to run the current test case;

- **input** - it indicates for each input the value that has to be supplied to the system in order to carry out the current test case;
- **notes** - it contains all the additional information that is not specified by the other fields, useful for understanding or executing the current test case;
- **execution procedure** - it indicates the procedure that has to be strictly followed in order to run the test case;
- **expected value** - it indicates the expected value at the end of the test case execution. Please note that "value" does not necessarily mean an output, but it can also be a change in the state of the system without producing any outputs. This field should be specified before the execution of the test case;
- **obtained value** - it indicates the value obtained after running the test case. Please note that "value" does not necessarily mean an output data, but can also be a change in the state of the system without producing output;
- **impacts** - it indicates the set of other test cases that get affected by a positive result of the current test case, that is, all the test cases in which the test case being validated is called.

In the specific case, after reading the documentation associated with the project and having identified the two target customers to whom the survey will be addressed, it is possible to complete the associated test plan.



id	Method	Description	Dependencies	System status	Input	Notes	Execution procedure	Expected value
1.1	Popup_view	Test the display of the popup only to those who meet the requirements	None	User logged-in	Number of accesses=2 Customer since: 5 months and 28 days	None	Manual test: log-out	No pop-ups displayed

Figure 15.2: Example of compilation of the test plan for the "Survey" project

Chapter 16

Project 3

The objective of the fourth sample project called ”Restaurant” is to create a dedicated Feasibility Specification Document (FSD). In this specific case the client wants to create a software system able to support the management of a restaurant. It should provide support for the customer data management, creating a card of each of the clients and in particular of those who booked or organized ceremonies at the restaurant. Based on the functional requirements listed in the supporting documentation, the BA has to fill in the following parts:

1. **Introduction** : indicate the references used for the drafting of the document and for the main definitions, acronyms, and abbreviations.

2. **Objectives of the project**: elaborate a description of the needs and of what it is hoped to obtain by achieving the objectives;

3. **Needs**: list the wishes of the project stakeholders;

4. **TO BE processes**: summarize the innovations expected with the implementation of the AS IS, which are preparatory to the definition of the future process;

5. **Features**: indicate the specific characteristics of the project. This paragraph requires a subdivision into sub-paragraphs containing the work breakdown structure (WBS), the project assumptions, the functions foreseen in the project budget (in

scope) and outside the project (out of scope). Finally, describe and analyze every single feature.

Specifically, the activities/steps performed for the development of the project are indicated as follows.

1st STEP – insert a brief introduction to the document explaining its purpose and defining the development of a software with the necessary characteristics useful to the management of the restaurant business.

- **1.1 STEP** – report any documentation from which the analysis started. In this case, refer to a slides deck with the list of requirements/features that the software has to possess in order to achieve the project goal and meet the needs of the stakeholders.

- **1.2 STEP** – list the references used for the drafting of the document, if any. In this case, refer to the list/database of loyal customers known by the management with any details regarding their preferences as recorded over time.

2nd STEP - briefly describe the project and the feasibility analysis that is being carried out, referring to the context in which the product needs to be created and with which the company objectives are associated.

3rd STEP - highlight the needs of the stakeholders and the needs that prompted the management to accept the project.

4th STEP – the 'TO BE Process' part is usually filled in by the BA with a brief description of the TO BE functions provided by the stakeholders, but that will not necessarily meet the needs. In this specific case, it can be filled in with the requirements that are out of scope at the time of the analysis.

5th STEP – in the 'Features' part the following paragraphs should be reported, which are focused on: "WBS", "Project assumptions", "In scope", and "Out of scope". In particular, in the last two paragraphs, the requirements that will be implemented are to be distinguished from those that will not be realized (for example, because they are not compatible with the project budget defined by the management).

6th STEP – finally, the BA has to list in detail the software impacts expected for each requirement, distinguishing between:

A. **USER REQUIREMENTS** - sentences in natural language (and diagrams) relating to the services that the system provides and to its operational constraints; they can be:

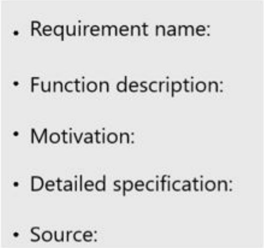
   i. **functional requirements** - phrases describing what the system has to do, how it will react to inputs, and how it will behave in various situations;

   ii. **non-functional requirements** - constraints on the services imposed by the system (such as constraints on response times, ...) or that should be considered during the development. They are distinguished between:

      - **product requirements** - they specify the behavior of the product (usability, efficiency, reliability, portability);
      - **organizational requirements** - they derive from the policies and procedures of the customer's and developer's organizations (e.g. process standards to be used, platforms, delivery requirements, etc.);
      - **external requirements** - they derive from factors external to the system and to its development process (such as interoperability, legislative, ethical requirements, etc.);

   iii. **domain requirements** - requirements deriving from the domain of application of the system and that reflect its characteristics and limits;

B. **SYSTEM REQUIREMENTS** - A structured document that provides a detailed description of the system's services. It can be part of the contract between the customer and the developer.



- Requirement name:
- Function description:
- Motivation:
- Detailed specification:
- Source:

Figure 16.1: Example of a format describing a requirement

List of Figures

1.1 Structures - peoples - processes 16
1.2 Processes 18
1.3 Primary and Support Activities 19
1.4 Hierarchy - Units - Systems 21
1.5 Organizational Structures 22
1.6 Example Functional O.S. 23
1.7 Example Divisional O.S. 24
1.8 Example Matrix O.S. 26
1.9 Objectives Organizational Structure 28
1.10 Organizational Structure - Objectives 29
1.11 Organizational model 30
1.12 Organization - Environment 30
1.13 Mechanical and Organic System 31
1.14 Organization chart 32
1.15 Company life cycle associated with the Organizational Structure 35
1.16 Evolution S.O. 35

2.1 7 S 37
2.2 Characteristics of Task 40
2.3 Walsh & Ungson Organizational Memory Process 47

3.1 Deming cycle 57

4.1 Process - Project 69
4.2 WBS 76
4.3 Levels of the WBS 77
4.4 WBS Regulation 847/2015 77
4.5 OBS 78
4.6 Levels of the OBS 79
4.7 Example of the OBS 79
4.8 RACI Matrix 81
4.9 RAM (an italian Example) 82
4.10 GANTT (an italian Example) 84
4.11 CBS 86
4.12 RBS 87

5.1 'Waterfall' method 92
5.2 Agile method 93
5.3 SCRUM method 94
5.4 Hybrid method 95
5.5 CPM method 96
5.6 DMAIC method 97
5.7 DMADV method 98
5.8 DevOps method 99

7.1 Fishbone diagram 105

9.1 The phases of a project team 125

10.1 The phases of a project ICT 126
10.2 Phases of Agile 129
10.3 Life cycle of a project 130

12.1 Process of carrying out a company project 149
12.2 BPMN Graphic Representation 150

12.3 Graphic representation of events 151
12.4 Graphic representation of activities 152
12.5 Graphic representation of a Gateway 152
12.6 Graphic representation of Connectors 153
12.7 Graphic representation of lanes 153
12.8 Graphic representation of artifacts 154
12.9 Graphic representation of the characteristic elements of the flow chart . . 155
12.10UML Diagram Hierarchy 159
12.11Class Diagram Example - (Real italian diagram case) 161
12.12Example inclusion 163
12.13Example extension 164
12.14Graphic representation of the starting/ending point of the process 166
12.15Decision - Merge 167
12.16Type of signals 168
12.17Horizontal/vertical swimlanes 169
12.18Sequence Diagram Example - (Real italian diagram case) 170
12.19UML messages 171
12.20Frame & Operators 172
12.21State diagram 173
12.22Example model E-R 176
12.23Workflow testing 188

14.1 Generic paragraphs of a feasibility analysis 198
14.2 Features paragraph list 200
14.3 WBS AS IS functionality 201
14.4 CRD paragraph with TO BE task list 203

15.1 Process linked to the ”Survey” project 205
15.2 Example of compilation of the test plan for the ”Survey” project 207

16.1 Example of a format describing a requirement 211

List of Tables

1.1 Functional structure strengths/weaknesses 24
1.2 Divisional structure strengths/weaknesses 25
1.3 Matrix structure strengths/weaknesses 27
1.4 Horizontal structure strengths/weaknesses 27

4.1 Core activities of a project 75

8.1 Structure of a PID 112

12.1 UCS structure 178
12.2 Elements of the test plan 187

14.1 List of TO BE functions required in the "Intranet" project 197
14.2 Functional requirements of AS IS tasks 202

Acknowledgements

This Handbook is the first document released in September 2020 by the QBAI®Fundamental TeamWork. It was formally approved by the QBAI®Quality Board and by the shareholders' meeting on (24 September 2020).
QBAI®'s dedicated Team Work was composed by: R. Murabito, E. Landi, F. Speciale, P. Ventura, E. Eritreo. The supervision of the development of this Handbook has been assigned to R. Fonso (Quality Board of QBAI®).
The team would like to thank all the people involved in the project and who contributed to the release of this Handbook.
The team would like to thank in particular Dr. R. Murabito for the concept and contents reported in this Handbook, the result of years of experience in training professionals of Business Analysis.