

**POLITECNICO DI TORINO**

Collegio di Ingegneria Gestionale

**Corso di Laurea Magistrale  
in Ingegneria Gestionale**

Tesi di Laurea Magistrale

**Valutazione della fase di test in un progetto  
di sviluppo Software**



**Relatore**

Prof. Marco Torchiano

**Candidato**

Edoardo Russo

Luglio 2019



*Non ci sono errori, ma  
opportunità per conoscere le cose.  
(Ugo Foscolo)*

# Indice

Introduzione .....	7
Capitolo 1 - Introduzione ai Sistemi Informativi .....	9
1.1 - Prime definizioni.....	9
1.2 – Componenti di un Sistema Informativo .....	10
1.2.1 – Hardware.....	10
1.2.2 - Software .....	11
1.2.3 – L’importanza dei dati .....	11
1.2.4 – Comunicazione.....	13
1.2.5 - Risorse umane.....	13
1.2.6 – Processi .....	14
1.3 – L’importanza di un Sistema Informativo .....	15
1.3.1 – Tipologia di decisioni.....	16
1.3.2 – Le cinque fasi del processo decisionale .....	16
1.4 – Le diverse tipologie dei sistemi informativi.....	18
1.4.1 – Transaction Processing System .....	19
1.4.2 – Sistemi per la Business Intelligence.....	19
Capitolo 2 - I sistemi informativi nella PA .....	21
2.1 - Amministrazione e Pubblica Amministrazione .....	21
2.2 - Struttura organizzativa della PA .....	23
2.2.1- Implementazione delle politiche pubbliche in EU .....	23
2.2.2- Complessità organizzativa della PA .....	24
2.3 - Struttura generale dei processi nella Pubblica Amministrazione.....	25
2.4 - E-Government.....	26
2.4.1 – Soluzioni di E-Government.....	26
2.4.2 - Livelli di interazioni.....	27
2.4.3 - Struttura informatica dei modelli di E-Government .....	29
Capitolo 3 - Progetto di sviluppo Software .....	32
3.1 – Software .....	32
3.1.1 – Storia del Software.....	32
3.2 – Progetto: Definizione, vincoli e classificazione .....	34
3.2.1- Progetto - Definizione.....	34
3.2.2- Tipologie di progetto.....	35
3.2.3 – Vincoli di Progetto .....	36
3.3 – Progetto di sviluppo Software .....	37

3.3.1 – Persone e i processi di un progetto di sviluppo software .....	38
3.3.2 – Persone .....	38
3.3.3- Processi di sviluppo Software .....	41
3.4 – Modelli di sviluppo Software .....	42
3.4.1 – Modello a Cascata .....	43
3.4.2 – Modello Incrementale .....	45
3.4.3 – Modello a V .....	46
3.4.4 – Modello a spirale .....	47
3.4.5 - Modello Agile .....	49
Capitolo 4 - Stadi di lavorazione dello sviluppo software .....	51
4.1- Studio di Fattibilità .....	51
4.2 – Acquisizione, analisi e specifica dei requisiti .....	52
4.2.1 – Definizione e classificazione di requisito .....	53
4.2.2- Specifica dei requisiti .....	54
4.3-Definizione e Progettazione del Sistema .....	55
4.4 – Codifica del software .....	56
4.5 – Test del Software .....	56
4.5.1- Classificazione degli Errori .....	57
4.5.2 – Gli Obiettivi della fase di Test .....	57
4.5.3 – I limiti della fase di test .....	58
4.5.4- Classificazione dei Test .....	59
4.6 – Rilascio e Manutenzione del Software .....	61
Capitolo 5 – Misurazione delle prestazioni .....	63
5.1 – La teoria della Misurazione .....	63
5.1.1 – Elementi fondamentali della teoria della misurazione .....	64
5.1.2 - Scale di Misura .....	65
5.1.3 – Classificazione delle scale di misura .....	66
5.2 – Gli indicatori .....	68
5.2.1- Definizione di indicatore .....	68
5.2.2 – Classificazione degli indicatori .....	70
5.3 – Indicatori chiave di Prestazioni .....	71
5.3.1- Proprietà fondamentali .....	72
5.3.2 – Classificazione dei KPI .....	72
5.3.3- Costruzione dei KPI .....	73
Capitolo 6 – Caso studio: Pagamento Unico per l’Agricoltura .....	75
6.1-Il Contesto del progetto .....	75
6.1.1- La Politica Agricola Comune .....	75

6.1.2 – Pagamento Unico per l’Agricoltura (PUA).....	82
6.2- L’iter di una Richiesta Unica di Finanziamento .....	86
6.2.1 – Presentazione della RUF .....	87
6.2.2 – Istruttoria della Pratica .....	88
6.2.3 – Calcolo del premio e generazione delle liste.....	90
6.3 – L’istruttoria di pratiche con premi pascolo .....	93
6.3.1 – Come recuperare le informazioni da BDN.....	95
6.3.2 – L’algoritmo del Calcolo Carico UBA .....	97
6.3.3 – I controlli Pascoli .....	102
6.3.4- Adeguamenti del Software effettuati per l’anno 2018 .....	103
Capitolo 7 - Costruzione di indicatori di Performance per il monitoraggio e controllo della fase di Testing.....	106
7.1 – Fase di Test del progetto PUA .....	106
7.2 – Software di tracciamento delle fasi di sviluppo .....	108
7.3 – Definizione degli Indicatore di Performance per la fase di Testing.....	111
7.4 – Misurazione e Valutazione dei degli indicatori.....	115
7.5 – Piano di intervento .....	117
Conclusioni .....	119

# Introduzione

I sistemi informativi sono un set di strumenti integrati per raccogliere, archiviare e processare dati e per fornire informazioni utili nel momento giusto alle persone che ne hanno bisogno. Molte imprese ed organizzazioni si affidano ai SI per eseguire e gestire le proprie attività principali, per interagire con i propri fornitori e clienti per gestire i propri bilanci, per coordinare le risorse umane e per competere nel mercato di riferimento.

In generale i Sistemi Informativi sono un insieme di strumenti che sono utili per l'esecuzione delle attività principali di un'azienda. (britannica, 2019)

L'obiettivo della tesi è quello di illustrare parte del lavoro svolto dall'autore all'interno dell'azienda AizoOn, società di consulenza tecnologica di innovazione, che opera a livello globale. Si parte esaminando argomenti molto generali, come i Sistemi informativi, per arrivare ad argomenti molto specifici, come le fasi del ciclo di vita di sviluppo software. Dopo una prima parte introduttiva teorica, l'autore presenterà un caso studio. In particolare l'elaborato analizzerà le prestazioni della fase di testing per lo sviluppo di nuovi requisiti per un applicativo software impiegato dalla pubblica amministrazione (PA) utile per la gestione del fondo FEAGA (Fondo europeo agricolo di garanzia) in ambito di Richiesta di Finanziamento Unico (RFU).

La tesi verrà organizzata nel seguente modo:

- Nel Primo capitolo sono analizzati i Sistemi informativi, ovvero quali componenti lo compongono, perché sono così importanti al giorno d'oggi e quali sono le principali categorie;
- Nel secondo capitolo, dopo aver presentato alcune definizioni ed analizzato in maniera sintetica la struttura organizzativa della PA, viene evidenziato come i SI supportano la PA;
- Nel terzo capitolo viene analizzata la componente Software dei sistemi informativi ed i progetti di sviluppo Software;

- Nel quarto capitolo sono dettagliate singolarmente ogni fase del progetto di sviluppo di un software;
- Nel quinto capitolo, dopo aver presentato la Teoria della Misurazione, sono presentati gli Indicatori;
- Nel sesto capitolo viene presentato il caso di studio: il contesto, gli stakeholder, i processi principali ed il processo seguito ed modificato dall'autore.
- Nel settimo capitolo sono definiti e misurati indicatori specifici per valutare la fase di test condotta su evolutive del software presentato nel sesto capitolo.



# Capitolo 1 - Introduzione ai Sistemi Informativi

In questo capitolo l'autore analizzerà sinteticamente le varie componenti di un sistema informativo (SI), illustrerà i motivi principali per cui sono così importanti e quali sono le tipologie più diffuse.

## 1.1 - Prime definizioni

Nella società moderna le informazioni giocano un ruolo cruciale non solo per le attività e i processi aziendali, ma anche e soprattutto nella vita giornaliera di tutte le persone. Le informazioni sono gestite e fornite attraverso particolari strumenti e metodologie chiamati appunto SI. Di seguito sono riportate tre definizioni:

- *“Il sistema informativo è lo studio di una rete complementare, costituita da componenti software e hardware, che le persone ed organizzazioni utilizzano per raccogliere, filtrare processare, creare e distribuire dati.”* (Wikipedia, l'enciclopedia libera, s.d.)
- *“I sistemi informativi sono una combinazione di hardware, software e strumenti di comunicazione che le persone costruiscono ed utilizzano per generare e fornire dati utili, organizzate in una forma predefinita.”* (Schneider, 2010)
- *“I sistemi informativi sono componenti interconnessi che lavorano insieme per raccogliere, processare, immagazzinare e distribuire informazioni per supportare i processi di decisione, di coordinamento, di controllo, di analisi all'interno di un'organizzazione”* (Kenneth C. Laudon, 2012).

È facile notare come tutte e tre le definizioni sottolineano la differenza tra le componenti che compongono un sistema informativo ed il ruolo che queste componenti giocano all'interno di un'organizzazione. È possibile affermare che i SI sono costituiti da componenti differenti, che lavorano insieme per

valorizzare maggiormente una precisa organizzazione. È molto importante precisare e non confondere i sistemi informativi con i sistemi informatici. Nel paragrafo successivo sono analizzate nel dettaglio le componenti di un sistema informativo e sarà evidenziata meglio la differenza tra sistema informativo e sistema informatico.

## **1.2 – Componenti di un Sistema Informativo**

Un SI è formato da diverse componenti che lavorano contemporaneamente per realizzare una gestione migliore delle informazioni con lo scopo di aumentare il valore dell'organizzazione.

In un sistema informativo possiamo distinguere sei componenti fondamentali:

1. Hardware;
2. Data;
3. Software;
4. Communication, in italiano Comunicazione;
5. Risorse umane;
6. Processi.

Le prime tre componenti compongono il sistema informatico. Nei paragrafi successivi sono analizzate singolarmente tutte le componenti con lo scopo di fornire una migliore comprensione di come queste componenti si integrino al meglio.

### **1.2.1 – Hardware**

L'hardware rappresenta la parte fisica del sistema informativo. A seconda della grandezza e della tipologia del sistema informativo, la componente Hardware può essere composta da più o meno elementi. In generale esso comprende:

1. Computer di diverse funzionalità e potenza di calcolo;
2. Dispositivi di input (mouse, tastiera, ecc.);
3. Dispositivi di output (Stampanti, Monitor, altoparlanti, ecc.);

4. Dispositivi di Storage (Hard-Disk, CD, ecc);
5. Dispositivi di Telecomunicazione (Internet, Reti Virtuali Private, ecc);

## 1.2.2 - Software

Il Software è un insieme di informazioni e di istruzioni che dicono alla componente hardware che cosa fare. Il software è la parte intangibile di un sistema informatico. Esistono diverse categorie di software, ma possiamo individuare due tipologie di categorie che comprendono tutte le altre:

- I sistemi operativi software permettono l'utilizzo delle componenti hardware, come ad esempio Android per gli smartphone e Microsoft Windows per i personal computer;
- Gli *applicativi software*, invece, sono programmi progettati per aiutare gli utilizzatori della componente Hardware ad eseguire determinate attività. Alcuni esempi di Applicativi Software sono 'Microsoft Word' e 'Power Point' rispettivamente utilizzati per la stesura di un documento e per la creazione di una presentazione.

In generale l'utente interagisce con le applicazioni software attraverso il desktop di un Computer o lo schermo di un dispositivo portatile. L'applicazione software comunica con il Sistema Operativo il quale a sua volta comunica con l'hardware. Le frecce indicano i versi del flusso informativo. Nel capitolo successivo si analizzeranno in maniera più dettagliata gli applicativi software e si introdurrà una nuova scienza denominata 'Ingegneria del Software'.

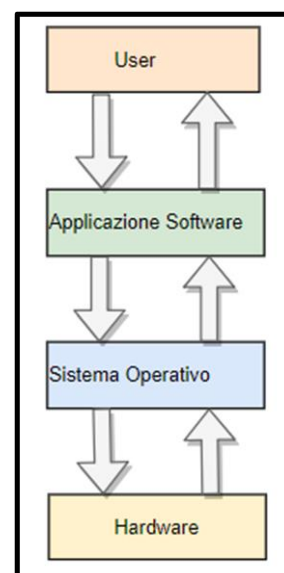


Figura 1- Interazione tra Utente, Software e Hardware

## 1.2.3 – L'importanza dei dati

La terza componente dei sistemi informativo è rappresentata dai dati. Essi, come la componente software, sono intangibili. I dati di per se non aggiungono

nessun particolare valore alle organizzazioni, in quanto di per se sono privi di significato. Un dato può essere rappresentato da un numero, un codice, una data, una parola. Un'informazione, invece, è l'interpretazione di un'aggregazione di dati che assume un significato ben preciso in un determinato contesto. Come si legge da una ricerca effettuata da Khurana and Mandke nel 2009, il valore di un'informazione dipende dalla sua integrità che deriva dall'accuratezza, dalla consistenza e dall'affidabilità dell'informazione. L'integrità di un'informazione gioca un ruolo fondamentale per determinare un vantaggio competitivo duraturo nel tempo. (Reema Khurana, 2009)

Si legge invece da una seconda ricerca, effettuata da Stair e Reynolds, che un'informazione per essere preziosa deve essere completa e l'azienda non deve impiegare grandi risorse per ottenerla. Suggestiscono inoltre che il valore dell'informazione può essere calcolato come il profitto che ha permesso all'azienda di ottenere meno il costo per produrre l'informazione. (Stair Ralph, 2006)

Una volta che l'azienda ha prodotto un grande numero di informazioni, si può affermare che essa possiede tutti i requisiti necessari per prendere le giuste decisioni. Il consumo di informazioni produce la Knowledge (Conoscenza) che può essere utilizzato per prendere decisioni, stabilire nuove policy aziendali e persino incoraggiare l'innovazione. Di seguito viene riportata la definizione di Conoscenza presentata da Gamble and Blackwell (2001):

*" La conoscenza è un fluido misto di esperienza raccolta, di abilità, di informazione contestuale e di visioni di esperti, che costituisce un modello per valutare ed incorporare nuove esperienze e nuove informazioni. Essa trae origine e risiede nella mente degli individui, soggetti del sapere. Nelle organizzazioni la conoscenza è racchiusa non solo nei documenti e negli archivi, ma anche nelle routine, nei processi, nelle pratiche e nelle norme". (Paul R. Gamble)*

Da questa definizione possiamo affermare che la conoscenza è:

1. Un agglomerato di elementi differenti;

2. La conoscenza può essere strutturata o non strutturata
3. Può essere generata sia come processo che come documento.

La gestione ottimale delle informazioni e lo sfruttamento della Knowledge permette all'organizzazione di raggiungere una posizione rilevante nel proprio business.

### **1.2.4 – Comunicazione**

Oltre alle componenti appena descritte, che per molto tempo sono state considerate come le più importanti di un sistema informativo, negli ultimi decenni si è aggiunta una nuova componente: Communication, che tradotto letteralmente significa 'Comunicazione'. Tutti i SI possono esistere e funzionare anche senza l'abilità di comunicare con altri SI. Si pensi infatti che i primi computer erano semplici macchine isolate ognuna dalle altre e che venivano impiegati per eseguire una ed una sola istruzione. Tuttavia nel corso degli anni si sono sviluppate tecnologie avanzate, tutt'oggi ancora in evoluzione, che rendono la connessione di un dispositivo ad altri estremamente facile, veloce e poco costosa. In una realtà così evoluta è molto difficile che esistano ancora SI che lavorano in maniera isolata da altri e la comunicazione è diventata così importante che è diventata una componente fondamentale dei SI.

### **1.2.5 - Risorse umane**

Le persone sono coloro che utilizzano un sistema informativo e che seguono procedure ben precise cosicché la conoscenza(Knowledge) presente nei database e nelle stesse persone può essere utilizzata per comprendere che cosa è accaduto in passato e capire qual è la scelta migliore per il futuro.

Possiamo distinguere due categorie principali di persone:

- Specialisti dei Sistemi Informativi, ovvero persone esperte di SI che si occupano di progettare, programmare e mettere in funzione un sistema

informativo. Essi possono essere analisti, programmatori, progettisti ed altre risorse. Questa categoria di persona costruisce un sistema informativo secondo le specifiche e le risorse che gli vengono assegnate dal cliente.

- Utente Finale, sono quella categoria di persona che utilizzano un sistema informativo o le informazioni che esso produce. Essi possono essere ad esempio contabili, ingegneri, addetti alle vendite ed altre ancora.

### **1.2.6 – Processi**

I processi sono le regole, le istruzioni o meglio ancora i passi che bisogna seguire per raggiungere un obiettivo specifico. I SI sono diventati sempre più integrati con i processi, con l'effetto di aumentare la produttività ed ottenere un migliore controllo dei processi stessi. È solito dire, infatti, che “I processi stanno alle Persone come i Software stanno agli Hardware”.

In ogni organizzazione è importante sapere riconoscere quali sono i processi principali, per distinguere quali sono i processi core che aggiungono valore al prodotto finale e quali sono solo di supporto. In quest'ottica possiamo distinguere tre tipologie di processi differenti:

- Processi di gestione, non aggiungono valore al consumatore finale. Questi processi sono più orientati al monitoraggio ed al controllo delle attività core;
- Processi di supporto, anche essi non aggiungono valore al prodotto/servizio finali, ma sono essenziali affinché i processi primari funzionino correttamente;
- Processi primari, sono da considerare i più importanti perché sono quelle forniscono valore ai consumatori finali.

### **1.3 – L'importanza di un Sistema Informativo**

Nel mondo attuale le imprese sono sempre minacciate da una serie di fattori (politici, ambientali, sociali o economici) imprevedibili che possono minare le varie attività di business di un'impresa, ed in alcuni circostanze causare la scomparsa di interi mercati. Le imprese sono sempre in competizione tra di loro e sono sempre alla ricerca della migliore strategia da implementare per ottenere un vantaggio competitivo.

Per queste ragioni in quasi tutte le imprese sono implementati i SI aziendali che sono utili nella raccolta, nella conservazione e nella trasformazione di dati insignificanti provenienti da diverse fonti, sia interne che esterne, in informazioni di valore per migliorare in ogni livello dell'organizzazione, le strategie di breve e di lungo periodo, il processo decisionale ed i risultati.

I principale vantaggi di un sistema informativo sono elencati di seguito:

- Interpretare una grande quantità di dati rapidamente ed efficacemente;
- Consente l'accesso ad una vasta gamma di informazioni Worldwide;
- Immagazzina un grande ammontare di informazioni occupando poco spazio;
- Fornisce una veloce e accurata collaborazione e comunicazione all'interno e all'esterno dell'aziende.
- Aiuta l'azienda a svolgere le proprie attività di business.

In conclusione i SI, oltre ad assistere l'organizzazione in tutte le sue attività quotidiane, sono indispensabili per gestire al meglio le informazioni che sono raccolte sia dall'ambiente esterno che interno e renderle disponibili ai manager per prendere correttamente e più rapidamente decisioni. (Dave Bourgeois, 2019)

### **1.3.1 – Tipologia di decisioni**

Una decisione è definita come una scelta tra diverse opzioni alternative effettuata da una persona o da un software per raggiungere un determinato risultato in una determinata situazione.

Si distinguono:

1. **Decisioni Strutturate:** Sono decisioni che riguardano un'attività che è stata eseguita più volte dall'azienda e per cui è stata definita una procedura standardizzata per gestirla.
2. **Decisioni Non Strutturate:** sono decisioni che riguardano un'attività che non è stata mai gestita in precedenza. Per tale motivo non esiste una procedura standardizzata per capire quale sia la decisione migliore da intraprendere. Sarà compito del Manager, anche grazie al supporto del Sistema informativo, a decidere come procedere nella gestione dell'attività. Rientrano in questa tipologia di decisione le strategie di medio lungo periodo.
3. **Decisioni Semi-Strutturate:** sono decisioni che riguardano un problema che non è nuovo, ma per il quale esiste non esiste una procedura standardizzata utile nella gestione dell'intera attività, ma è in grado di eseguirne solo una parte di essa. Rientrano in questa tipologia di decisioni la definizione di strategie breve periodo.

In generale possiamo affermare che il maggior numero di decisioni strutturate sono prese nei livelli di management inferiori, mentre man mano che si sale di livello le decisioni diventano sempre meno strutturate.

### **1.3.2 – Le cinque fasi del processo decisionale**

Secondo il premio Nobel Herbert A. Simon (1960) il processo di decisione da parte dei manager segue alcuni passi standard, ovvero il processo decisionale attraversa tre fasi principali, in particolare:

1. Fase di Intelligence;



2. Fase di Progettazione;
3. Fase di Scelta; (Simon, 1960)

Nel 1980 il professore universitario George P. Huber aggiunge due nuovi stadi al modello presentato da Simon, in particolare:

4. Fase di Implementazione;
5. Fase di Monitoraggio (Huber, 1980)

Tale processo è eseguito da tutti i manager quando si ritrovano a prendere delle decisioni riguardo la strategia da adottare per raggiungere uno o più obiettivi o per risolvere un determinato problema.

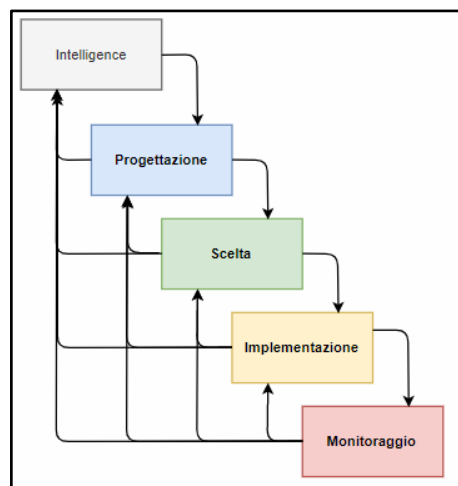
Durante la prima fase, il manager ha il compito di risalire e studiare sia l'ambiente interno che l'ambiente esterno in modo tale da individuare tutti quei fattori che possono minare l'efficacia della strategia da implementare. Dopo aver esaminato approfonditamente tutte le variabili, il management decide quali sono gli obiettivi e i risultati che bisogna raggiungere con l'implementazione di una determinata strategia. Solo se tutti gli obiettivi sono stati raggiunti è possibile affermare che la strategia è stata efficace.

Durante la fase di progettazione, il manager si occupa di sviluppare una serie di strategie alternative, valutandone alcuni fattori come il costo, la fattibilità della soluzione e l'efficacia della strategia in termini di raggiungimento degli obiettivi.

Durante la fase di scelta, il manager sulla base degli obiettivi e del budget prefissati seleziona un'alternativa.

Nella Fase di implementazione il manager implementa la soluzione scelta. Durante l'ultima fase il manager monitora che la soluzione implementata produca gli effetti che sono stati previsti.

Nella *figura 2* posta a lato sono mostrate in modo sequenziale ciascuna fase. Dall'immagine si può facilmente osservare che il processo non è unidirezionale ma bidirezionale. Infatti qualora il manager comprende che una fase sia stata eseguita in modo errato oppure intuisce che la strategia implementata non produca i risultati desiderati, ovvero sono stati falliti gli obiettivi



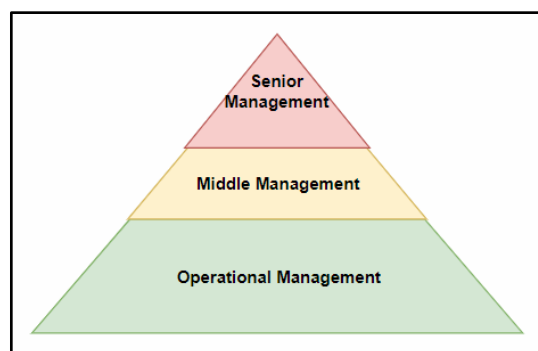
*Figura 2 – Le 5 fasi del processo decisionale*

e/o non sono raggiunti i risultati sperati, può decidere di rieseguire una fase già conclusa e far ripartire il flusso da quel punto.

Gli strumenti attraverso il quale il management fissa gli obiettivi e valuta i risultati prodotti da una determinata strategia sono gli indicatori che saranno analizzati in maniera approfondita nel capitolo 5.

## **1.4 – Le diverse tipologie dei sistemi informativi**

Lo scopo principale dei SI è quello di collezionare e processare dati in modo tale da trasformarli in informazioni preziose e renderle disponibili alle persone giuste per migliorare il processo di definizione delle strategie e il processo decisionale. Pertanto le informazioni messe a disposizione devono essere consistenti con il livello del Management a cui sono presentate. In particolare, ogni azienda ha una propria struttura che è composta da un numero diverso di livelli Management. Nei livelli più alti troviamo il senior Management che si occupa di definire la strategia di lungo periodo e di assicurare le performance



*Figura 3 – Livelli di Management*

finanziarie dell'azienda. Ad un livello intermedio troviamo invece il Middle Management che si occupa di eseguire i programmi ed i piani prefissati dal

senior Management. Mentre nei livelli più bassi il Management operativo si occupa di eseguire le attività quotidiane e ripetitive del business aziendale.

Un'azienda tipica ha differenti SI che supportano le attività di ciascun livello appena descritto. In particolare possiamo dividere la famiglia dei SI in due grandi categorie:

- Sistemi di elaborazione delle transazioni (TPS, inglese per " Transaction processing System ")
- Sistemi di Business Intelligence (BI).

### **1.4.1 – Transaction Processing System**

I sistemi TPS elaborano transazioni predefinite, producono output predefiniti e conservano nei database le informazioni ritenute necessarie. I sistemi TPS, inoltre, forniscono ai manager dei livelli operativi gli strumenti necessari per tracciare e storicizzare tutte le attività elementari e le transazioni dell'azienda. Lo scopo principale dei sistemi a questo livello dell'organizzazione è quello di rispondere a domande di routine quotidiane per cui le informazioni necessarie sono facilmente reperibili, aggiornate ed affidabili, ad esempio “Qual è il livello delle scorte in magazzino di quel prodotto?” oppure “Quanti prodotti sono stati venduti oggi?”.

I manager attraverso questi sistemi monitorano lo status delle operazioni interne e le relazioni dell'azienda con l'ambiente esterno. Inoltre forniscono una grossa mole di informazioni che sono processati da altri sistemi come quelli per la Business Intelligence.

### **1.4.2 – Sistemi per la Business Intelligence**

Con il termine Business intelligence si indica l'utilizzo di strumenti software che si concentrano sulla raccolta, sull'immagazzinamento, sull'elaborazione e sulla distribuzione di informazioni per aiutare i manager durante il processo

decisionale e la definizione di strategie corrette. (STANISLAUS DEPARTMENT OF COMPUTER SCIENCE, 2018)

All'interno di questa categoria di SI è possibile distinguere tre sottocategorie principali:

1. Management Information Systems (MIS, Sistemi di Gestione dell'Informazione);
2. Decision-Support System (DSS, Sistemi di Supporto alle decisioni);
3. Executive Support System (Sistema di supporto all'esecutivo).

I primi due sono sistemi che sono di supporto principalmente per il middle Management, mentre l'ultimo è invece di supporto per il Senior Management.

I sistemi di gestione dell'informazione (MIS) forniscono ai middle manager statistiche sulle performance correnti dell'organizzazione. Tali informazioni sono utilizzati per monitorare e controllare i processi di business e predire le performance future. MIS sintetizza e documenta informazioni sulla base di informazioni fornite dai sistemi TPS. Sono sistemi molto rigidi e poco elastici. Infatti forniscono risposte a domande di routine che sono state specificate precedentemente e per cui è stata predefinita una procedura per rispondere.

I sistemi di supporto alle decisioni (DSS) si concentrano su problemi che sono unici e cambiano rapidamente, per i quali è stata implementata una procedura non del tutto automatica. Tali sistemi cercano di supportare il processo decisionali dei middle manager processando non solo informazioni provenienti da sistemi MIS e TPS, ma anche dall'ambiente esterno all'azienda.

Infine, i sistemi di supporto all'esecutivo (ESS) aiutano il senior management a prendere decisioni su problemi per cui non esistono procedure automatiche per arrivare alla soluzione. Tali sistemi presentano grafici, diagrammi ed dati provenienti da diverse fonti interne, come MIS e DSS, ed eventi esterni, come nuovi competitors o nuovi prodotti sostituiti. (Jane Price Laudon, 2014)

## Capitolo 2 - I sistemi informativi nella PA

In questo capitolo, dopo aver presentato e commentato alcune definizioni generali, sarà presentata la struttura della Pubblica Amministrazione. Alla fine del capitolo sarà, invece, analizzato l'impatto dei SI su di essa.

### 2.1 - Amministrazione e Pubblica Amministrazione

Si introducono alcune definizioni per capire meglio a cosa ci riferiamo quando parliamo di Amministrazione ed in particolare di Pubblica Amministrazione.

Di seguito sono elencate tre definizioni di Amministrazione:

1. *“Con il termine Amministrazione si indica una determinata azione presa per il conseguimento di un preciso obiettivo. È un ordinamento sistematico degli affari ed un uso calcolato delle risorse, con lo scopo di far accadere quelle cose che vogliamo far accadere e simultaneamente evitare sviluppi che sono contrari alle nostre intenzioni”.* (Fritz Morstein Marx, 1947)
2. *Amministrazione è l'organizzazione e l'uso di risorse umane e/o non per raggiungere un obiettivo.* (Lloyd G. Nigro, 1988)
3. *L'arte dell'amministrazione è la direzione, il coordinamento e il controllo di persone per raggiungere alcuni traguardi precedentemente fissati.* (White, 1926)

Dalle seguenti definizioni è facile capire che con il termine Amministrazione ci si riferisce a due elementi fondamentali:

1. Un impiego di risorse comune;
2. Il raggiungimento di un obiettivo comune;

È possibile concludere quindi che con il termine Amministrazione si identificano tutte le risorse, gli asset, le infrastrutture e ecc. che sono coinvolti per ottenere un obiettivo comune basato su una cooperazione reciproca e sforzi congiunti.

In base all'assetto istituzionale, è possibile distinguere la Pubblica Amministrazione dall'Amministrazione privata.

Con la prima ci si riferisce ad un'amministrazione che opera in un quadro statale, mentre con la seconda ad un'amministrazione che non opera in un quadro statale ovvero imprese private.

Con l'obiettivo di avere una migliore comprensione di cosa sia una Pubblica Amministrazione sono presentate di seguito tre definizioni e quindi evidenziati gli elementi comuni.

- *“Pubblica amministrazione è l'applicazione dettagliata e sistematica della legge. Ogni applicazione della legge è un atto dell'amministrazione.”* (Wilson, The Study of Administration, 1887)
- *“Pubblica Amministrazione consiste in tutte quelle operazioni che hanno come scopo il compimento o il rafforzamento delle politiche pubbliche.”* (White, 1926)
- *“Pubblica amministrazione:*
  - *È un gruppo cooperativo che opera in un'istituzione pubblica;*
  - *Copre i tre poteri governativi (esecutivo, legislativo, giudiziario) e le loro interdipendenze;*
  - *Ha un ruolo importante nella formulazione delle politiche pubbliche e per questo motivo anche nei processi politici*
  - *È differente dall'amministrazione privata.”* (Lloyd G. Nigro, 1988)

Arrivare ad una completa definizione di PA è difficile a causa delle innumerevoli attività che ricadono sotto il suo nome. Tuttavia dalle definizioni appena elencate si evince che la PA è collegata alla implementazione delle politiche pubbliche e delle pubbliche organizzazioni. È possibile quindi definire pubblica amministrazione come l'amministrazione di una pubblica organizzazione che ha come obiettivo comune quello di portare a termine una serie di programmi ed attività al fine di raggiungere obiettivi per la società.

## **2.2 - Struttura organizzativa della PA**

Si può dividere la pubblica amministrazione in centrale e locale. Si parla di Amministrazione centrale quando l'attività di competenza è svolta da uffici che hanno l'ubicazione nella capitale e che posseggono l'autorità su tutto il territorio nazionale. Sono riconosciute come PA centrale i ministeri e gli enti centrali dello stato.

Si parla, invece, di Amministrazione periferica o locale quando l'amministrazione è organizzata in più sedi che hanno sede locale e posseggono un'autorità territoriale limitata. Sono riconosciuti come PA locali o periferiche le regioni, le provincie e i comuni. Le dimensioni degli enti locali sono molto diverse, basti pensare che gli abitanti dei grandi comuni (Roma, Milano) sono dieci volte maggiori delle regioni minori (Molise e Valle D'Aosta) e mediamente mille volte superiori dei comuni più piccoli.

### **2.2.1- Implementazione delle politiche pubbliche in EU**

La struttura organizzativa della pubblica Amministrazione Europea è di tipo gerarchico, e ciò comporta che l'implementazione delle politiche pubbliche attraversa diversi livelli gerarchici della pubblica amministrazione, in particolare:

- Le istituzioni comunitarie definiscono le politiche comunitarie nei settori nei quali è avvenuto un passaggio di competenza dagli stati membri alla Comunità.
- Gli stati membri della comunità europea emanano leggi, promulgano regolamenti e definiscono il budget. Ogni stato deve rispettare le politiche definite dall'UE per non rischiare di subire sanzioni pecuniarie;
- Le regioni, possedendo una certa indipendenza di regolamentazione e di spesa, definiscono proprie norme seguendo i principi dettati dallo stato;
- I comuni, infine, insieme ad altre amministrazioni locali forniscono ai cittadini ed alle imprese servizi locali.

## 2.2.2- Complessità organizzativa della PA

Per complessità organizzativa si intendono quei fattori di natura diversa che rendono più difficile l'attività svolta dalle PA.

Come abbiamo più volte sottolineato la PA svolge tutte quelle funzioni, come l'erogazione di servizi o la regolamentazione dei comportamenti, che la collettività ha demandato allo Stato.

In particolare, l'attività di regolamentazione è effettuata da quasi tutti enti della PA, i quali sono strutturalmente legati e dipendenti alla regolamentazione di altri enti che si trovano ad un livello gerarchico superiore. La dipendenza non è né univoca e né fissa. Non è univoca perché l'ente potrebbe svolgere la propria attività seguendo i principi definiti da più enti; non è fissa perché l'ente non esercita l'attività di regolamentazione seguendo i principi definiti dall'ente immediatamente superiore di gerarchia (vedere la figura 4).

Invece, l'attività di erogazione di uno specifico servizio può essere svolta da un unico ente oppure può essere necessaria la collaborazione di diversi

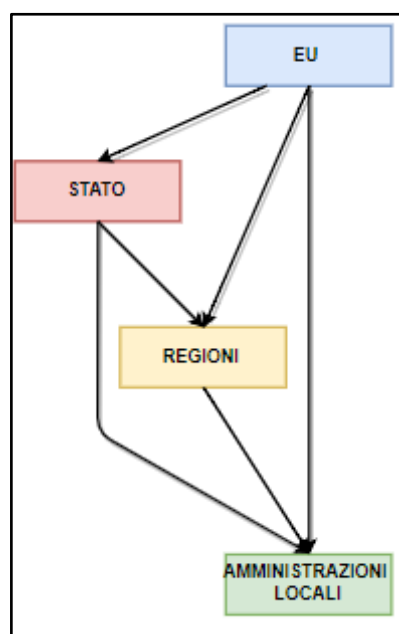


Figura 4 – Gerarchie nella PA

enti, che possono essere posti a livelli organizzativi diversi, aumentando di fatto la complessità organizzativa.

A ciò si aggiungono le differenze esistenti tra i diversi stati dell'unione Europea. Ad esempio la struttura gerarchica della PA amministrazione italiana è completamente differente da quella della Gran Bretagna.

Infine, sebbene la comunità europea fornisce delle direttive da seguire ed obiettivi comuni da raggiungere, ogni stato è autonomo nella scelta delle strategie da implementare.

Tutti questi fattori aumentano le complessità organizzative della PA e spiegano perché non è possibile costruire un sistema informativo comune per tutti i



cittadini europei e perché non si è sviluppato un mercato di applicativi software per le pubbliche amministrazioni, come invece è successo per le organizzazioni private.

## 2.3 - Struttura generale dei processi nella Pubblica Amministrazione

Nel primo capitolo abbiamo identificato i processi come una componente fondamentale dei sistemi informativi e li abbiamo classificati in sistemi di supporto, di gestione e primari. I processi, in particolare i processi primari, sono influenzati dalla funzione e dal livello gerarchico dell'ente. Nelle PA centrali i processi primari saranno principalmente di regolamentazione; mentre nelle PA locali i processi saranno rivolti ad erogare servizi al consumatore finale che può essere il cittadino, un'impresa o un altro ente pubblico.

Nella tabella seguente (*Tabella 1*) sono presentati esempi di tipologie di processi differenti.

Processo	Finalità	Esempi
Processo di gestione	Gestire l'ente e i servizi forniti	Pianificazione budget e controllo di gestione
Processo di supporto	Funzionamento interno dell'ente.	Gestione del personale, Contabilità, Acquisti
Processi primari di regolamentazione	Definire i principi e le regole di comportamento per i cittadini, imprese e PA.	Regolamentazione comunale, provinciale, ecc.
Processi primari di erogazione di servizi	Fornire il bene e/o servizio al consumatore finale.	Erogare e controllare servizi alle imprese, ai cittadini ed ad altri enti pubblici

Tabella 1 – Esempi di processi nella P0041

## **2.4 - E-Government**

L'utilizzo da parte della PA di tecnologie dell'informazione e della comunicazione è avvenuto in ritardo rispetto alle imprese ed organizzazioni private. Tuttavia soprattutto negli ultimi decenni molti governi internazionali, comprendendo le innumerevoli potenzialità, hanno incominciato a diminuire questo gap tecnologico aumentando gli investimenti nel settore ICT. La chiusura di tale divario tecnologico è favorito principalmente da una continua diminuzione dei costi delle tecnologie informatiche, in particolare le tecnologie che riguardano la componente hardware.

Il fenomeno, secondo il quale le tecnologie ICT sono utilizzate dalla PA per incrementare l'efficacia e l'efficienza e raggiungere una maggiore trasparenza e semplificazione dei servizi forniti alla collettività, è indicato con il nome di E-Government (Electronic Government, che si traduce in italiano Governo Elettronico). (Jeong, 2007)

È importante sottolineare che con il termine E-Government non si riferisce al semplice utilizzo di nuove tecnologie che hanno promosso l'informatizzazione e la digitalizzazione della PA, ma riguarda l'uso delle tecnologie ICT, combinato ad un cambiamento organizzativo e normativo, da parte della PA per migliorare i processi primari e allo stesso tempo ridurre drasticamente i costi.

### **2.4.1 – Soluzioni di E-Government**

Con il termine E-Government ci si riferisce all'utilizzo di tecnologie ICT e ad un radicale cambiamento organizzativo da parte della PA per aumentare l'efficienza e l'efficacia, per facilitare l'accesso ai servizi pubblici da parte dei cittadini e per agevolare l'accesso alle informazioni.

Abbiamo inoltre anticipato, che la PA è composta da un numero elevato di enti pubblici e gestisce differenti processi primari che posso coinvolgere uno o più

enti. Di seguito proponiamo una classificazione dei sistemi di E-Government che si differenziano in base alla tipologia di attore che interagisce con la PA, in particolare:

- Government to Citizen (G2C, che si traduce Governo e Cittadini), si riferisce all'insieme di soluzioni software che supportano la relazione tra la PA e i cittadini. L'obiettivo principale è quello di fornire attraverso il Web o applicazioni mobile un maggior numero di informazioni e servizi ai cittadini.
- Government to Business (G2B, che si traduce Governo e Aziende private), si riferisce all'insieme di interazioni tra la PA e le imprese private. In questo modello di business, le soluzioni e-Procurement sono quelle più evolute e più utilizzate. Per e-Procurement ci riferiamo a tutte quelle soluzioni digitali, principalmente tramite web, attraverso cui la PA pubblica bandi e concorsi per l'acquisto di beni e servizi da imprese private o per la vendita di beni a privati.
- Government to Government (G2G, che si traduce Governo e Governo), si riferisce all'insieme di soluzioni software che hanno l'obiettivo di migliorare la comunicazione tra i vari enti pubblici, migliorando tutti quei processi che necessitano l'interazione di più enti pubblici.

Il Caso Studio che sarà presentato nel Capitolo 6 si tratta di un progetto di sviluppo software per aiutare la pubblica Amministrazione Regionale nell'espletamento dei compiti previsti dalla legge riguardo i finanziamenti alle aziende agricole. In quest'ottica possiamo inquadrare, il sistema che descriveremo nel sesto capitolo, un sistema di G2B.

## **2.4.2 - Livelli di interazioni**

Nonostante la diversità delle attività svolte dalla PA è possibile individuare un modello generale che è seguito da tutti gli enti pubblici. In particolare possiamo distinguere due canali ben distinti:

- Front-End, costituisce l'interfaccia con cui il cittadino, l'impresa privata o un ente pubblico formula una richiesta alla PA. L'interfaccia può essere un sito web, un contact-center oppure uno sportello fisico. All'interno di questo canale è possibile distinguere tre fasi ben distinte:
  1. La fase di informazioni in cui il cittadino, impresa o ente pubblico raccoglie le informazioni necessarie per sottoporre in maniera corretta la richiesta alla PA;
  2. La fase di richiesta, in cui il cittadino, impresa o ente pubblico inoltra la propria richiesta attraverso canali ufficiali precedentemente menzionati.
  3. La fase di consegna, ovvero la fase in cui l'ente che ha ricevuto la richiesta e comunica al cliente l'esito della richiesta.
- Back-end, comprende tutte quelle fasi che svolgono la PA senza il necessario bisogno di interagire con il cliente. All'interno di questo blocco è possibile distinguere due fasi ben distinte:
  1. La fase di istruttoria, ovvero la fase in cui uno o più enti della PA effettuano una serie di controlli sulla richiesta effettuata, ovvero si esamina la correttezza e la completezza dei dati inseriti nella richiesta;
  2. La fase di esecuzione della richiesta, ovvero la fase in cui uno o più enti della PA elaborano le richieste ricevute;

Il seguente schema è illustrato nella figura 5 dove si mettono in luce i due blocchi principali, le fasi che appartengono a ciascun blocco e la dipendenza logica tra ogni fase.

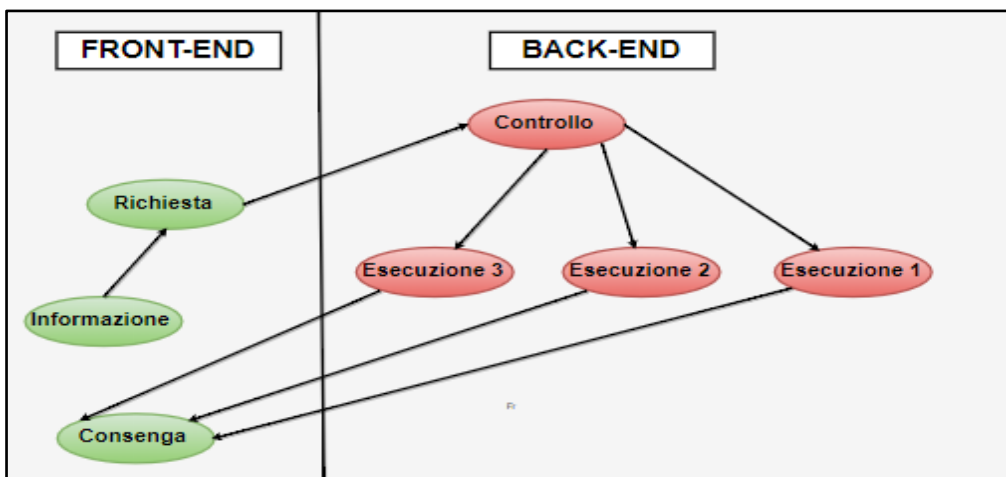


Figura 5- Schema interazionale con la PA

### 2.4.3 - Struttura informatica dei modelli di E-Government

La struttura informatica dei modelli di E-Government può essere suddivisa in tre categorie di sistemi:

1. I sistemi di accesso, ovvero i sistemi con cui la PA si relaziona con il proprio cliente;
2. I sistemi di supporto, ovvero i sistemi che supportano la PA nello svolgimento delle proprie attività;
3. I sistemi di cooperazione, ovvero i sistemi che permettono lo scambio di informazioni e di servizi tra i diversi enti della PA;

#### 2.4.3.1- Sistemi di accesso

Le modalità con cui un Ente pubblico si interfaccia con il cliente sono realizzate attraverso il front-end di un modello di E-Government. Una proprietà fondamentale dei sistemi di accesso è la possibilità di interazione tra la PA e il cliente attraverso canali differenti (sito internet, applicazione mobile, sportello fisico). A seconda della tipologia di attività svolta e del livello di digitalizzazione dell'ente pubblico, le fasi dell'attività possono proseguire sullo stesso canale o intraprendere canali differenti. Inoltre è possibile che la stessa tipologia di attività può essere eseguita con canali differenti. Sarà il cliente a scegliere quale canale prediligere.

### **2.4.3.2 - Sistemi di supporto**

I sistemi di supporto, come ci suggerisce la parola, sono un insieme di applicazioni informatiche riservate al supporto dei processi aziendali. È possibile pertanto distinguere all'interno di questo insieme di applicazioni informatiche due categorie ben distinte:

- Applicativi software orizzontali, forniscono aiuto ai processi di gestione e di supporto;
- Applicativi software verticali, supportano i processi primari.

I software orizzontali sono applicativi creati per una specifica utilità (per questo sono detti anche “software specifici”) che sono adattabili a soddisfare esigenze comuni ad imprese appartenenti a settori differenti. I software verticali, invece, sono tutti gli applicativi creati per una determinata specificità, ma che non sono di utilità a tutto il mercato. (emotori, 2019)

### **2.4.3.3 - Sistemi di cooperazione**

Nei paragrafi precedenti abbiamo più volte evidenziato che la PA può eseguire i propri processi primari autonomamente, ovvero senza avere il bisogno di interagire o comunicare con altri uffici della PA. Il più delle volte è necessaria, invece, una forte collaborazione che è possibile solo se sono stati implementati adeguati sistemi di cooperazione.

Le componenti principali per creare sistemi di cooperazione che possano favorire l'interoperabilità dei SI sono:

- Un canale di comunicazione che possa assicurare sia un'elevata sicurezza che una notevole velocità. Internet, grazie alla sua rapida diffusione e grazie soprattutto ai numerosi strumenti di comunicazione che mette a disposizione, è diventato lo strumento principale con il quale le PA comunicano tra di loro e si scambiano. Un esempio sono le reti privati virtuali meglio conosciute con l'acronimo VPN.

- Una serie di servizi standard di interoperabilità utili per un rapido ed efficace scambio di informazioni e documenti. La concretizzazione di questi servizi è concepibile sia attraverso applicativi software verticali che orizzontali. Esempi di questo genere sono la posta elettronica e dischi di reti condivisi.
- Servizi specifici di cooperazione assicurano un saldo collegamento fra i diversi applicativi e i diversi enti che sono chiamati a collaborare tra di loro. (G. Bracchi, 2010)

## Capitolo 3 - Progetto di sviluppo Software

Al giorno d'oggi la forma di organizzazione più diffusa per lo sviluppo di Software è rappresentata dal progetto. In questo capitolo analizzeremo alcune definizioni della componente Software, presenteremo le caratteristiche principali di un progetto ed infine ci concentreremo sui progetti di sviluppo software presentandone diversi modelli.

### 3.1 – Software

Nel primo capitolo abbiamo definito il software come quella parte intangibile dei SI in cui è definito il comportamento della componente Hardware. Per comprendere a fondo di cosa sia un software, sono proposte due definizioni:

1. *“Un software è un insieme di istruzioni e/o programmi che istruiscono il computer nel compiere un'attività specifica.”* (Software, 2019)
2. *Computer software, o semplicemente software, è un generico termine che si riferisce ad un insieme di dati e/o istruzioni che dicono al computer come funzionare.*

È facile notare come le parole chiavi di entrambe le definizioni sono: un insieme di istruzioni e un computer. Si può concludere quindi che il software è quella componente fondamentale di un sistema informatico che definisce il modo di lavorare della componente hardware.

#### 3.1.1 – Storia del Software

Nei primi anni di sviluppo dell'informatica, il software viene utilizzato unicamente per risolvere problemi di calcolo matematico da parte di alcuni scienziati ed ingegneri. Il problema da risolvere era ben noto allo scienziato che si preoccupava semplicemente di sviluppare un software che fosse in grado di risolvere la sua esigenza.



A partire dalla metà degli anni 60' il mondo del software visse una grande e profonda crisi. Durante la conferenza sull'ingegneria del software tenutasi nel 1968 a Garmisch in Germania, lo scienziato Edsger Dijkstra affermò che la causa della crisi fosse da ricercare nella differenza di potenza e complessità che esisteva tra la componente Hardware e la componente software. Egli sosteneva che la programmazione di software si fosse poco evoluta nel tempo perché gli scienziati si sono sempre occupati a potenziare la parte hardware. Tuttavia quando la componente Hardware progredì e raggiunse elevate performance si decise di utilizzare i computer anche in contesti diversi, come il controllo dei processi industriali e la gestione delle imprese, ma la componente software si rilevò fortemente inadeguata.

La crisi si manifestò in diversi modi:

- Il budget iniziale e la data di consegna sono raramente rispettati;
- Software sono poco efficienti;
- Software non rispettano i requisiti;

La necessità di una migliore gestione del processo di sviluppo del software ha dato origine alla disciplina dell'ingegneria del software.

Per comprendere a fondo la nuova disciplina sono riportate tre definizioni:

1. È l'applicazione sistematica di scientifiche e tecnologiche conoscenze, metodologie ed esperienze per la progettazione, per l'implementazione e per la documentazione di un software.
2. È l'applicazione di un approccio sistematico, organizzato e quantitativo per lo sviluppo, per il funzionamento e per la manutenzione del software.
3. È una disciplina ingegneristica che si occupa di tutti gli aspetti e le fasi per la produzione di un software.

Dalle tre definizioni è facile constatare che con la nuova disciplina ci riferiamo all'utilizzo delle scienze e metodologie ingegneristiche per supportare tutte le fasi del ciclo di vita di un software.

Ogni produzione di software consta di diverse fasi fondamentali: analisi dei requisiti, progettazione, implementazione, test, installazione. Non eseguire o

concludere correttamente una delle fasi può creare grandi problemi nello sviluppo di un software.

Per lo sviluppo di software di piccole dimensioni, tutte le fasi sono eseguite da una singola persona. Invece per lo sviluppo software di medio/grandi dimensione è necessaria la collaborazione di diverse persone per portare a termine correttamente ogni fase. Al giorno d'oggi la forma di organizzazione più utilizzata per pianificare nel tempo le diverse attività e per coordinare il team di lavoro è il progetto.

Prima di discutere nel dettaglio le varie fasi di sviluppo software è importante approfondire il termine 'Progetto'. Nei prossimi paragrafi sarà analizzato in dettaglio il progetto di sviluppo software, mentre le diverse fasi di sviluppo software saranno oggetto del capitolo 4.

## **3.2 – Progetto: Definizione, vincoli e classificazione**

In questo paragrafo è approfondito il termine Progetto di cui viene fornita una definizione e sono chiarite le parole chiavi. In seguito sono presentate diverse tipologie di progetto che si differenziano in base ad alcune caratteristiche. Infine sono presentate diversi modelli di progetti di sviluppo software.

### **3.2.1- Progetto - Definizione**

Con la parola progetto l'Istituto di Project Management (PMI) indica uno *sforzo temporaneo* per creare un *risultato*.

In primo luogo, l'esecuzione di un progetto comporta uno sforzo ben preciso, ovvero sono impiegate un certo numero di risorse umane, finanziarie e tecnologiche.

In secondo luogo, il progetto è limitato nel tempo. Un progetto ha sempre una data di inizio ed una data di fine attività. È possibile che la data di fine attività

non sia stata definita fin dall'inizio, ma può essere stabilita anche in corso d'opera.

Infine il progetto ha il compito di produrre un risultato. Tale risultato, l'obiettivo del progetto, può essere sia un prodotto o servizio.

### **3.2.2- Tipologie di progetto**

Ogni progetto è diverso da un altro e può essere classificato sulla base di differenti caratteristiche. In base alle proprie caratteristiche i progetti sono eseguiti secondo metodologie diverse. Ogni metodologia è adatta per l'esecuzione, nella maggior parte dei casi, di uno ed un solo progetto perché è basata su specifiche considerazioni tecniche, organizzative, di progetto e di squadra.

La classificazione dei progetti, che viene presentata di seguito, si basa un numero diverso di fattori come la complessità, le fonti di capitali, gli attori coinvolti e gli obiettivi da raggiungere.

Secondo la complessità un progetto può essere classificato come:

- *Progetto di bassa complessità* si ha quando le relazioni fra le diverse attività sono basilari; non è richiesta una dettagliata programmazione o una grande organizzazione.
- *Progetto di alta complessità* si ha quando le attività sono numerose e sulle attività sono presenti molti vincoli di interdipendenza.

Secondo le fonti di capitali un progetto può essere classificato come:

- *Progetto pubblico* è finanziato da enti pubblici;
- *Progetto privato* è finanziato da enti privati;
- *Progetto misto* è finanziato sia da enti pubblici che da enti privati;

Secondo il contenuto del progetto è possibile distinguere:

- *Progetto di costruzione* riguardano i progetti che hanno a che fare con la costruzione fisica di infrastrutture;

- *Progetto di Information Technology (IT)* riguardano i progetti che hanno a che fare lo sviluppo e la manutenzione di software e/o SI;
- *Progetto per la produzione di un servizio o prodotto* riguardano quei progetti che hanno l'obiettivo di produrre un servizio o prodotto innovativo. Sono tipici del dipartimento di Ricerca & Sviluppo;

Secondo le risorse umane che sono coinvolte è possibile distinguere:

- *Progetto di dipartimento* riguarda i progetti che coinvolgono solo persone che appartengono allo stesso dipartimento;
- *Progetto interno* riguarda i progetti che coinvolgono tutta l'azienda per l'esecuzione del progetto;
- *Progetto matriarcale* riguarda i progetti che coinvolgono uno o più dipartimenti di un'azienda;
- *Progetto esterno* riguardano i progetti che coinvolgono team di lavoro appartenenti ad aziende esterne.

Un progetto può ricadere in una sola o più classificazione. Ad esempio un progetto per la costruzione di un complesso residenziale commissionato da una società di investimenti ad una società esperta nel settore delle costruzioni è un progetto *privato, esterno, di costruzione e di alta complessità*.

### 3.2.3 – Vincoli di Progetto

In ogni progetto, qualsiasi sia la sua natura, esistono delle tensioni tra chi investe nel progetto e chi ha il compito di portare a termine il progetto. Queste tensioni si riferiscono principalmente a tre domande:

- 1 In quanto tempo deve essere completato il progetto?
- 2 Quanto costa l'esecuzione del progetto?
- 3 Qual è la qualità minima che deve essere assicurata al progetto?

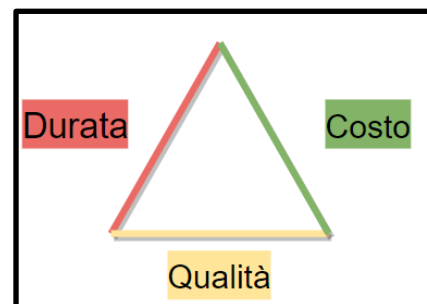


Figura 6 – Vincoli di un Progetto

Si afferma che sia possibile soddisfare solo due vincoli su tre e che non sia possibile rispettarli tutti contemporaneamente. Un progetto che si prefigge l'obiettivo di essere completato in un breve periodo e ottenere un'alta qualità deve necessariamente avere un costo elevato. Questa situazione è facilmente comprensibile, se si guarda il triangolo della *figura 6*. Un progetto, a seconda di come il capo progetto decida di soddisfare i differenti vincoli, può essere posizionato in qualsiasi punto all'interno del triangolo. Il capo progetto decide di posizionarsi in un punto o in un altro a seconda dei requisiti e delle risorse messi a disposizione dagli investitori. Se i lati del triangolo rappresentano la miglior soddisfazione di un vincolo, un progetto potrà al massimo posizionarsi in uno dei tre vertici soddisfacendo al meglio delle possibilità solo due vincoli.

### **3.3 – Progetto di sviluppo Software**

Come anticipato nel paragrafo precedente con la parola progetto si riferisce ad un impegno comune per raggiungere un obiettivo ben preciso. Nel caso di progetto di sviluppo Software l'obiettivo da perseguire è quello di sviluppare un nuovo Software.

Le motivazioni per la produzione di un software sono numerose e differenti, tuttavia le motivazioni più comuni sono:

1. Soddisfare esigenze specifiche di uno specifico cliente (Software personalizzato e venduto ad un solo cliente);
2. Soddisfare l'esigenza percepita di alcuni gruppi potenziali di utenti (Software standardizzato e commercializzato a più clienti);
3. Soddisfare un'esigenza personale, ovvero un programmatore scrive un piccolo software per svolgere una determinata azione e/o per risolvere un determinato problema;

Le prime due motivazioni riguardano in genere di sviluppi software di medio/grandi dimensioni che richiedono uno o più team di progetto.

### **3.3.1 – Persone e i processi di un progetto di sviluppo software**

Ogni progetto ed in particolare i progetti di sviluppo software dipendono da tre componenti fondamentali: le persone o risorse umane, i processi e gli strumenti o le tecnologie del settore. Ogni componente fornisce un valore differente al progetto. La risorsa più importante è senza dubbio la componente umana poiché le persone sono creative, forniscono una visione del futuro che non si è ancora realizzata ed applicano il loro intelletto per trasformare quella visione in realtà. Tuttavia le persone sono anche la causa primaria dei problemi che si verificano sui progetti. Le persone commettono errori che si ripercuotono sulla durata del progetto e sul suo costo. Per sfruttare al meglio le potenzialità che le persone offrono e ridurre al minimo il numero di errori durante lo svolgimento di un progetto è necessario introdurre i processi.

Nel paragrafo 1.2.6 abbiamo definito i processi come delle procedure che bisogna seguire per raggiungere un obiettivo ben preciso. I processi, infatti, aiutano le persone a svolgere diverse attività senza avere la necessità di pensare che cosa fare ogni volta che si vuole ottenere un risultato. In altre parole i processi liberano la mente di concentrarsi su quelle attività che hanno bisogno di una maggiore attenzione fornendo una serie di passi da seguire per quelle attività che si svolgono regolarmente.

Dopo aver fornito un'introduzione generale delle persone e dei processi e sottolineato la loro importanza nei progetti, nei paragrafi successivi ci concentreremo ad analizzare le diverse risorse umane e i processi principali che sono impiegati all'interno di un progetto di sviluppo software.

### **3.3.2 – Persone**

A seconda della tipologia di software che si vuole sviluppare è necessario un team di lavoro diverso. Ogni persona ha un differente bagaglio di competenze e di esperienze. Pertanto è facile comprendere che non tutte le risorse sono

adeguate per ogni tipologia di progetto. In generale si può affermare che all'aumentare della complessità o dell'importanza del progetto è necessario impiegare sia un maggior numero di risorse sia quelle che hanno acquisito una maggiore esperienza.

Per organizzare al meglio il lavoro delle persone e per non creare ambiguità sui compiti è necessario una chiara e completa definizione dei ruoli e delle responsabilità, ovvero chi deve fare cosa, quando e quanto tempo ci deve impiegare.

Con il termine 'ruolo' si riferisce ad una posizione ben definita che ha un set definito di caratteristiche utili per l'attuazione di determinati compiti. Il numero di ruoli impiegati all'interno di un progetto, come il numero di persone impiegate, dipende dalla sua dimensione e dalla sua complessità. È importante chiarire che i ruoli sono separati dagli individui, in quanto questi ultimi possono svolgere più ruoli contemporaneamente e lo stesso ruolo può essere svolto da più persone. Nella seguente tabella sono elencate i differenti ruoli che sono impiegati a seconda delle differenti dimensioni del progetto.

<b>Dimensione del progetto</b>	<b>Ruoli</b>
Molto piccole	Non è necessaria una suddivisione dei ruoli.
Piccole	Sono presenti un programmatore, analista e tester in una sola persona
Normali	Sono presenti un Capoprogetto, un'analista, un progettista, un programmatore, un tester e un garante della qualità
Grandi	Sono presenti più persone per ogni ruolo presente in un progetto di dimensione normale

*Tabella 1 – I differenti ruoli che sono presenti in progetti di diverse dimensioni*

Di seguito sono analizzati i ruoli che solitamente sono presenti in un progetto di dimensioni normali. In particolare:

- Il **Capoprogetto** o **Project Manager** è la persona responsabile di conoscere tutti gli stakeholder del progetto ed essere in grado di saper comunicare efficacemente con ognuno di essi. È responsabile della gestione del budget del progetto e pianificare, monitorare e controllare tutte le attività di un progetto.
- L'**Analista funzionale** si occupa di tradurre i requisiti di business del cliente in soluzioni tecniche. È fondamentale che durante la raccolta dei requisiti dal cliente sia coinvolto anche l'analista funzionale per ascoltare il suo punto di vista e affinché possa, in base alla sua esperienza, porre domande mirate per evitare ambiguità.
- Il **Progettista**, sulla base degli obiettivi da raggiungere definisce la struttura dati, le funzionalità ed i vincoli da rispettare. Si occupa, in altre parole di tradurre in un linguaggio tecnico, quelle che sono le esigenze del cliente.
- Il **Programmatore**, oltre a fornire una stima sui tempi e sul costo per lo sviluppo, è responsabile per la costruzione del prodotto finale. È fondamentale che il progettista comunichi correttamente i requisiti tecnici al programmatore per ridurre i rischi ed aumentare le percentuali di successo del progetto.
- Il **Tester** assicura che la soluzione costruita dai programmatori rispetta tutti i requisiti presentati forniti dal cliente e quindi assicura che il prodotto finale sia privo di malfunzionamenti, errori e difetti. È compito del tester, dopo aver analizzato e valutato i requisiti tecnici fornitogli dall'analista funzionale, individuare e/o costruire dei casi di test per verificare il corretto funzionamento del software.
- Il **Garante della Qualità** si occupa di verificare che il prodotto finale fornito funzioni correttamente anche per gli utenti finali. Si tratta di una fase diversa dalla fase dei test sui requisiti funzionali perché durante questa fase, chiamata anche User Acceptance Testing (UAT, Test di accettazione da



parte dell'utente), viene assicurata la correttezza del prodotto per gli utenti finali. Questi test, generalmente, sono effettuati dal proprietario o cliente che ha commissionato lo sviluppo software e rappresentano l'ultimo stadio prima dell'accettazione da parte del cliente del nuovo software.

### 3.3.3- Processi di sviluppo Software

Come anticipato nel paragrafo 3.3 i progetti di sviluppo software sono fondamentali per *organizzare le attività* per eseguire tutti gli stadi di lavorazione fondamentali per lo sviluppo di un *software*.

Nel comma precedente sono stati evidenziate due concetti chiave:

1. Organizzare le attività, ovvero definire chi deve fare cosa e quali sono le condizioni di inizio e di fine attività.
2. Software, rappresenta l'output del processo;

Le fasi o stadi di lavorazioni principali che sono comuni alla maggior parte dei progetti di sviluppo software sono descritti nella *tabella 2*.

<b>Nome della Fase</b>	<b>Descrizione della Fase</b>
<i>Analisi</i>	Fase in cui si raccolgono i requisiti del software.
<i>Progettazione</i>	Fase in cui si definiscono la struttura del software e la logica delle varie funzionalità.
<i>Implementazione</i>	Fase in cui il sistema incluse tutte le sue funzionalità sono tradotte in un linguaggio di programmazione.
<i>Test</i>	Fase in cui sono testate tutte le funzionalità del software per verificare che siano rispettati tutti i requisiti
<i>Rilascio</i>	Fase in cui il software viene consegnato al cliente.
<i>Manutenzione</i>	Fase finale del progetto in cui il fornitore del software si preoccupa di fornire prestazioni di manutenzione per

	rimediare a tutti i malfunzionamenti che non si sono evidenziati durante le fasi precedenti.
--	--

Tabella 3 – Stadi di lavorazione in un progetto di Sviluppo Software

Gli stadi di lavorazione sono dettagliati singolarmente nel capitolo 4.

Nel corso di un progetto di sviluppo software è possibile svolgere le singole fasi una sola volta o più volte. Ogni progetto, in base alla dimensione, alla complessità, alle risorse impiegate e al settore in cui si troverà ad operare, è unico. Perciò l'organizzazione delle varie fasi di lavorazione, modello di sviluppo del software, può rivelarsi una mossa vincente per una tipologia di progetto ma può risultare una fallimentare per altre tipologie. È doveroso chiarire che non esiste un modello di sviluppo ideale ma sarà il team incaricato del progetto, ed in primo luogo il capo progetto, a decidere come organizzare e pianificare le varie fasi ed attività.

Ad esempio un'azienda che si trova ad operare su sistemi critici utilizzerà dei processi ben strutturati per ridurre al minimo l'errore umano; mentre con un software gestionale, con frequenti cambi di requisiti, un processo meno informale e rigido sarà più efficiente.

### 3.4 – Modelli di sviluppo Software

Come anticipato nel paragrafo precedente i modelli di sviluppo software sono una rappresentazione astratta del processo che può essere usato per spiegare i differenti approcci allo sviluppo di un software. Ogni modello può essere modificato e contestualizzato in modo da aumentarne l'efficacia.

I modelli di sviluppo software possono essere suddivisi in due grandi famiglie:

- Modelli predittivi, ovvero modelli in cui si conosce alla perfezione quale deve essere il software da consegnare e si predicono e si pianificano tutte le attività per raggiungere quel tipo di prodotto;
- Modelli adattivi, ovvero modelli in cui è possibile cambiare gli obiettivi del progetto, se necessario, durante lo sviluppo;

In generale sono da preferire i modelli adattivi a quelli predittivi, ma ci sono i casi in cui i modelli predittivi funzionano piuttosto bene, ovvero progetti di piccole dimensioni dove la probabilità di un cambio di requisiti è molto bassa. Nei paragrafi successivi saranno analizzati sia modelli adattivi che predittivi.

### 3.4.1 – Modello a Cascata

Il primo modello di sviluppo software che andiamo ad analizzare è il modello a cascata, maggiormente conosciuto con il nome in inglese Waterfall methods. Il nome deriva dal fatto che gli stadi di lavorazione sono eseguiti sequenzialmente. In particolare lo stadio successivo non può iniziare se non è stato completato del tutto lo stadio precedente.

Il modello a cascata è un modello adattivo e come anticipato nei paragrafi precedenti si tratta di un approccio allo sviluppo molto rigido e dovrebbe essere utilizzato solo quando i requisiti sono chiari e difficilmente possono cambiare durante l'esecuzione delle fasi successive alla raccolta dei requisiti. Un altro limite di questo

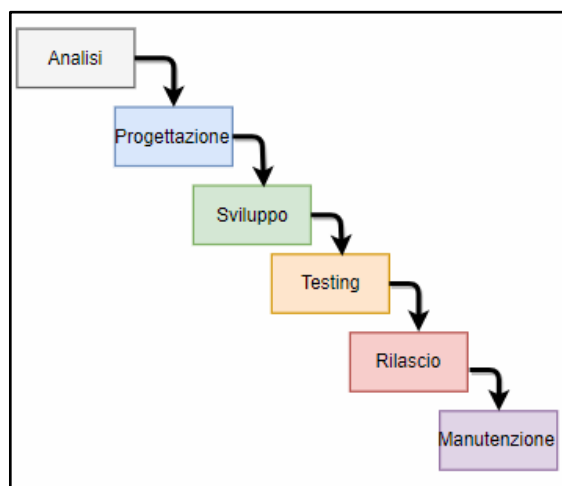


Figura 7 – Rappresentazione del modello a Cascata

modello deriva dal fatto che le attività successive non possono iniziare se non è stata interamente completata l'attività precedente e per tale motivo la durata del progetto sarà inevitabilmente piuttosto lunga.

Al giorno d'oggi sono pochi i progetti che utilizzano questo approccio, piuttosto utilizzano alcune varianti che risolvono in parte alcuni limiti, in particolare il modello a cascata con feedback ed il modello a cascata con fasi sovrapposte.

Il primo approccio ti permette di muoverti non solo in avanti, ma anche di poter tornare indietro di uno stadio di lavorazione. Con questa variante, ad esempio, se il progetto è arrivato nello stadio di progettazione e si individua un errore nei requisiti si può tranquillamente tornare alla fase precedente ed effettuare le necessarie correzioni. Queste azioni non erano permesse con il tradizionale modello a

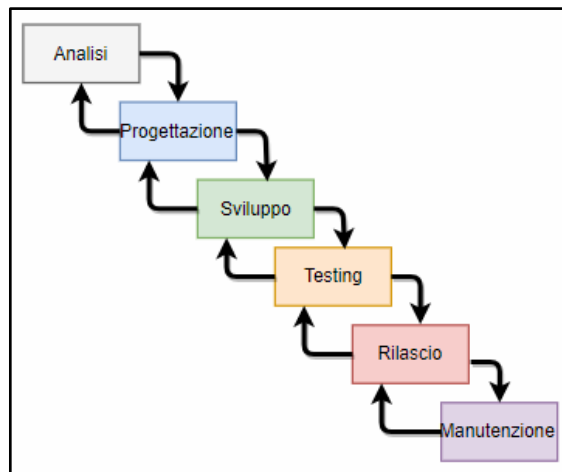


Figura 8 – Rappresentazione del modello a cascata con Feedback

cascata. Pertanto se si fosse commesso un errore negli stadi iniziali difficilmente l'output del processo di sviluppo sarebbe stato un prodotto software che non avrebbe soddisfatto tutti i requisiti.

Il secondo approccio, chiamato SASHIMI waterfall (Modello a cascata con fasi sovrapposte), è molto simile al modello tradizionale; la differenza consiste nella possibilità di poter sovrapporre in parte gli stadi di lavorazione. In questo modo non bisogna necessariamente aspettare la conclusione integrale di uno stadio di lavorazione per poter iniziare quello successivo.

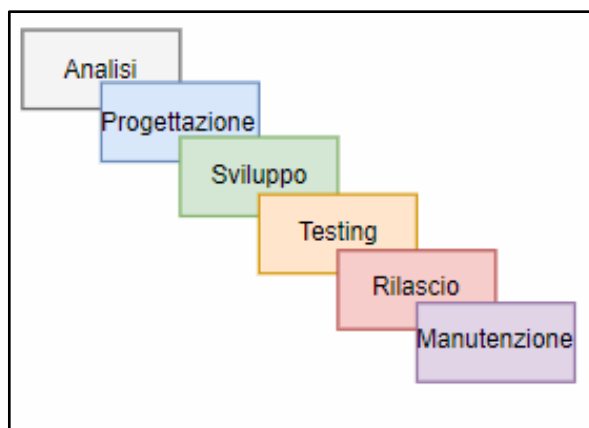


Figura 9 – Rappresentazione del modello a cascata 'Sashimi'

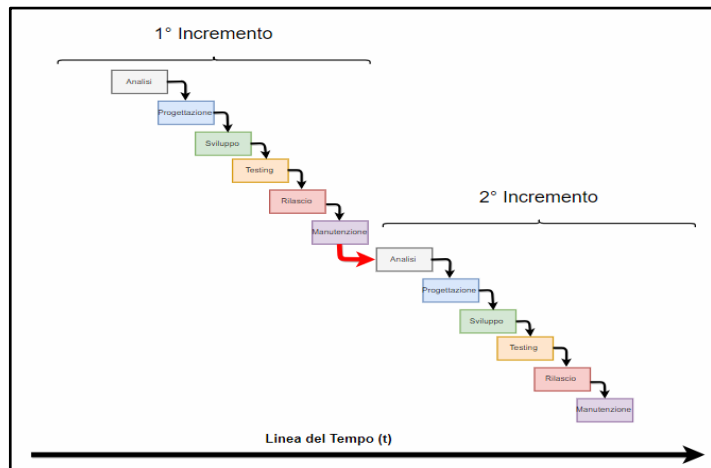
I vantaggi di questa variante sono molteplici. In primo luogo risorse umane con abilità differenti possono concentrarsi su attività diverse senza dover aspettare che altre persone finiscano i loro compiti. In secondo luogo questo approccio consente agli stadi di lavorazione successiva di modificare il lavoro svolto dagli stadi precedenti.

Tuttavia quando si utilizza un approccio del genere è consigliabile al team di lavoro di non spingersi troppo in avanti con gli stadi di lavorazione, in modo

tale da non dovere riscrivere o cancellare parte del codice a seguito di un cambio di requisiti.

### 3.4.2 – Modello Incrementale

Il modello incrementale, chiamato anche modello multi-waterfall, utilizza una serie di modelli a cascata. Ogni modello a cascata termina con la consegna al cliente di un software utilizzabile chiamato



*incremento*. Ogni incremento contiene più funzionalità di quello precedente fino ad arrivare all'incremento finale che corrisponde alla versione finale del software in cui sono rispettati tutti i requisiti forniti dal cliente. Il cliente, inoltre, fin dall'inizio definisce le priorità sulle varie componenti in modo tale che con gli incrementi iniziali sono sviluppate le funzionalità più importanti e/o urgenti.

In questo modo il cliente può valutare il sistema alla fine di ogni incremento per verificare che il software fornito rispetti i requisiti. In

*Figura 10 – Rappresentazione del modello incrementale*

caso contrario la versione attuale del software consegnata verrà sostituita con un nuovo incremento. Questo modello è un modello predittivo, iterativo ed in alcune situazioni anche adattivo. Questo approccio risulta essere molto meno rigido rispetto al modello a cascata tradizionale perché ti permette di aggiungere nuove funzionalità o modificare quelle già esistenti all'inizio di ogni incremento. Tuttavia l'elasticità di questo modello è limitata in quanto ogni nuovo incremento è sempre influenzato da quello precedente.

Questo approccio di sviluppo software è consigliabile quando le dimensioni del progetto sono molto grandi e quando la raccolta dei requisiti risulti essere complicata.

### 3.4.3 – Modello a V

Il modello V è un approccio molto simile al modello a cascata. La peculiarità di questo modello consiste nel suddividere il processo di sviluppo software in due fasi ben distinte:

- Fase di *decomposizione* composta dai seguenti stadi di lavorazione:
  - Analisi dei requisiti
  - Progettazione del sistema
  - Progettazione dei moduli;
- Fase di *integrazione* composta dai seguenti stadi di lavorazione:
  - Test dei moduli;
  - Test di sistema
  - Test di accettazione;

Con le attività su lato sinistro della V si scompone il software partendo dal suo livello concettuale più alto fino ad arrivare ad una descrizione dettagliata delle singole attività da eseguire.

Sul lato destro invece le attività considerano il software finito partendo dal livello di astrazione inferiore fino ad arrivare a quello massimo. È facile notare come

l'attività di implementazione sia stand-alone e rappresenta il punto di inversione del processo di astrazione.

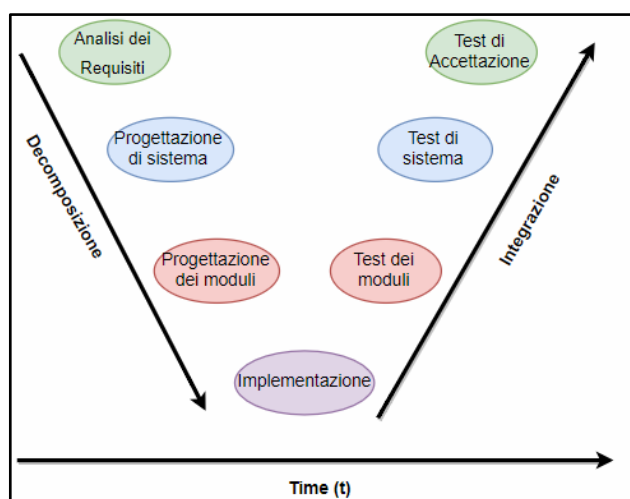


Figura 11 – Rappresentazione del modello a V

Ogni attività sul lato sinistro corrisponde ad un'attività sul lato destro con lo stesso livello di astrazione.

Al livello più alto, l'analisi dei requisiti corrisponde al test di accettazione. Al livello successivo, la progettazione di sistema corrisponde al test di sistema. All'ultimo livello, infine, la progettazione dei moduli corrisponde al test dei moduli.

La novità di questo modello è quello di mettere in evidenza la relazione esistente tra le attività di decomposizione ed i relativi test.

### 3.4.4 – Modello a spirale

Il modello a spirale è a differenza degli altri già esaminati si tratta di un modello di sviluppo adattivo ed iterativo. Iterativo perché non si arriva alla soluzione finale con un solo ciclo di sviluppo software, adattivo perché eventuali errori o cambi di requisiti possono essere facilmente gestiti. La novità principale di questo approccio è la modalità con cui vengono gestiti gli eventuali rischi del progetto. Il rischio è definito come il prodotto tra due fattori:

- Probabilità che si verifichi una determinata situazione;
- Danno conseguente a quella specifica situazione;

Bisogna chiarire inoltre che il termine rischio si riferisce sia ad eventi positivi che negativi per il progetto. Per questo motivo il capo progetto deve cercare di massimizzare la probabilità di rischi positivi. Invece per gestire un rischio negativo si possono intraprendere quattro differenti azioni:

- **Accettare il rischio**, non intraprendere nessuna contromisura;
- **Trasferire il rischio**, trasferire le conseguenze economiche su un altro soggetto;
- **Mitigare il rischio**, intraprendere azioni che hanno l'effetto di ridurre la probabilità ed il danno degli eventi negativi;
- **Evitare il rischio**, pianificare in modo differente le attività per evitare l'evento rischioso.

Il modello a spirale è composto da quattro fasi ben distinte:

- 1 ° Fase** Sono determinati gli obiettivi, le varie alternative ed i vincoli del ciclo corrente;
- 2 ° Fase** Si effettua un'analisi dei rischi per determinare quali sono i possibili rischi che possono compromettere o favorire il raggiungimento degli obiettivi del ciclo. Dopo aver determinato tutti i fattori di rischio sono implementate alcune strategie per la loro gestione.
- 3 ° Fase** In questa fase sono generati i prodotti stabiliti per questo ciclo, ovvero sono realizzate le componenti o funzionalità decise nella fase 1°.
- 4 ° Fase** Si valuta il progresso appena portato a termine e si accerta che la soluzione prodotta sia accettata dagli stakeholder più importanti. Se in questa fase emergono alcuni errori, bisogna effettuare un nuovo ciclo intorno alla spirale per risolvere i problemi. Dopo aver risolto tutti i problemi si pianifica il prossimo giro intorno alla spirale.



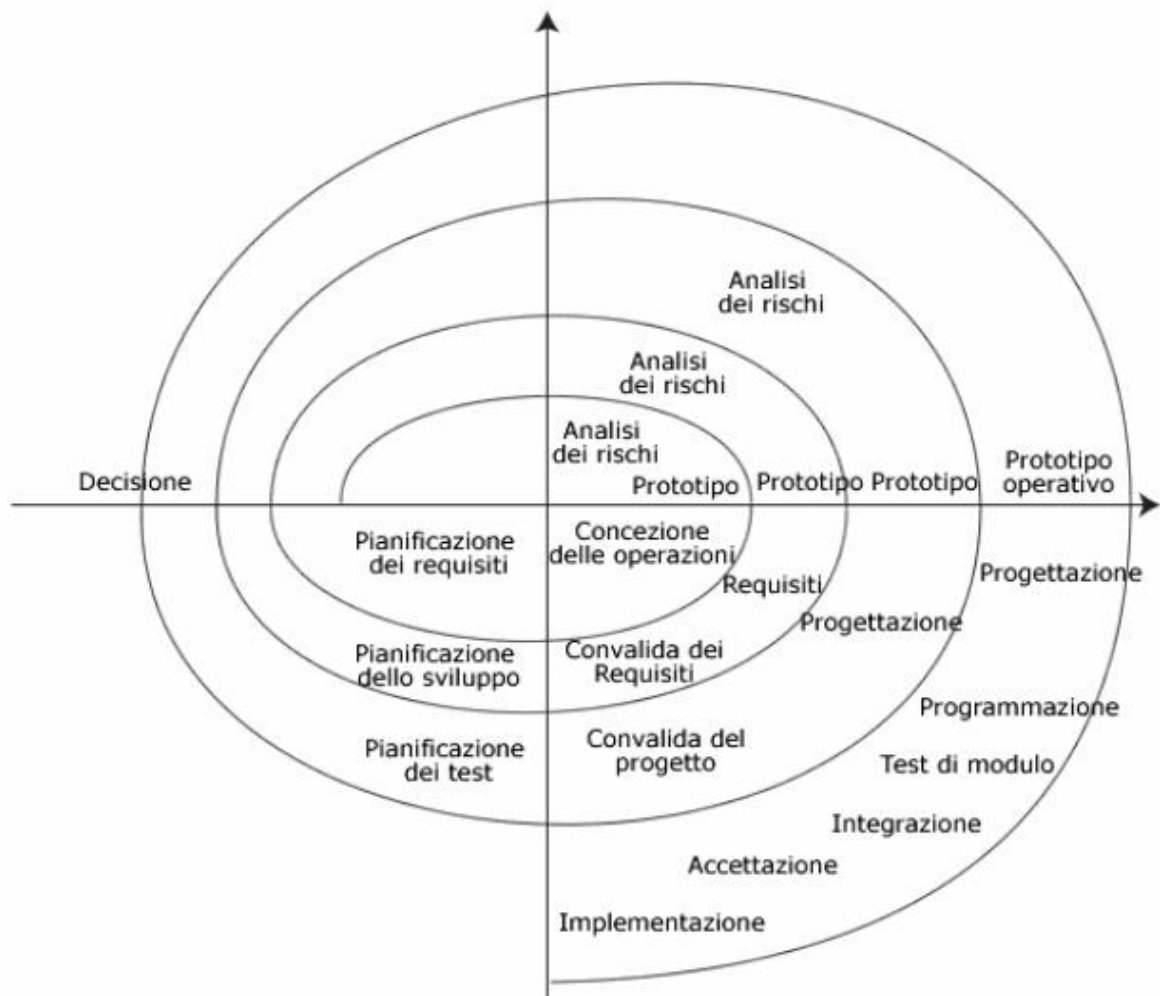


Figura 12 – Rappresentazione del modello a spirale – Immagine tratta da [https://it.wikipedia.org/wiki/Modello\\_a\\_spirale](https://it.wikipedia.org/wiki/Modello_a_spirale)

Questo modello è utilizzato principalmente quando i progetti sono di grosse dimensioni e non è possibile avere una chiara definizione di tutti i requisiti oppure la probabilità che questi cambiano durante lo sviluppo del software è molto elevata.

### 3.4.5 - Modello Agile

La metodologia di sviluppo software agile si riferisce ad un gruppo di modelli di sviluppo software basati sull'approccio incrementale, in cui gli incrementi sono piccoli ed nuove versioni del software sono create e messe a disposizione del cliente ogni 2-4 settimane.

I principi principali su cui si basa questa metodologia sono spiegati nella tabella.

<b>Principio</b>	<b>Descrizione</b>
<b>Coinvolgimento del cliente</b>	I clienti devono essere coinvolti in tutto il processo di sviluppo. Il loro compito è quello di fornire e definire le priorità di nuovi requisiti e valutare le nuove versioni del software
<b>Sviluppi incrementali</b>	Il software è costruito attraverso incrementi consecutivi ed il cliente deve specificare quali sono i requisiti da includere in ogni incremento
<b>Persone e non processi</b>	Le abilità del team di sviluppo devono essere riconosciute e sfruttate al massimo. Ogni persona del team dovrebbe essere libera di lavorare senza avere l'obbligo di seguire processi standard.
<b>Favorire il cambiamento</b>	Bisogna aspettarsi che i requisiti possono cambiare e perciò bisogna progettare il sistema che favorisca i cambiamenti.
<b>Mantenere la semplicità</b>	Bisogna evitare durante la progettazione e lo sviluppo di sistema di introdurre fattori di complessità.

*Tabella 4 – Caratteristiche principale della metodologia Agile*

Con questa metodologia il cliente viene coinvolto nel processo di sviluppo per proporre cambiamenti ai requisiti. La documentazione ridotta al minimo e le comunicazioni informali sono preferite a quelli formali.

# Capitolo 4 - Stadi di lavorazione dello sviluppo software

Come anticipato nei capitoli precedenti, in questo capitolo analizzeremo nel dettaglio ciascun stadio di lavorazione che forma il ciclo di vita di sviluppo software ed in particolare vedremo a quale ruolo spetta eseguire ogni fase e come ciascuna fase possa essere maggiormente dettagliata.

Nell'immagine seguente riproponiamo il modello a cascata in cui sono mostrate tutte le fasi del ciclo di vita di sviluppo software in maniera sequenziale.

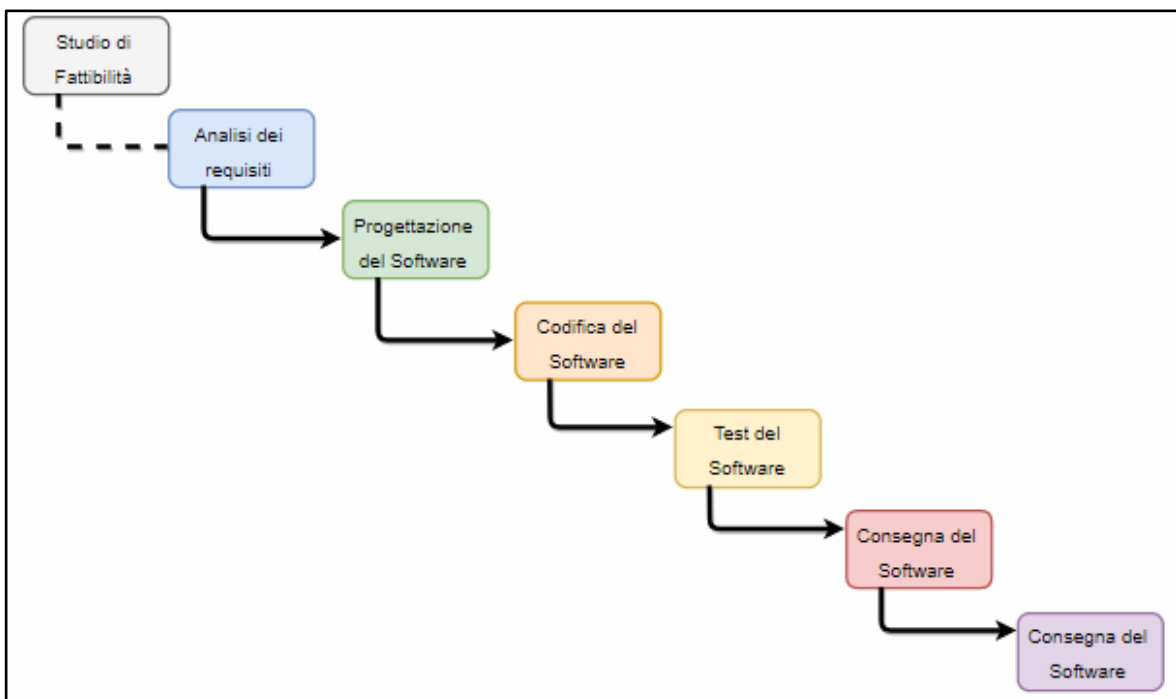


Figura 13 – Rappresentazione del modello a cascata utile ad evidenziare ogni singola fase del progetto di sviluppo software

## 4.1- Studio di Fattibilità

Lo studio di fattibilità si differenzia dagli altri stadi di lavorazione perché non si tratta di un'attività fondamentale per lo sviluppo di un software o di una sua evoluzione. Questo stadio di lavorazione viene in genere effettuato prima di incominciare il vero e proprio processo di produzione e per questo motivo è collegato con gli stadi di lavorazione successivo attraverso una linea tratteggiata.

Lo studio di fattibilità si differenzia a seconda del fatto che si tratti di uno sviluppo che mira alla creazione di un software nuovo o di uno sviluppo che riguarda una o più componenti di un software già esistente.

Nel primo caso l'input dello studio di fattibilità sono una serie di requisiti, ovvero una descrizione pratica di come il sistema si debba comportare in tutte le condizioni possibili e di come esso supporta i processi di business del cliente. Il cliente inoltre si impegna a fornire una stima del budget e della data di consegna.

Nel secondo caso l'unica differenza consiste nel fatto che lo studio di fattibilità si preoccupa di trovare miglioramenti per problemi messi in evidenza dal cliente su un software già costruito.

In entrambi l'IT manager, grazie anche al supporto di tecnici (come l'analista funzionale, sviluppatori...), si occupa di costruire, valutare e proporre diverse soluzioni software fornendo anche una valutazione sui tempi, sui costi, e sulla qualità. La qualità, ovviamente, è maggiore per quelle soluzioni che supportano al meglio il processo di business del cliente.

Tuttavia bisogna precisare che non esiste *'la soluzione ottima'*, ma la soluzione deve essere contestualizzata alle esigenze e alle priorità del cliente. Sarà infatti il cliente stesso a scegliere le soluzioni proposte.

In questa fase, inoltre, soprattutto per i progetti di grosse dimensioni è possibile definire degli obiettivi di performance che il sistema deve raggiungere. Questi obiettivi sono valutati attraverso quelli che sono chiamati i KPI, ovvero *Key Performance Indicator*. Lo studio dei KPI verrà ripreso nel capitolo successivo.

## **4.2 – Acquisizione, analisi e specifica dei requisiti**

Prima di concentrarci sullo studio del seguente stadio di lavorazione è bene chiarire un concetto fondamentale che è stato ripetuto più volte ma non è stato mai approfondito: 'Requisito'. Il concetto di requisito risulta essere indispensabile per comprendere a fondo questo stadio di lavorazione che risulta

essere uno dei più importanti. Sbagliare questa fase, che di solito risulta essere una delle prime ad essere eseguite, significa compromettere inevitabilmente quelle successive. Per tale motivo è considerata uno degli stadi di lavorazione più importanti durante lo sviluppo di un software.

#### 4.2.1 – Definizione e classificazione di requisito

Un requisito, secondo il glossario standardizzato di ingegneria del software, è definito come:

1. Una condizione o capacità di cui necessita l'utente del software per risolvere un problema o per raggiungere un obiettivo;
2. Una condizione o una capacità che deve essere soddisfatta o posseduta dal sistema o da una componente del sistema per soddisfare un contratto, uno standard o una specifica particolare;

Nell'ingegneria del software i requisiti si distinguono in requisiti funzionali ed in requisiti non funzionali.

I requisiti funzionali specificano come il sistema debba comportarsi una volta che si determina una condizione o l'utente effettui una precisa azione.

I requisiti non funzionali descrivono invece la modalità attraverso il quale il sistema si comporta e quali sono i limiti che esistono quando viene eseguita una determinata funzionalità.

Nella seguente tabella sono riassunte le principali differenze dei requisiti funzionali e non.

<b>Requisito funzionale</b>	<b>Requisito non funzionale</b>
Caratteristica del prodotto	Proprietà del prodotto
Descrive il lavoro che deve essere svolto	Descrive le caratteristiche del lavoro
Descrivono le azioni con cui il lavoro	Descrive l'esperienza dell'utente mentre utilizza una determinata funzionalità
Caratterizzato da verbi	Caratterizzato da aggettivi

Tabella 5 – Differenza tra Requisito funzionale e Requisito non Funzionale

## 4.2.2- Specifica dei requisiti

L'obiettivo principale di questo stadio di lavorazione è quello di definire tutti i requisiti, funzionali e non funzionali. Non bisogna includere alcun dettaglio riguarda la struttura o la progettazione delle varie componenti software. Si dovrebbe lasciare al progettista la piena autonomia e responsabilità di agire come meglio crede. Idealmente bisognerebbe spiegare semplicemente il comportamento esterno del sistema ed i vincoli che esistono tra le varie operazioni. Tuttavia, soprattutto nello sviluppo di software complessi, risulta quasi impossibile non includere informazioni riguardanti la progettazione del software. Le motivazioni sono diverse in particolare:

1. La definizione della struttura del sistema potrebbe aiutare a comprendere meglio alcuni requisiti;
2. L'uso di una specifica struttura potrebbe essere cruciale per la soddisfazione di un particolare requisito.

L'output di questo processo è un documento il quale deve essere redatto attraverso l'utilizzo di un linguaggio semplice e chiaro e deve essere fornito sia di diagrammi che di tabelle per rendere più efficace la comprensione del documento che risulta cruciale per la fase successiva, 'Progettazione del sistema'.

Parallelamente a questo documento, in alcuni casi, viene redatto un secondo documento: 'Il manuale utente' che spiega in maniera dettagliata come l'utente finale interagirà con il sistema per eseguire una determinata azione. Questo tipo di documento viene consegnato al committente in modo tale che può verificare che tutti i requisiti richiesti siano stati effettivamente raccolti e/o compresi.

Come anticipato più volte nei paragrafi precedenti, questa fase di lavorazione risulta essere cruciale per il successo dell'intero sviluppo software. Infatti non eseguire correttamente la raccolta dei requisiti dal cliente, inevitabilmente comporterà degli errori anche nelle fasi successive.

### **4.3-Definizione e Progettazione del Sistema**

La fase successiva alla raccolta e alla specificazione dei requisiti del software è la definizione e la progettazione dettagliata della struttura del software. In questo stadio di lavorazione i progettisti insieme all'aiuto degli analisti e sviluppatori cercano di descrivere come i requisiti sono soddisfatti attraverso il software ed effettuano un'analisi a diversi livelli di dettaglio. Il numero di livelli di dettaglio dipendono dalla complessità e dalla grandezza del software. Ai livelli più bassi di dettaglio i progettisti si preoccupano di definire le componenti principali e quali sono le macro relazioni che esistono fra di loro. A questo livello di progettazione viene definita, inoltre, la componente hardware, i sistemi operativi da adoperare, il sistema di gestione dei dati (DBMS- Database Management System) ed i linguaggi di programmazione da utilizzare nella stadio di lavorazione successiva di codifica.

Ad un livello intermedio di dettaglio i progettisti si preoccupano di scomporre ogni componente in moduli definendone i vincoli e le iterazioni tra di esse.

Al livello più alto di dettaglio i progettisti, grazie anche all'aiuto degli sviluppatori, definiscono il funzionamento di ogni modulo e quali sono le tipologie di informazioni che sono processate in ogni modulo. In altre parole i progettisti in questo livello di dettaglio prendono decisioni rilevanti riguardo la struttura tecnica di ogni modulo. Non è importante specificare tutti i dettagli necessari per sviluppare ciascun modulo, ma bisogna fornire il numero giusto di informazioni sufficienti a guidare le persone che svilupperanno i moduli. Questa fase risulta essere cruciale perché le decisioni tecniche prese dai progettisti in questa fase incidono principalmente sul soddisfacimento dei requisiti non funzionali.

## 4.4 – Codifica del software

Una volta che la progettazione del software è completata, le principali scelte tecniche sono state già prese. L'obiettivo di questa fase è quello di tradurre la struttura del sistema, definita nella fase precedente, in codice attraverso uno o più linguaggi di programmazione.

Data una precisa progettazione del software, tuttavia possono corrispondere diverse scritture di codice. Pertanto lo sviluppatore grazie alla sua esperienza deve cercare di implementare la struttura del sistema nel miglior modo possibile. Lo sviluppo incide sia sulla fase dei test che sulla fase di manutenzione. Ovviamente una specifica implementata con un codice ben scritto sarà più facile da testare e sarà meno impegnativa la fase di manutenzione. Poiché la fase dei test e di manutenzione hanno un impatto sul costo del software molto più alto della fase di sviluppo, l'obiettivo di questa fase dovrebbe essere quello di sviluppare programmi in cui si favorisce la semplicità e la chiarezza. In questo modo si favoriscono l'esecuzione delle fasi di test e di manutenzione.

È abbastanza comune da parte degli sviluppatori effettuare alcuni test prima di consegnare il software al team dei test per i test formali.

## 4.5 – Test del Software

Prima che un'applicazione viene rilasciata e messa a disposizione di chi ha commissionato il progetto, è necessario assicurarsi che l'applicazione software appena costruita lavori correttamente, ovvero che siano rispettati tutti i requisiti funzionali e non. Quando il software assume un comportamento che produce un risultato inaspettato e/o non conforme ai requisiti si dice che il software ha prodotto un errore.

Nei prossimi paragrafi è presentata:

1. Una classificazione delle diverse tipologie di errori;



2. Gli obiettivi principali della fase di Test
3. Un'analisi sui limiti della fase di Test;
4. Una classificazione dei Test, partendo da test con un livello di dettaglio maggiore per arrivare a quelli con un livello di dettaglio più basso.

#### 4.5.1- Classificazione degli Errori

Gli errori individuati durante la fase di test possono essere classificati in:

1. **Errori funzionali:** si verificano quando il software non si comporta nel modo in cui era atteso, producendo quindi un risultato inaspettato;
2. **Errori di comunicazione:** si verificano quando la comunicazione tra il software e l'utente finale viene interrotta causando problemi nell'utilizzo del Software;
3. **Errori di comando mancanti:** si verificano quando un comando previsto non viene trovato;
4. **Errori di sintassi:** si verificano quando emergono errori di sintassi grammaticale;
5. **Errori nella gestione delle eccezioni:** si verificano quando gli errori non sono gestiti correttamente attraverso l'utilizzo di messaggi esplicativi;
6. **Errori di calcolo:** sono causati da una cattiva logica, dalla mancata corrispondenza di tra le tipologie di dati, dalla formula di calcolo errata ecc;

#### 4.5.2 – Gli Obiettivi della fase di Test

L'obiettivo principale della fase di test è quello di individuare il maggior numero di errori il più presto possibile. Infatti maggiore sarà il numero di errori individuati in questa fase, minori saranno quelli individuati dai clienti durante i test di accettazione e dopo la consegna del prodotto. Oltre ad individuare il

maggior numero di errori, è indispensabile identificarli il più presto possibile, perché risolvere un errore ha un costo differente nelle differenti fasi. In particolare il costo di individuazione e risoluzione degli errori cresce esponenzialmente al progredire delle fasi di sviluppo.

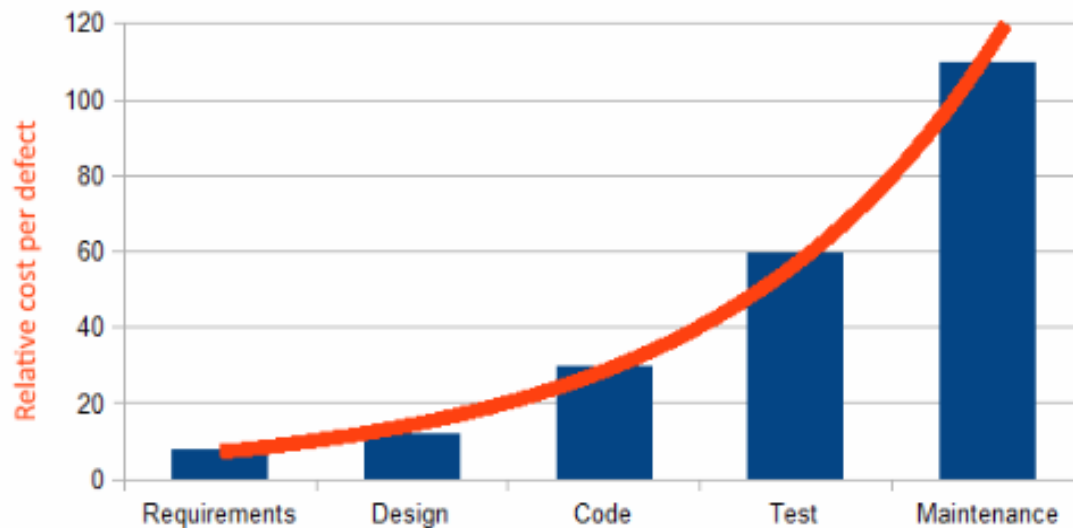


Figura 14 – Curva di Bohem- immagine tratta da '[http://thklein.com/en\\_US/cost-of-defect/](http://thklein.com/en_US/cost-of-defect/)'

Come si evince dall'immagine, se l'errore viene individuato durante la fase di raccolta e specifica dei requisiti, risulterà piuttosto basso il costo per risolverlo. È necessario semplicemente riscrivere il requisito. Tuttavia se l'errore viene individuato durante la fase dei test o nella fase di manutenzione, il costo di risoluzione è molto più elevato perché oltre a dover rieseguire interamente il processo di sviluppo (analisi, progettazione, ecc.) è necessario studiare l'impatto che la risoluzione implementata ha su altre componenti del software per evitare di introdurre nuovi errori.

### 4.5.3 – I limiti della fase di test

Sebbene la fase di test è utile per portare alla luce eventuali malfunzionamenti o errori è necessario comunque essere consapevoli dei suoi limiti. In particolare:

1. La fase di test è utile per individuare errori o difetti del software, tuttavia la non rivelazione non implica la non esistenza. La Fase di test non può individuare tutti gli errori;
2. La fase di test non fornisce alcun aiuto quando è necessario prendere decisioni del tipo ‘Rilasciare il prodotto con errori per rispettare la scadenza’ o ‘Risolvere gli errori e tardare il rilascio del prodotto compromettendo il rispetto della scadenza’;

#### **4.5.4- Classificazione dei Test**

Nel maggior parte dei casi il software prima di poter essere consegnato al cliente deve superare 4 fasi di test che si svolgono ad un livello di dettaglio di progettazione differente partendo da quello con un livello di dettaglio più alto:

- Test dei moduli;
- Test di integrazione;
- Test di sistema;
- Test di accettazione;

La fase ‘Test del software’ insieme allo stadio di lavorazione ‘Raccolta e specifica dei Requisiti’ risulta essere uno dei più importante. È fondamentale che ciascuna fase dei test sia effettuata scrupolosamente in modo tale da evidenziare qualsiasi tipo di difetto. Maggiore sarà il numero dei difetti che saranno trovati in questa fase, minori saranno i difetti che saranno evidenziati dal cliente dopo che il Software è stato consegnato.

##### **4.5.3.1 - Test dei moduli**

Durante il Test dei moduli, il software è sottoposto ad una valutazione che si concentra su specifiche unità o componenti per determinare quali sono pienamente funzionanti. Lo scopo principale è quindi quello di verificare che tutti i moduli si comportano nel modo in cui è stata definita la progettazione.

Uno dei principali benefici di questa fase è che si può testare ogni modulo del software ogni qualvolta il codice corrispondente viene modificato, consentendo di risolvere piuttosto velocemente gli eventuali problemi.

#### **4.5.3.2 - Test di integrazione**

I test di integrazione consentono ai tester di combinare i moduli come un programma e, quindi, testare gruppi di moduli e non più singolarmente ogni modulo. Questa tipologia di test è progettata per individuare eventuali difetti di connessione tra differenti moduli e/o componenti. Questa tipologia di test determina quanto efficientemente i moduli e/o componenti lavorano insieme. Bisogna chiarire che una connessione non adeguata tra i differenti moduli e/o componenti, nonostante questi singolarmente siano efficienti, impatterà in modo negativo su tutto il software.

#### **4.5.3.3 – Test di sistema**

Se nella fase precedente il software viene testato in gruppi di moduli e/o componenti, in questa fase il software viene testato nella sua interezza utilizzando casi di test reali. L'obiettivo di questo test è quello di valutare se il software ha rispettato tutti i requisiti tecnici e gli standard di qualità, ovvero si verifica che siano rispettati tutti i requisiti funzionali e non funzionali che sono stati definiti nel primo stadio di lavorazione insieme al cliente.

#### **4.5.3.4 – Test di Accettazione**

L'ultimo stadio dei test, i test di accettazione, sono condotti per determinare se il sistema è pronto per il rilascio e quindi essere messo a disposizione del cliente. Durante questo stadio finale i test saranno condotti dall'utente finale, quindi dal cliente e non più dai tester, il quale sarà incaricato di assicurarsi che tutti i requisiti siano stati soddisfatti. Completata l'ultima fase dei test, il Software può essere rilasciato al cliente.

## 4.6 – Rilascio e Manutenzione del Software

Dopo che la fase dei test è stata completata con successo inizia la fase di rilascio. Il prodotto è consegnato al cliente collocando nell'ambiente di produzione il nuovo software o le nuove componenti.

Dopo che la fase di rilascio è terminata correttamente il cliente può iniziare ad utilizzare il software per supportare i propri processi. Per un periodo di tempo concordato tra il cliente e l'it-manager, il software entra in uno stadio durante il quale l'azienda produttrice del software si impegna ad eseguire manutenzione al prodotto.

Secondo lo standard IEEE[1999] la manutenzione è quella fase del ciclo di vita di sviluppo software che ha luogo immediatamente dopo il rilascio e che ha l'obiettivo di correggere errori o adattare il prodotto per il soddisfacimento di requisiti nuovi e/o modificati.

Esistono quattro tipologie di manutenzioni:

1. Manutenzione Correttiva
2. Manutenzione adattiva;
3. Manutenzione perfettiva;
4. Manutenzione preventiva.

La manutenzione correttiva si preoccupa di correggere gli errori che sono osservati dal cliente durante l'utilizzo del software.

La manutenzione adattiva consiste nell'adattare il software per operare in un nuovo ambiente. Con il termine nuovo ambiente ci riferiamo ad una nuova condizione che influenza il modo di operare del software come un cambiamento delle politiche governative, delle regole di mercato oppure un cambiamento della componente hardware o del sistema operativo.

La manutenzione Perfettiva si occupa di implementare nuovi requisiti o modificare quelli già disponibili. La manutenzione perfettiva comporta un rafforzamento delle componenti del software già implementate con lo scopo di

migliorare le performance del software anche se queste modifiche non sono state suggerite da alcuna necessità.

La manutenzione preventiva si occupa di eseguire una serie di attività per prevenire gli errori che si presentano con una maggiore frequenza. La manutenzione preventiva comporta a ridurre la complessità del software aumentando la comprensione e la manutenibilità del Software. Le attività più frequenti di questa fase sono:

1. Aggiornare i documenti che sono interessati dalle modifiche apportate al software in modo tale che corrispondono sempre allo stato attuale del software;
2. Ottimizzare il codice per aumentare la velocità di esecuzione dei programmi oppure per migliorare la raccolta delle informazioni dei vari processi;
3. Ristrutturare il codice per ridurre la complessità nel codice ed aumentare quindi la sua comprensibilità;

Secondo uno studio condotto da Lientz e Swanson i costi della manutenzione di un progetto di sviluppo software sono suddivisi come è mostrato nella figura 15.

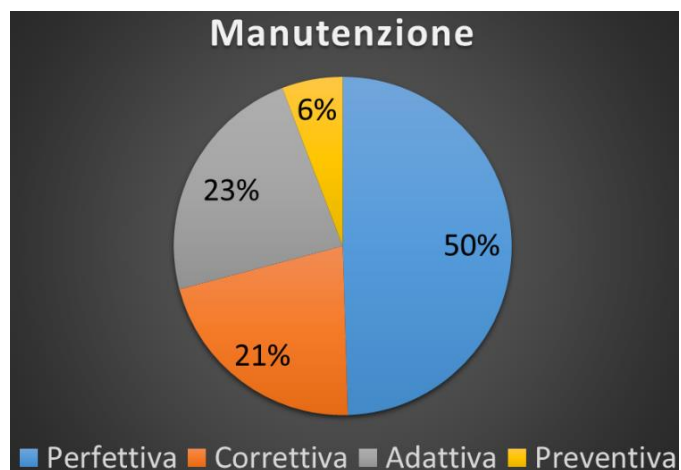


Figura 15 – Distribuzione dei costi di manutenzione

## Capitolo 5 – Misurazione delle prestazioni

La misurazione delle performance (dall'inglese Performance Measurement) è l'attività di raccogliere, esaminare e fornire informazioni utili ai diversi livelli di management riguardo le prestazioni di un gruppo di lavoro, di un processo, di una strategia e/o dell'intera azienda. Il processo di misurazione avviene confrontando i risultati attuali di ciò che si vuole misurare con degli obiettivi prestabiliti. Gli strumenti attraverso i quali i manager di ogni realtà aziendale fissano gli obiettivi e misurano le performance attuali sono gli indicatori. Questo capitolo è diviso in due parti. Nella prima parte sarà introdotta la teoria della misurazione mentre nella seconda parte analizzeremo nel dettaglio gli indicatori.

### 5.1 – La teoria della Misurazione

Ancora prima di analizzare a fondo gli indicatori, è necessario introdurre la teoria della misurazione, elemento chiave per effettuare una corretta misurazione e definizione degli indicatori.

*“La misurazione è un processo di assegnazione di numeri, misure, a proprietà di oggetti od eventi. Questa assegnazione deve essere fatta in modo che le relazioni tra i numeri associati agli oggetti corrispondano alle relazioni esistenti tra le manifestazioni della proprietà degli oggetti stessi (Finkelstein, 1982).”*

La teoria della misurazione è la teoria che tratta in modo formale il passaggio dal mondo empirico delle osservazioni a quelle delle rappresentazioni numeriche delle quantità misurabili.

### 5.1.1 – Elementi fondamentali della teoria della misurazione

La teoria della misurazione si basa sulla definizione di due sistemi e sul rispetto di una condizione, in particolare:

1. Un sistema relazionale empirico
2. Un sistema relazione numerico
3. Una condizione di rappresentazione

Un sistema empirico  $L$  è costituito da due insiemi, in particolare:

- $Q \rightarrow$  Insieme delle manifestazioni che qualificano la proprietà dell'entità empirica;
- $R \rightarrow$  Insieme delle relazioni empiriche che legano le grandezze dell'insieme tra loro;

Le relazioni possono essere di equivalenza, di transazione e di combinazione. La relazione di equivalenza ci permette di affermare che due entità empiriche sono equivalenti rispetto una proprietà. La relazione di transazione ci permette invece di ordinare le entità empiriche rispetto una proprietà. La relazione di combinazione ci permette di combinare tra loro le proprietà di entità diverse.

Un sistema numerico è costituito anche esso da due insiemi:

- $N \rightarrow$  Insieme di numeri ( $N, R, Z, \dots$ ).
- $P \rightarrow$  Insieme di relazioni che qualificano i numeri.

La misurazione lega elementi del sistema empirico ad elementi del sistema numerico. Pertanto la condizione di rappresentazione è rispettata se esiste:

- Una relazione  $M$  che proietta l'insieme  $Q$  sull'insieme  $N$ , ovvero stabilisce una corrispondenza tra le manifestazioni  $Q_i$  e i numeri  $N_i$ .
- Una funzione  $F$  che mette in relazione l'insieme  $R$  e  $P$  e tra le relazioni  $R_i$  del sistema empirico e  $P_i$  del sistema numerico.



## 5.1.2 - Scale di Misura

L'operazione empirica di misurazione di una proprietà specifica di un'entità empirica viene effettuata confrontando la proprietà dell'entità in esame con la proprietà di elementi standard che compongono la scala di misurazione. Una scala di misurazione  $S$  non è nient'altro che un insieme di elementi standard che sono caratterizzati da una precisa manifestazione della proprietà indagata  $s_i \in Q$  e che sono legati da precise relazioni. Questi elementi costituiscono un insieme standard di riferimento per la misurazione delle proprietà degli elementi dell'insieme empirico.

La mappatura  $M$  di  $Q$  su  $N$  non è univoca, ovvero esistono funzioni  $M$  diverse in grado di misurare le proprietà di entità empiriche di un sistema  $L$ . Perciò è possibile realizzare diverse scale di misurazione, ognuna associata ad una ed una sola funzione  $M$ .

Data una funzione di misurazione  $M$  ed una funzione di misurazione  $M'$  ottenuta tramite una trasformazione di  $M$  le possibili trasformazioni di scala sono rappresentate nella seguente tabella.

Nome della Trasformazione	Dettaglio della Trasformazione
Trasformazione di similitudine	$M' = K * M$ con $K > 0$
Trasformazione di potenza	$M' = K * M^n$ con $K > 0$
Trasformazione lineare	$M' = a + K * M$
Trasformazione monotona crescente	$M' = f_1(M)$ con $f_1$ una qualsiasi funzione monotona crescente.
Trasformazione di permutazione	$M' = f_2(M)$ con $f_2$ una qualsiasi funzione che determina una sostituzione ad uno ad uno di tutti gli elementi

Tabella 2 – Descrizioni delle Trasformazioni

Le tipologie di trasformazioni sono presentate in ordine di severità crescente. Una trasformazione è più severa di un'altra trasformazione se la seconda può

essere considerata un caso particolare della prima. La severità della trasformazione di potenza e trasformazione lineare sono equivalenti, perché nessuna delle due può essere considerata un caso particolare dell'altra.

Il numero di trasformazioni ammissibili decrescono con l'aumentare del numero di relazioni empiriche di una scala.

Per capire se una trasformazione è ammissibile per un certo tipo di scala, bisogna verificare se tale trasformazione muta le proprietà della scala. In altre parole la trasformazione della scala è ammissibile se le relazioni esistenti fra gli elementi che la compongono prima della trasformazione sono valide anche dopo la trasformazione.

### **5.1.3 – Classificazione delle scale di misura**

Ogni scala è caratterizzata da un certo numero di trasformazioni che ne lasciano inalterata la struttura. È possibile identificare cinque diverse scale di misura:

1. La scala nominale
2. La scala ordinale
3. La scala lineare di intervallo
4. La scala logaritmica di intervallo
5. La scala di rapporto

Nei sotto paragrafi successivi analizzeremo ciascuna scala analizzando il sistema empirico, le trasformazioni ammissibili e un esempio della scala ed una sua applicazione.

#### **Scala nominale**

Gli elementi misurati attraverso una scala nominale sono raggruppati tra loro in modo da costituire insiemi di elementi caratterizzati dalla stessa proprietà presa in esame. L'unica relazione empirica esistente tra le categorie di una scala nominale è quella di equivalenza.

- Sistema empirico:  $L \langle Q, \sim \rangle$
- Trasformazioni ammissibili: Sono ammesse tutte le trasformazioni

- Esempio di scala: (Nazionalità Italiana Si, Nazionalità italiana No)

La scala proposta nell'esempio è composta da soli due elementi ben distinti. Tale scala può essere utilizzata per contare la popolazione italiana e/o la popolazione diversa da quella italiana.

### **Scala Ordinale**

La scala ordinale è composta da una serie diverse elementi per i quali è possibile definire un ordinamento ben preciso rispetto una proprietà precisa. Con la scala ordinale è possibile indicare quali oggetti presentano un valore della caratteristica presa in esame maggiore o minore rispetto ad altri, ma non è possibile indicare di quanto un elemento sia superiore o inferiore rispetto ad un altro.

- Sistema empirico:  $L < Q, \sim, < >$
- Trasformazioni ammissibili: Sono ammesse tutte le trasformazioni al netto della trasformazione di permutazione
- Esempio di scala: scala di durezza HRB

La scala proposta nell'esempio è una scala i cui elementi non sono equi spazati tra di loro, ma fornisce solo l'informazione di ordinamento tra i vari livelli di durezza.

### **Scala lineare di intervallo**

La scala lineare di intervallo è molto simile ad una scala lineare. L'unica differenza consiste nel fatto che le distanze della proprietà misurata tra due elementi consecutivi è sempre uguale.

Le informazioni che si ottengono da una scala lineare di intervallo sono:

1. Diversità tra i diversi elementi;
  2. Ordinamento degli elementi;
  3. Distanza tra le categorie;
- Sistema empirico:  $L < Q, \sim, <, \circ >$
  - Trasformazioni ammissibili: Trasformazione di similitudine, di potenza e lineare;
  - Esempio di scala: Scala celsius.

## Scala di Rapporto

La scala di Rapporto è una scala lineare di intervallo in cui l'origine della scala non è arbitrario ma corrisponde all'assenza di manifestazione della caratteristica dell'elemento empirico.

Le informazioni che si ottengono da una scala lineare di intervallo sono:

1. Diversità tra i diversi elementi;  $A \neq B \neq C \neq D$
  2. Ordinamento degli elementi;  $A > B > C > D$
  3. Distanza tra le categorie;  $A - B = B - C = C - D$
  4. Rapporti tra categorie;  $A/B = B/C = C/D$
- Sistema empirico:  $L < Q, \sim, <, \circ_{\text{rapp}} >$
  - Trasformazioni ammissibili: Trasformazione di similitudine
  - Esempio di scala: Scala Kelvin

## 5.2 – Gli indicatori

Come anticipato all'inizio di questo capitolo gli indicatori sono gli strumenti necessari sia per fissare gli obiettivi che per valutare le performance attuali di un piano strategico, di una decisione, di processo e/o di un gruppo di lavoro. In questo paragrafo analizzeremo la definizione di indicatore, vedremo le funzionalità intrinseche di un indicatore, le diverse categorie di indicatore ed infine vedremo come costruire e definire correttamente un indicatore.

### 5.2.1- Definizione di indicatore

Di seguito sono riportate alcune definizioni di indicatore per rendere più facile la comprensione di tale strumento.

Secondo OECD (Organizzazione per la cooperazione e lo sviluppo economico) un indicatore è:

*“Un fattore quantitativo o qualitativo o una variabile che fornisce un semplice ed affidabile significato per misurare un risultato, per mettere in luce i*

*cambiamenti connessi ad un intervento o per aiutare a valutare le prestazioni di attori di sviluppo.”*

Secondo lo standard UNI11097 del 2003 un indicatore è:

*“Un’informazione quantitativa e/o qualitativa riguardo un fenomeno esaminato o un processo o un risultato, che permette di analizzare la sua evoluzione e di controllare se gli obiettivi sono stati raggiunti, influenzando le azioni e le decisioni.”*

Dalle seguenti definizioni è facile dedurre che un indicatore è una misura facile da eseguire che permette di ottenere un’informazione consistente riguardo l’attività per cui è stato costruito. È importante evidenziare la differenza esistente tra misura ed indicatore. La misura è un valore assegnato alla proprietà di un oggetto rispetto ad uno standard. Invece come anticipato, l’indicatore è una misura che ci fornisce un’informazione precisa sulle performance ed i risultati raggiunti. È possibile quindi affermare che un indicatore è una misura, mentre una misura non sempre è un indicatore.

L’indicatore, inoltre, si differenzia perché svolge tre funzioni fondamentali:

- Controllare le performance: attraverso gli indicatori i manager possono valutare e monitorare le performance delle risorse materiali e non di cui sono responsabili;
- Comunicare le prestazioni: gli indicatori svolgono la funzione di comunicare le prestazioni non solo all’ambiente interno ma anche all’ambiente esterno e quindi a tutti gli stakeholder;
- Migliorare le performance: gli indicatori permettono di evidenziare le differenze fra gli obiettivi stabiliti e le performance raggiunte e ciò permette di effettuare delle correzioni alle decisioni e/o strategia messe in atto per raggiungere prestazioni migliori.

È fondamentale, affinché gli indicatori eseguano correttamente tutte le funzioni, che esso sia definito e costruito nel modo corretto.

## 5.2.2 – Classificazione degli indicatori

In base alla tipologia di indicatore differisce l'informazione che l'indicatore trasmette. Possiamo pertanto suddividere gli indicatori in due categorie principali:

- Result indicator (RI, indicatore di risultato), misura il risultato ottenuto dalla conclusione di molte attività;
- Performance indicator (PI, indicatore di performance), misura le azioni e gli eventi che portano ai risultati;

Gli indicatori di risultato sono indicatori che trasmettono l'informazione di quanto bene si è fatto in un preciso orizzonte temporale. Tutte le attività che possono influenzare i risultati sono concluse, pertanto gli indicatori di risultato sono utili a comunicare ai stakeholder se gli obiettivi prefissati sono stati raggiunti. Gli indicatori chiave di risultato (KRI, Key Result Indicator) non sono altro che indicatori chiave che trasmettono informazioni più importanti rispetto ad altri.

Gli indicatori di prestazione invece, si concentrano sulle attività e sui processi che incidono sul risultato finale. È sulla base di questi indicatori, e dell'informazione che questi trasmettono, che i manager definiscono le strategie per raggiungere gli obiettivi prestabiliti o eseguire delle correttive a strategie già implementate. Gli indicatori chiavi di prestazione (KPI, Key Performance Indicator) sono un sottoinsieme degli indicatori di prestazione e differiscono dagli altri perché si riferiscono a fattori critici per il raggiungimento degli obiettivi (CSF, Critical Success Factors).

Nella tabella successiva sono presentate le principali differenze fra PI e RI.

<b>Indicatore di Risultato</b>	<b>Indicatore di Performance</b>
Di origine finanziaria	Di origine non finanziaria
Connesso ad eventi conclusi in passato	Connesso ad eventi ancora non terminati

Molto importanti per gli azionisti	Molto importanti per tutto il gruppo di lavoro
Difficile da ottenere	Facile da ottenere
Si misura una volta sola	Si può misurare frequentemente

*Tabella 6 –Differenze tra indicatori di Performance ed indicatori di Risultato*

Una seconda classificazione degli indicatori che in questo paragrafo viene presentata si riferisce al numero di misurazioni necessarie per la misurazione di un determinato indicatore. Sulla base di questa caratteristica distinguiamo due categorie:

- Indicatore di base, è ottenuto dalla diretta osservazione e misurazione della proprietà di una specifica entità empirica.
- Indicatore derivato, è ottenuto aggregando una o più misurazioni di proprietà differenti di una specifica entità empirica.

Ricordiamo che con il termine entità empirica ci riferiamo all'oggetto che vogliamo misurare. In generale possiamo affermare che gli indicatori chiave, sia che siano di risultato o che siano di prestazione, sono degli indicatori derivati.

### **5.3 – Indicatori chiave di Prestazioni**

Come anticipato nel paragrafo precedente, gli indicatori chiave di prestazione sono delle misure che trasmettono precise informazioni riguardo un preciso processo aziendale e sono utili ai manager per effettuare una valutazione di esso e per suggerire le decisioni da intraprendere

Nei paragrafi successivi analizzeremo quali sono le caratteristiche fondamentali dei KPI, presenteremo diverse classificazione dei KPI ed infine i passaggi per definirli correttamente.

### **5.3.1- Proprietà fondamentali**

Il vantaggio dell'utilizzo dei KPI da parte dei manager consiste nel fatto che con una sola misura essi possono capire all'istante l'andamento del processo e se necessario intraprendere delle azioni migliorative. Affinché ciò sia possibile è necessario che i KPI posseggano 5 caratteristiche fondamentali, definite anche come caratteristiche SMART:

- Specifico, l'indicatore chiave di performance è unico per il processo che bisogna misurare e non può riferirsi a processi diversi da quello per cui è stato costruito;
- Misurabile, l'operazione di misura dell'indicatore deve essere precedentemente definita, deve essere economica (scarso utilizzo di risorse umane e non umane) e praticamente fattibile (non tutte le manifestazioni empiriche sono misurabili).
- Achievable (Raggiungibile), le soglie dell'indicatore poste come obiettivi da raggiungere devono essere realisticamente raggiungibile e devono essere chiari a chi possiede il controllo del processo;
- Rilevante, è necessario che l'informazione che l'indicatore di prestazione trasmette sia utile per valutare l'andamento del processo e suggerire le azioni correttive da intraprendere.
- Timely, la misura deve essere fatta in tempi ragionevolmente breve. L'operazione di misura se dura troppo tempo rischia di fornire un'informazione diventata obsoleta e quindi inutile.

### **5.3.2 – Classificazione dei KPI**

È importante che i KPI utilizzati siano sufficienti a ricoprire ogni caratteristica del processo in esame. Diversamente il decisore sarebbe influenzato ad implementare strategie solo per migliorare alcune caratteristiche del processo e ciò porterebbe ad una gestione errata del processo.



Per questo motivo è molto importante definire nella gestione di un processo diverse categorie di KPI che ricoprono diverse caratteristiche. In particolare si possono individuare tre principali categorie di KPI:

- KPI di Efficienza;
- KPI di qualità.
- KPI di servizio;

I KPI di efficienza cercano di rispondere alla domanda quanto efficientemente un processo lavora. Un processo lavora in modo efficiente se sulla base delle risorse, materiali e non, riesce a produrre la massima quantità di output. La parola efficienza può essere declinata in diverse forme e per questo possiamo parlare rispettivamente di KPI di efficienza tecnica, KPI di efficienza umana e KPI di efficienza allocativa. Con i primi si vuole valutare l'efficienza delle tecnologie utilizzate, con i secondi si vuole valutare l'efficienza delle risorse umane, mentre con i KPI di efficienza allocativa riassume le informazioni trasmesse dai primi due.

Con i KPI di qualità invece si cerca di valutare la qualità di un certo processo. Anche in questo caso è possibile identificare diversi parametri di qualità in termini di conformità rispetto a requisiti, di affidabilità del processo e di soddisfazione del cliente.

Infine i KPI di servizio cercano di valutare il processo mentre elabora una specifica richiesta in termini di durata di elaborazione ed in termini di flessibilità.

### **5.3.3- Costruzione dei KPI**

La costruzione di KPI risulta essere un'operazione fondamentale e per nulla banale. Un KPI costruito in modo errato inevitabilmente comporta la comunicazione di un'informazione errata e/o fuorviante. Per tale motivo è molto importante che siano seguiti alcuni passi standard per eseguire in modo corretto la definizione di esso.

1. Studiare a fondo il processo per cercare di capire qualsiasi sua evoluzione nel tempo.
2. Selezionare dei KPI che possano collegare gli obiettivi prefissati. È importante selezionare più di un KPI in modo da evitare la corruzione del processo.
3. Definire il profilo del KPI, ovvero bisogna definire i seguenti campi
  - a. Nome del KPI
  - b. Categoria: indicare la classificazione del KPI
  - c. Descrizione: spiegare lo scopo del KPI e l'obiettivo a cui è collegato
  - d. Metodo: definire il processo di misurazione del KPI e di ciascuna sua voce
  - e. Formula: definire l'operazione matematica per ottenere il KPI
  - f. Interpretazione: definire i valori attesi ed i valori minimi e massimi che il KPI può assumere
  - g. Scala: indicare la classificazione della scala di misura
  - h. Unità di misura: indicare se esiste l'unità di misura
  - i. Fonte: indicare le fonti necessarie per ottenere i dati necessari per l'operazione di calcolo del KPI
4. Definire le dimensioni del KPI, ovvero definire uno o più criteri per cui sono raccolte le misure per la costruzione del KPI. Il criterio può essere un prodotto o un servizio, un orizzonte temporale, una divisione dell'azienda e ecc...
5. Effettuare un'analisi finale del KPI, in modo da valutare alcune caratteristiche del KPI appena costruiti, quali trasparenza dell'informazione trasmessa, costo della misurazione, rapidità della misurazione ed ecc.

## **Capitolo 6 – Caso studio: Pagamento Unico per l’Agricoltura**

In questo capitolo presenteremo un caso studio in cui saranno fatti collegamenti ai riferimenti teorici presentati nei capitoli precedenti. Il caso studio fa riferimento a un’esperienza personale vissuta dall’autore della tesi durante il suo tirocinio nell’azienda AIZOON specializzata nel settore informatico. In particolare costruiremo ed applicheremo degli indicatori chiavi di performance per analizzare l’evoluzione di un processo del progetto in cui l’autore è stato inserito. Il seguente capitolo sarà organizzato nel seguente modo:

1. Analisi del contesto in cui il progetto opera;
2. Analisi dei principali processi del progetto;
3. Studio di un particolare processo del progetto;

### **6.1-Il Contesto del progetto**

Il progetto PUA è un progetto di supporto alla Pubblica Amministrazione e alle aziende Agricole, per l’espletamento dei compiti previsti dalla legge in materia di finanziamenti per il settore Agricolo. Il progetto riguarda di eseguire la fase di manutenzione ad un applicativo Web utile per la gestione del fondo comunitario FEAGA (Fondo europeo agricolo di garanzia) istituiti dall’unione Europea per raggiungere gli obiettivi prefissati dalla Politica Agricola Comune.

#### **6.1.1- La Politica Agricola Comune**

La politica Agricola comune è una delle politiche comunitarie più importanti istituite dall’Unione Europea. È stata la prima politica comunitaria istituita prima dalla comunità Economica Europea (CEE) e poi dalla Unione Europea e si riferisce ad un insieme di norme e finanziamenti adottati nel settore agricole per favorire una crescita ed uno sviluppo stabile ed equo degli stati membri.

Gli obiettivi della Politica Agricola Comune, non sono cambiati dalla sua prima istituzione, ed sono descritti dall'articolo 29 del TFUE:

- Soddisfare le esigenze dei consumatori in termini di disponibilità, di prezzo, di qualità, di varietà e di sicurezza;
- Assicurare un tenore di vita dignitoso alla popolazione agricola;
- Stabilizzare i mercati.

Tuttavia è possibile individuare altri obiettivi che sono presenti in altre articoli del TFUE come:

- Raggiungere un alto livello di occupazionale (articolo 9);
- Tutelare l'ambiente (articolo 11);
- Salvaguardare il benessere degli animali (articolo 13);

La PAC cerca di indirizzare il settore agricolo verso un'agricoltura nuova, sostenibile e solidale tramite aiuti finanziari.

Il motivo principale per cui sono necessari finanziamenti per il settore agricolo è che questo risulta essere un settore fortemente influenzato da fattori economici, sanitari ed atmosferici; fattori che la popolazione agricola non è in grado di controllare. Per questo motivo si è reso necessario sostenere la popolazione agricola con aiuti economici affinché continuasse la propria attività nonostante fattori di incertezza che minassero la stabilità economica: senza un aiuto diretto la popolazione agricola europea difficilmente riuscirebbe a competere allo stesso tempo con gli agricoltori extracomunitari e continuare a soddisfare le diverse esigenze dei consumatori europei. In altre parole l'agricoltura europea non sarebbe in grado di assicurare gli standard minimi in termini di sicurezza, di qualità e di disponibilità dei prodotti.

Data la crucialità della PAC essa risulta una delle politiche comunitarie più costose istituite dall'UE sebbene il settore agricolo rappresenta solo una piccola quota dell'economia dei paesi dell'Unione Europea. Nonostante negli anni il budget destinato alla PAC è progressivamente diminuito, oggi impegna circa il 40 % del bilancio Europeo e si conferma la politica comunitaria più dispendiosa.

La politica Agricola Comune si rivolge non solo a 15 milioni di aziende agricole ed a 46 milioni di posti di lavoro, ma anche a 500 milioni di cittadini-consumatori. L'unione Europea, inoltre, sovvenzionando metodi agricoli sostenibili, si impegna a qualificare alcuni beni pubblici che non hanno un vero e proprio valore di mercato:

1. La tutela dell'ambiente;
2. La salute ed il benessere degli animali;
3. La ricchezza e la diversità del paesaggio rurale.

Al giorno d'oggi la PAC è suddivisa nei cosiddetti due pilastri. Ciascuno di essi è finanziato da un fondo europeo diverso. Il primo pilastro è finanziato dal fondo FEAGA e comprende il regime dei pagamenti diretti e le misure di mercato ed è interamente finanziato dall'UE. Il secondo pilastro è finanziata dal fondo FEASR comprende le misure di sostegno allo sviluppo rurale ed è cofinanziato a livello nazionale.

#### **6.1.1.1- I° Pilastro della PAC: Pagamenti diretti**

I pagamenti diretti si dividono in diverse tipologie, alcuni dei quali sono obbligatori altri invece sono a discrezione degli stati membri. I regimi obbligatori sono: il pagamento di base, il pagamento 'Verde' oppure Greening ed infine il pagamento per i giovani agricoltori.

Il **Pagamento di Base** viene erogato sulla base del possesso, da parte dell'agricoltore, di un certo numero di diritti all'aiuto ("titoli"). Questi sono stati attribuiti ad ogni agricoltore sulla base di una ricognizione che teneva conto della superficie coltivata in un triennio cosiddetto "periodo di riferimento" (campagne 2000, 2001 e 2002). I diritti all'aiuto (o titoli) equivalgono a cedole di valore monetario fisso, strettamente collegato ad un ammontare in ettari: tipicamente un diritto all'aiuto corrisponde ad un ettaro di superficie. L'agricoltore per poter ottenere il pagamento dei diritti all'aiuto in suo possesso, deve dimostrare di coltivare una superficie almeno equivalente

alla superficie a cui ammontano i suoi diritti all'aiuto, coltivandovi colture ammissibili. Ad esempio, se un'azienda richiede il pagamento di 10 titoli dovrà dimostrare di coltivare almeno 10 ettari di terreno. Qualora un'azienda agricola non posseda titoli o è intenzionato ad acquisirne altri può effettuare una richiesta specifica, richiesta di attribuzione di titoli, affinché gli siano attribuiti nuovi titoli.

Il **Pagamento Verde**, meglio conosciuto con il termine Greening, è un aiuto supplementare al pagamento di base che viene riconosciuto a tutte quelle aziende agricole che utilizzano strumenti ed attuano metodologie che sono benefiche per il clima e l'ambiente. Gli stati membri sono obbligati a stanziare per il pagamento verde il 30% del loro stanziamento per i pagamenti diretti. Per ricevere il pagamento verde le aziende agricole sono obbligate a rispettare tre requisiti:

1. Diversificare le colture nell'ottica di rafforzare il terreno agricolo;
2. Mantenere e preservare i pascoli e prati permanenti che rappresentano un mezzo molto efficace per catturare il carbonio e contribuiscono a ridurre il riscaldamento globale;
3. Garantire la presenza di Aree di interesse ecologico nell'ottica di salvaguardare e migliorare la biodiversità nelle aziende agricole.

Il **Pagamento per i Giovani Agricoltori** è un supplemento al pagamento di base ed è concesso per un massimo di 5 anni dal momento in cui un giovane agricoltore, ovvero di età inferiore ai 40 anni, assume la gestione dall'azienda agricola. Per tale aiuto gli stati membri possono destinare fino al 2% degli stanziamenti totali per i pagamenti diretti.

Invece, tra i pagamenti facoltativi troviamo: sostegni accoppiati alla produzione, pagamenti per le zone soggette a vincoli naturali ed il pagamento redistributivo.

I **Sostegni Accoppiati alla Produzione** sono aiuti che erogano un finanziamento sulla base delle quantità prodotte dall'aziende. Lo scopo di questa tipologia di aiuto è quello di garantire un livello minimo di produzione

di uno o più prodotti in una certa regione o settore agricolo. Al sostegno accoppiato può essere destinato fino all'8% degli stanziamenti totali per i pagamenti diretti. Anche questo pagamento è supplementare al Pagamento di Base

Il **Pagamento Ridistributivo** è una tipologia di aiuto facoltativo che consiste in un sostegno supplementare al Pagamento di Base e viene calcolato sulla base degli ettari coltivati dall'azienda. Il numero di ettari per i quali può essere assegnato questa tipologia di aiuto è limitato e tale soglia è determinata da ogni singolo stato membro. L'importo per ettaro è uguale per tutti gli agricoltori e non può essere superiore del 65% del pagamento medio di base per ettaro. Al Pagamento Ridistributivo può essere destinato fino all'30% degli stanziamenti totali per i pagamenti diretti.

I **Pagamenti per le Zone soggetti a Vincoli Naturali** sono un sostegno supplementare al Pagamenti di Base e sono riconosciuti a tutte quelle aziende agricole che operano in aree dove l'agricoltura o il pascolamento di animali è reso più complicato da vincoli naturali. Queste aree sono individuate autonomamente da ciascun stato membro sulla base di alcuni criteri biofisici. Tale tipologia di pagamento è attualmente adottata solo dalla Danimarca e dalla Slovenia. Gli stati membri possono destinare a quest'aiuto fino ad un massimo del 5% degli stanziamenti totali per i pagamenti diretti.

Come è deducibile dalla descrizione appena effettuata dei pagamenti, non è possibile richiedere singolarmente le diverse tipologie di pagamento, in quanto possono essere richiesti solo in concomitanza del Pagamento di Base. In altre parole se non viene richiesto il Pagamento di base, le aziende agricole non possono effettuare la richiesta degli altri pagamenti.

Inoltre affinché le aziende agricole ricevano l'importo completo dei pagamenti diretti ai quali hanno diritto, esse devono rispettare una serie di norme dettate dall'Unione Europea in ambito di: sicurezza alimentare, salute e benessere animale, protezione dell'ambiente, protezione delle risorse idriche, ecc. In caso

contrario l'importo può essere penalizzato e la percentuale di penalizzazione dipende dalla misura in cui l'azienda ha violato le norme.

Esiste un'altra tipologia di pagamento per il quale vige un regime meno rigido: **Pagamento per i Piccoli Agricoltori**. Tale aiuto a differenza degli altri appena descritti non può essere richiesto in concomitanza con le altre tipologie. L'importo di questo pagamento non dipende ne dagli ettari coltivati ne dalla produzione agricola, ma assicura un importo che è deciso a livello nazionale e che non può superare 1250 €. Con questa tipologia di pagamento gli agricoltori non potranno incorrere in eventuali penalità pecuniarie e non sono obbligati a rispettare le altre norme europee.

#### **6.1.1.2- I° Pilastro della PAC: Misure di Mercato**

Il primo pilastro della PAC è completato dalle **Misure di Mercato** che non sono nient'altro che alcuni meccanismi messi in atto dall'unione europea per far fronte a crisi di mercato o settoriali imprevedibili. Un'emergenza sanitaria in ambito agricolo può provocare un calo della domanda imprevista oppure un eccesso di offerta di un prodotto può provocare una diminuzione di prezzo che può incidere in maniera rilevante sul reddito degli agricoltori. Le misure di mercato si distinguono in **Aiuti Finanziari** e in **Norme Regolatrici**. Gli aiuti finanziari hanno lo scopo di assicurare all'agricoltore il giusto reddito nel caso in cui sia stato colpito da un evento fortuito e non controllabile. Le norme regolatrici invece sono appunto delle regole che hanno il compito di regolare il mercato di alcuni prodotti che non sono in grado di auto equilibrarsi da soli. Tali norme possono incidere sulla produzione di un particolare prodotto stabilendo un limite sulla produzione oppure regolare gli scambi commerciali sia all'interno dell'Unione Europea che al di fuori. Le misure di mercato rappresentano una spesa molto bassa, impegnando solo il 5 % dell'intera spesa agricola europea.



### **6.1.1.3- 2° Pilastro della PAC: Politica di sviluppo Rurale**

Il secondo pilastro della PAC è composto dalle politiche di sviluppo rurale (PSR) e sono cofinanziate dall'Unione Europea tramite il fondo FEASR e da fondi nazionali o regionali. Gli obiettivi del PSR si riassumono in sei principali:

1. Favorire il trasferimento di conoscenze e l'innovazione nei settori Agricolo e forestale e nelle zone rurali;
2. Rafforzare la competitività dell'agricoltura e migliorare e stabilizzare la redditività delle aziende agricole;
3. Incentivare l'organizzazione della filiera agroalimentare ed attuare una migliore gestione dei rischi nel settore agricolo
4. Preservare, ripristinare e valorizzare gli ecosistemi dipendenti dall'agricoltura e dalle foreste;
5. Rendere la produzione agricola più efficiente per ridurre l'utilizzo di risorse ed incoraggiare un'agricoltura che abbia effetti positivi sull'ambiente e sul clima
6. Promuovere l'inclusione sociale, la riduzione della povertà e lo sviluppo economico.

La realizzazione della politica di sviluppo Rurale si basa sulla creazione da parte degli stati membri di programmi di sviluppo rurale personalizzati. In questo modo è possibile attuare strategie diverse che agiscono in contesti sociali, economici ed ambientali differenti. Nonostante i paesi membri abbiano una certa autonomia nella creazione dei propri programmi questi devono essere strettamente collegati agli obiettivi Europei. Inoltre questi programmi si basano su una combinazione di misure selezionate da un apposito elenco messo a disposizione dall'UE ed annoverate nel regolamento sullo sviluppo rurale (regolamento (UE) n. 1305/2013). I tassi di cofinanziamento differiscono tra i vari programmi e sono dipendenti dalle misure che lo compongono. Inoltre i

programmi prima di diventare operativi devono essere approvati dalla Commissione Europea e devono comprendere un piano di cofinanziamento.

Per la PAC 2014-2020 la commissione Europea ha approvato 118 programmi di sviluppo rurale presentati dai 28 stati membri. Venti stati membri hanno optato per un solo programma nazionale, mentre otto stati membri hanno presentato più di un programma per adeguarlo meglio alle caratteristiche naturali e sociali del territorio. Per l'Italia sono stati approvati, per il periodo 2014-2020, ventuno Programmi di sviluppo rurale regionali (19 regionali + 2 Province autonome).

### **6.1.2 – Pagamento Unico per l'Agricoltura (PUA)**

Il progetto PUA si inserisce all'interno del Sistema Informativo Agricolo Regionale (SIAR) con l'obiettivo di:

- Permettere alle aziende agricole ed ai loro intermediari CAA (Centri di Assistenza Agricola) di poter presentare in modo corretto ogni anno una richiesta per ricevere i finanziamenti, meglio conosciuta come Richiesta Unica di Finanziamento (RUF).
- Permettere all'organismo pagatore regionale di poter effettuare l'istruttoria delle RUF, calcolare l'importo da liquidare ed autorizzare il pagamento dei finanziamenti.

Con la presentazione di una RUF le aziende agricole non fanno altro che richiedere gli aiuti finanziari del 1° Pilastro della PAC, cioè i pagamenti diretti. Nel paragrafo 6.1.1.1 abbiamo spiegato quali fossero le tipologie di aiuto che le aziende agricole possono richiedere, differenziando tra gli aiuti obbligatori e quelli facoltativi. In particolare in Italia per il settennato 2014-2020 si è deciso di finanziare i seguenti aiuti:

1. Pagamento di base (obbligatorio)
2. Pagamento verde o 'Greening' (obbligatorio)
3. Pagamento per i giovani agricoltori (obbligatorio)

4. Pagamenti Accoppiati;

5. Pagamento per i Piccoli Agricoltori.

Dal 2014 è possibile presentare una sola richiesta per poter richiedere diverse tipologie di pagamenti diretti e per questo motivo la richiesta di finanziamento viene comunemente chiamata Richiesta Unica di Finanziamento (RUF). Per ogni tipologia di pagamento diretto le aziende agricole sono impegnate a presentare una serie di documenti e ad effettuare una serie di dichiarazioni obbligatorie specifiche per ogni tipologia di aiuto. In particolare le condizioni necessarie per presentare una RUF sono:

- L'azienda deve aver costituito il fascicolo aziendale presso l'Organismo Pagatore Regionale di competenza. Con il termine fascicolo aziendale si indica un insieme di documenti che raccoglie le informazioni principali dell'azienda agricola. Tale fascicolo è salvato su un'altra componente del SIAR, ovvero Anagrafe;
- L'attività agricola dell'azienda deve risultare l'attività principale o esclusiva dell'azienda;
- L'azienda possiede titoli o ha presentato la richiesta di attribuzione di diritti all'aiuto.

L'organismo Pagatore Regionale nel settennato 2014-2020 ha ricevuto in media ogni anno 42000 richieste di finanziamento unico ed ha autorizzato il pagamento di 350 milioni di Euro ogni.

#### **6.1.2.1 - Utenti e altre parti interessate**

Gli utilizzatori del sistema "Regime di pagamento unico", si suddividono in due macro tipologie:

- **Utente esterno a OPR:** invierà domande all'Organismo Pagatore. Accedendo esclusivamente alle funzionalità di front-office del sistema con visibilità limitata ai dati di propria competenza;

- **Utente interno a OPR:** avrà accesso a tutte le funzionalità del sistema ed avrà visibilità di tutti i dati.

Sono di seguito brevemente descritti gli attori coinvolti nel progetto.

- **Organismo Pagatore Istruttore:** Rappresenta i Funzionari dell'Organismo Pagatore che ricevono e istruiscono le pratiche pervenute dagli Intermediari e/o aziende agricole.
- **Regione:** Rappresenta i Funzionari Regionali appartenenti all'Assessorato della Regione.
- **Intermediario (CAA):** Rappresenta i Tecnici dei CAA (Centri di Assistenza Agricola), i quali possono svolgere attività in nome e per conto dei loro assistiti, da cui hanno ricevuto specifica mandato. Possono quindi gestire le richieste di più aziende.
- **Organismo Pagatore:** Rappresenta i Funzionari dell'Organismo Pagatore che effettuano funzioni di controllo delle pratiche e autorizzazione al pagamento delle stesse.

#### 6.1.2.2- Le interazioni di PUA

Come abbiamo sottolineato più volte PUA si inserisce all'interno del sistema informativo Regionale ed interagisce sia con componenti interne al SIAR che con altri SI.

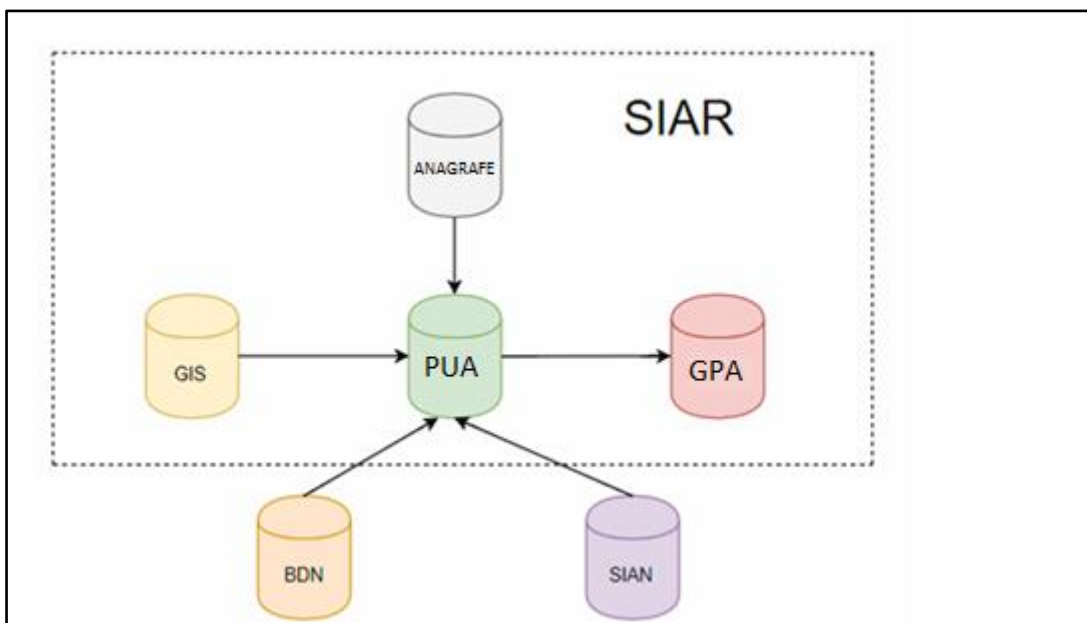


Figura 16- Sistema informativo Agricolo Regionale

In particolare, come è mostrato in figura, PUA interagisce con:

- **Anagrafe:** è la componente del SIAR sulla quale le aziende agricole costituiscono il proprio fascicolo aziendale, nel quale sono inserite le informazioni e i documenti riguardanti le aziende. Tra queste informazioni è presente anche l'elenco delle particelle di superfici possedute dall'azienda. L'insieme delle particelle comporrà la superficie posseduta dall'azienda.
- **Sistema Informativo Geografico (GIS):** è la componente interna al SIAR attraverso la quale PUA recupera informazioni riguardo le superfici dall'azienda Agricola. PUA confronta le superfici dichiarate su Anagrafe dall'azienda agricola con le superfici recuperate dal GIS.

Il confronto serve:

- Bloccare le particelle di superficie non presenti sul GIS o non foto interpretate correttamente;
- Determinare la superficie ammissibile (è la minore tra la superficie del GIS e la superficie dichiarata) dalle fotointerpretazioni delle foto aeree;

- **Gestione Pagamento Agricolo (GPA):** è la componente interna del SIAR alla quale PUA invia le liste di pagamento il quale si occuperà di procedere al pagamento effettivo alle aziende agricole tramite.
- **Banca Dati Nazionale Zootecnica (BDN)** viene utilizzata da PUA per recuperare informazioni riguardo:
  - Elenco Allevamenti;
  - Dati di Consistenza dell'allevamento;
  - Elenco Pascolo;
  - Dati di consistenza del Pascolo.
- **Sistema Informativo Agricolo Nazionale (SIAN)** fornisce una serie di informazioni raccolte a livello nazionale come:
  - Dati relativi al registro titoli (diritto all'aiuto);
  - Dati di anagrafe Tributaria;
  - Pratiche da bloccare per Arma;
  - Ecc.

Come è possibile dedurre dalla figura, PUA riceve informazioni da ANAGRAFE, SIAN, BDN, GIS mentre invia informazioni esclusivamente a GPA.

## **6.2- L'iter di una Richiesta Unica di Finanziamento**

Dal momento in cui l'azienda agricola, in prima persona o attraverso l'aiuto dei CAA, presenta una richiesta di finanziamento fino al pagamento della stessa, la pratica attraversa quattro fasi differenti in cui sono effettuati una serie di controlli e verifiche differenti sulla base degli aiuti richiesti. In particolare l'iter di una pratica RUF attraversa le seguenti fasi:

- Presentazione della RUF;
- Istruttoria della RUF
- Calcolo del premio;
- Generazione ed invio delle liste di Pagamento a GPA.

## **6.2.1 – Presentazione della RUF**

Le aziende agricole possono presentare la Richiesta Unica di Finanziamento ogni anno fino al 15 di maggio. La richiesta può essere inoltrata esclusivamente tramite il portale WEB messo a disposizione dell'Organismo Pagatore Regionale e può essere presentata in ritardo fino alla fine di maggio, ma per ogni giorno di ritardo sarà applicata una penalità che avrà impatto sul pagamento finale. Durante questa fase le aziende devono compilare una serie di quadri che sono specifici per ogni tipologia di aiuto. È possibile compilare i quadri in diverse sessioni e le informazioni ed i documenti forniti possono essere modificati fino al termine ultimo di presentazione. La presentazione della richiesta può essere ritenuta valida solo se questa supera tutti i controlli di presentazione. I controlli di presentazione sono effettuati dalle stesse aziende agricole tramite una funzionalità presente sul portale ed i controlli servono a verificare da una parte che siano stati forniti e/o allegati tutta la documentazione necessaria e dall'altra che l'azienda possa richiedere un determinato aiuto.

Dopo aver certificato che la richiesta ha superato tutti i controlli di presentazione è necessario 'Trasmettere' la Richiesta Unica di Finanziamento. L'organismo Pagatore ritiene valide solo le Domande che sono state trasmesse e considera le pratiche non valide, e quindi non aventi diritto di alcun pagamento, quelle che non sono state trasmesse anche se correttamente compilate. Per poter 'Trasmettere' una pratica è necessario che essa supera tutti i controlli di presentazione. Inoltre dopo la trasmissione di una pratica non è possibile più modificare le informazioni fornite ed i documenti allegati.

Come abbiamo anticipato la compilazione dei quadri obbligatori varia a seconda delle tipologie di aiuto richieste. Poiché il pagamento di base è l'unico aiuto che deve essere obbligatoriamente richiesto, di conseguenza sarà obbligatorio la compilazione dei relativi quadri. In particolare in questo quadro è necessario effettuare la compilazione del piano di utilizzazione che consiste nell'abbinare ogni particella di superficie dichiarata nel fascicolo aziendale ad

uso del suolo ammissibile, ovvero specificare l'attività agricola che viene svolta su tale superficie. Gli usi del suolo ammissibili si dividono in tre macro-categorie:

- Seminativi, il terreno è utilizzato per coltivazioni agricole oppure il terreno è disponibile per le coltivazioni agricole ma è lasciato a riposo;
- Prati permanenti e pascoli permanenti, il terreno è utilizzato per la coltivazione di erba o altre piante da foraggio;
- Colture permanenti, il terreno è occupato da colture per almeno 5 anni e forniscono raccolti ripetuti. Entrano in quest'ambito ad esempio le colture frutticole.

Per facilitare la compilazione è possibile utilizzare la funzionalità 'Compilazione automatica', che riconosce le superfici presenti nel fascicolo aziendale e associa gli usi che sono stati abbinati l'anno precedente. È possibile inoltre in un secondo momento apportare modifiche alla compilazioni automatiche e/o abbinare gli usi alle particelle non presenti nel fascicolo aziendale l'anno precedente.

## **6.2.2 – Istruttoria della Pratica**

L'istruttoria della Richiesta Unica è massiva ed informatica e tutte le pratiche che sono state trasmesse correttamente entro la fine di maggio vengono messe in istruttoria massivamente agli inizi di ottobre. In questa fase vengono eseguiti una serie di controlli per verificare se la richiesta di contributo è corretta ed non ha errori. L'istruttoria della richiesta unica avviene in modo graduale, cioè i vari controlli da effettuare al fine di verificare il rispetto di tutti i requisiti per ogni tipologia di aiuto non possono essere effettuati tutti nello stesso periodo a ottobre/novembre perché alcuni controlli dipendono da dati provenienti da fonti esterne non ancora disponibili all'apertura dell'istruttoria.

L'esecuzione dei controlli possono generare due tipologie di anomalie:

- Anomalie Bloccanti, che hanno impatto sul Calcolo del premio



- Anomalie Warning, che non hanno impatto sul calcolo del premio ma sono soltanto delle segnalazioni.

Ad ogni anomalia bloccante, se non viene risolta dove è possibile, è associata una percentuale di penalizzazione che avrà impatto durante la fase del calcolo del premio.

Le anomalie possono essere generate in funzione di:

- Incoerenze tra i dati dichiarativi presenti sulla Richiesta Unica e le banche dati certificate, che sono
  - G.I.S. (Sistema informativo geografico)
  - BDN (Banca dati nazionale degli allevamenti)
  - Anagrafe Tributaria
  - SIAN
  - Controllo manuale dei dati da parte dell'OPR
- Mancato rispetto della normativa comunitaria, nazionale e regionale da parte dei beneficiari nella stesura/compilazione della richiesta unica.

La presenza di anomalie bloccanti può portare al non pagamento totale o parziale del premio della richiesta.

Dopo la generazione delle anomalie viene data comunicazione ai CAA della situazione delle anomalie e viene dato un termine entro il quale sono ammesse le correttive. Con il termine correttiva si intende:

- Una rinuncia intera o parziale di un aiuto;
- Una modifica dei dati forniti in fase di presentazione.

La situazione delle anomalie può essere congelata in automatico dopo che è trascorso il tempo dato ai CAA e/o alle aziende agricole per la correttiva o può essere congelato dopo esplicita conferma di fine correttiva da parte dei CAA e/o azienda agricola sulla singola pratica.

Il congelamento dei controlli significa che:

- Non è più possibile presentare alcuna correttiva (né rinuncia o modifica dati, né correttiva con documento, né sincronizzazione con il fascicolo);
- Non vengono più rieseguiti i controlli su quella pratica;

- Lo stato delle anomalie non può essere più modificato.

Ogni anno l'OPR definisce:

- Quali sono i controlli da eseguire per l'istruttoria;
- Vengono indicati per quali aiuti i controlli devono essere eseguiti;
- Quali sono bloccanti e quali sono segnalazione (WARNING);
- Quali sono correggibili e di conseguenza quali tipi di correttive attivare;
- Il termine ultimo entro il quale è possibile effettuare una correttiva.

Il congelamento della situazione dell'anomalie comporta la fine della fase di istruttoria della pratica e l'inizio dell'ultima fase, 'Calcolo del premio e generazione delle liste di pagamento'.

### **6.2.3 – Calcolo del premio e generazione delle liste**

Una volta superato il termine concesso alle aziende agricole e/o ai CAA non è più possibile presentare una correttiva, la situazione delle anomalie è congelato e le pratiche entrano nell'ultima fase del progetto, ovvero quella che si occupa di determinare il premio per ciascuna richiesta. Di solito il termine ultimo per la presentazione delle correttive è il 15 febbraio dell'anno successivo all'anno in cui si è presentata la richiesta. L'ultima fase di una Richiesta Unica di Finanziamento è divisa in tre stadi:

- Calcolo del premio su tutte le pratiche dell'anno campagna
- Generazione delle liste di Pagamento
- Invio delle liste di Pagamento a GPA;

Per tutte le pratiche che hanno raggiunto l'ultima fase è eseguito il calcolo del premio che determina per ogni aiuto l'importo del premio che sarà liquidato. L'algoritmo del calcolo del premio genera per ogni aiuto di pagamento quattro valori:

- Il premio lordo (PL)
- La percentuale di penalizzazione (PP)

- Il premio netto (PN);
- Il premio totale (PT).

Il premio lordo corrisponde al premio che l'azienda agricola riceverebbe se per l'aiuto richiesto non sono presenti anomalie bloccanti, in altre parole se ha presentato correttamente tutte le informazioni obbligatorie ed ha rispettato tutti i requisiti richiesti dalla normativa. Nella seguente tabella viene analizzato in linea generale la modalità di calcolo della percentuale di penalizzazione:

Tipologia di Aiuto	Formula di calcolo	Range
Pagamento di base	$PP_i = (SB/STD)$ dove: <b>SB</b> =Superficie con anomalie Bloccanti. <b>STD</b> =Superficie totale Dichiarata, ovvero la superficie richiesta a premio.	$0 \leq PP_i \leq 1$
Greening	-	1
Pagamento per i giovani agricoltori	-	1
Sostegno Accoppiato	$PP_i = (QB/QTD)$ dove: <b>QI</b> =Quantità Bloccata, ovvero la quantità alla quale sono associate anomalie bloccanti. <b>QTD</b> =Quantità totale prodotta.	$0 \leq PP_i \leq 1$
Piccoli Agricoltori	-	0

Tabella 7 – Descrizione della formula di calcolo delle penalità per i differenti Aiuti di Pagamento

Dopo aver calcolato la percentuale di penalizzazione si calcola il premio netto che è data dalla seguente formula:

$$PN_i = PL_i(1 - PP_i)$$

Dopo aver calcolato il premio netto per ogni aiuto l'algoritmo calcola il Premio Totale che è data dalla somma dei singoli premi netti relativi a ciascun premio richiesto in fase di presentazione.

Dopo aver calcolato il premio per ciascun aiuto di pagamento, sono generate le liste di pagamento. Le liste di pagamento non sono altro che elenchi in cui sono riportati gli importi per ciascun aiuto ed altre informazioni necessarie per effettuare il pagamento. Per ogni anno campagna esistono più liste di pagamento e ciò è dovuto dal fatto che l'OPR non effettua il pagamento di tutti gli aiuti contemporaneamente, ma ogni lista è specifica per ogni aiuto. L'agricoltore che ha richiesto più aiuti riceverà pertanto tanti pagamenti, se meritevole, quante sono le tipologie di aiuto richiesto. Dopo che è stata generata la lista di pagamento essa viene trasmessa a GPA, la componente del SIAR che si occuperà di effettuare ulteriori controlli sui beneficiari e che si occuperà di autorizzare il pagamento.

I CAA e/o le aziende agricole possono conoscere le informazioni generate dal calcolo del premio tramite l'applicativo WEB in una apposita pagina chiamata 'Pagamento'. In questa pagina inoltre possono conoscere lo stato di avanzamento del pagamento di ciascun aiuto. Una volta che per una pratica sono stati pagati tutti gli aiuti richiesti, essa viene messa nello stato conclusa.

## 6.3 – L’istruttoria di pratiche con premi pascolo

Con il termine pratiche con premi pascolo si individuano un sottoinsieme di pratiche per le quali durante la compilazione del piano di utilizzazione è stata indicata su almeno una particella usi ammissibili appartenenti alla macro-categoria ‘Prati e pascoli permanenti’ (vedi paragrafo 6.2.1). La normativa richiede agli agricoltori sulle particelle con usi appartenenti alla categoria ‘Prati e pascoli permanenti’ di svolgere un’attività minima di pascolamento di 60 gg e che la densità animale (UBA/ha) calcolata su 365 giorni sia di almeno 0.2. Tuttavia in alcuni casi speciali, che saranno analizzati successivamente (vedi paragrafo 6.3.2), la normativa prevede dei limiti inferiori. Il rispetto dei requisiti previsti dalla normativa non è verificata a livello di una singola particella, ma a livello comunale. L’attività di pascolamento è limitata in alcuni periodi a seconda della quota:

<b>Altitudine (metri s.l.m.)</b>	<b>Inizio pascolamento</b>	<b>Termine pascolamento</b>
Inferiore a 800	01 Gennaio	31 Dicembre
Tra 800 e 1.500	31 Marzo	30 Ottobre
Oltre i 1.500	15 Maggio	15 Ottobre

*Tabella 8 – Periodi Pascolamento ammissibile a seconda dell’altitudine del Pascolo*

Il processo di istruttoria delle pratiche con premi pascolo inizia il 15 gennaio dell’anno successivo alla presentazione della richiesta e può essere diviso in tre fasi ben distinte:

1. Recuperare le informazioni riguardanti l’anagrafica zootecnica da BDN;
2. Eseguire l’algoritmo Calcolo Carico UBA;
3. Eseguire i controlli specifici, Controlli Pascoli, che verificano se gli agricoltori rispettano tutti i requisiti previsti dalla normativa.

Prima di studiare a fondo i tre stadi, analizzeremo alcune definizioni fondamentali utili per la comprensione dell’istruttoria di pratiche con premi pascolo.

<b>Termine</b>	<b>Descrizione</b>
Codice Stalla	È il codice univoco della stalla.
Detentore Allevamento	È la persona fisica o giuridica responsabile dei capi presenti nell'allevamento e può non coincidere con il proprietario dei capi.
Codice Pascolo	È codice univoco che identifica un pascolo;
Detentore dei capi al pascolo	È la persona fisica o giuridica responsabile dei capi al pascolo.
UBA	Unità di bestiame adulto. Le UBA sono una modalità standard di conversione delle singole specie e categorie di bestiame, al fine di un loro confronto. (vedi tabella)
UBA dell'allevamento	È un valore che è calcolato a livello di comune ed è la somma dei capi, espressi in UBA, presenti mediamente in uno o più allevamenti ubicati in un comune a premio pascolo.
UBA al pascolo	È un valore che è calcolato a livello di comune ed è la somma dei capi, espressi in UBA, mediamente presenti in uno o più pascoli ubicati in un comune a premio pascolo.
Superficie a Pascolo (ha)	È la somma delle superfici delle particelle richieste a premio pascolo raggruppate per comune.
Coefficiente minimo UBA	È la densità minima animale che deve essere soddisfatto sulle superfici richieste a premio pascolo.
Carico UBA Stalla	Rappresenta la densità animale È un valore che si calcola a livello di comune ed è dato dal rapporto di: $\frac{\text{UBA dell'allevamento}}{\text{Superficie a Pascolo}}$

Carico UBA Pascolo	È un valore che si calcola a livello di comune ed è dato dal rapporto di: $\frac{\text{UBA del pascolo}}{\text{Superficie a Pascolo}}$
--------------------	--

Tabella 9 – Definizioni utili per comprendere il processo di istruttoria delle pratiche con premi pascolo

Le specie animali valide ai fini del calcolo delle UBA sono: bovini, ovini, cavalli, bufalini, caprini, bardotti, asini e muli. Per calcolare correttamente le UBA in un determinato comune è necessario utilizzare la seguente tabella di conversione degli animali che associa ad ogni animale un fattore di conversione. In questo modo risulta è possibile calcolare in modo oggettivo la densità animale su una determinata superficie.

Specie-Codice	Fascia età	Fattore di conversione
Bovini-0121, Cavalli-0126, Bufalini-0129, Bardotti-0148, Asini-0149, Muli-0147	0-6 mesi	0.4
Bovini-0121, Cavalli-0126, Bufalini-0129, Bardotti-0148, Asini-0149, Muli-0147	6-24 mesi	0.6
Bovini-0121, Cavalli-0126, Bufalini-0129, Bardotti-0148, Asini-0149, Muli-0147	Oltre 24 mesi	1
Ovini	-	0.15
Caprini	-	0.15

Tabella 10 – Fattore di conversione delle diverse specie in base all'età

Nel paragrafo 6.3.2.1 saranno mostrati alcuni esempi pratici per il calcolo delle Uba della Stalla e Uba del Pascolo.

### 6.3.1 – Come recuperare le informazioni da BDN

Come anticipato nei paragrafi precedenti, PUA comunica con BDN per recuperare le informazioni riguardo l'anagrafica zootecnica. In particolare a fronte di alcuni dati in input si interroga la BDN tramite Web Services, la quale restituisce in output informazioni che sono salvati nella base dati di PUA che saranno utilizzati dall'algoritmo 'Calcolo UBA'. Per recuperare

tutte le informazioni utili è necessario effettuare 4 tipologie differenti di interrogazioni:

<b>Nome interrogazione</b>	<b>Parametri input</b>	<b>Descrizione</b>
1° Anagrafica della Stalla	Cuaa, anno	Restituisce le informazioni di tutte le stalle per le quali il CUAa passato in input è il detentore o proprietario della stalla il 15/05/ANNO.
2° Consistenza di Stalla	Codice stalla, anno	Restituisce il numero di animali presenti mediamente durante l'anno nella stalla passata in input suddivisi per specie, fascia d'età.
3° Anagrafica del Pascoli	Cuaa, anno	Restituisce le informazioni di tutti i pascoli sui quali hanno pascolato animali che provengono da stalle il cui detentore è il CUAa passato in input. Restituisce le seguenti informazioni: <ul style="list-style-type: none"> <li>• Codice pascolo</li> <li>• Istat Comune</li> </ul>
4° Consistenza del Pascolo	Codice pascolo, anno	Restituisce per ogni animale transitato al pascolo durante l'anno le seguenti informazioni: <ul style="list-style-type: none"> <li>• Data di nascita;</li> <li>• Data di ingresso al pascolo;</li> </ul>



		<ul style="list-style-type: none"> <li>• Data di rientro dal pascolo;</li> <li>• Specie animale</li> <li>• Codice stalla</li> </ul>
--	--	---

Tabella 11 – Classificazione delle interrogazioni di BDN

Le interrogazioni di BDN sono effettuate massivamente ovvero le interrogazioni per tutte le pratiche a premio pascolo sono effettuati in cascata nello stesso istante tramite un batch-job che acquisisce i parametri di input da una specifica tabella. Le interrogazioni della BDN sono effettuati in un ordine ben preciso perché alcuni dei parametri di input delle interrogazioni 2 e 4 corrispondono ad informazioni forniti in output dalle interrogazioni 1 e 3. Pertanto saranno prima effettuate in parallelo le interrogazioni ‘Anagrafica della stalla’ e ‘Anagrafica del pascolo’ e solo successivamente le interrogazioni ‘Consistenza della stalla’ e ‘Consistenza del pascolo’. È molto importante effettuare le interrogazioni nel modo corretto e nell’ordine giusto, altrimenti i parametri che calcolerà la procedura ‘Calcola Carico UBA’ saranno errati ed un errore in questo punto dell’intero processo potrebbe avere conseguenze gravi, su tutto il processo di istruttoria della pratica.

### 6.3.2 – L’algoritmo del Calcolo Carico UBA

Una volta terminate le interrogazioni BDN, viene eseguito su tutto le pratiche a premio pascolo l’algoritmo del Calcolo Carico UBA. Si tratta di un algoritmo che elabora i dati provenienti da BDN ed i dati dichiarativi recuperati dal piano culturale per calcolare una serie di parametri che saranno fondamentali nel momento in cui una pratica con premio pascolo sarà sottoposta ai controlli pascoli. In particolare l’algoritmo esegue le seguenti operazioni:

1. **Determina i Comuni a Premio Pascolo:** L’algoritmo recupera dal piano culturale quali sono i comuni sui quali esiste almeno una particella abbinata a premio pascolo;

**2. Calcola la Superficie a Pascolo per Comune (SPC):** Somma della superficie delle particelle abbinate a premio pascolo raggruppate per Comune;

**3. Determina l'Altitudine Media del Comune (AMC):** Per la determinazione dell'altitudine occorrerà fare riferimento al collocamento del centroide delle particelle classificate a pascolo e calcolare l'altitudine media del comune (AMC) con la seguente formula:

$$AMC_i = \frac{\sum_{j=1}^n sp_j \times ac_j}{\sum_{j=1}^n sp_j}$$

dove: AMC<sub>i</sub>= Altitudine media del comune i-esimo espressa in metri;  
sp<sub>i</sub>=Superficie della particella j-esima appartenente al comune i-esimo;

ac<sub>j</sub>=altitudine del centroide della particella j-esima;

**4. Determina il Coefficiente Minimo UBA (CMU):** la normativa prevede un coefficiente per ogni comune ed assegna:

- a. Un carico minimo UBA di 0,2 UBA/ettaro quando la superficie a pascolo di un comune è posta ad un'altezza inferiore a 1.000 s.l.m.,
- b. Un carico minimo UBA di 0,15 UBA/ettaro quando la superficie a pascolo di un comune è posta ad un'altezza superiore a 1.000 s.l.m.,
- c. Un carico minimo UBA di 0,1 UBA/ettaro quando la superficie a pascolo di un comune è posta ad un'altezza superiore a 2.000 m s.l.m.

**5. Determina il periodo minimo di pascolamento (PMP):** la normativa prevede un Periodo minimo di pascolamento per ogni comune ed assegna:

- a. Un periodo di pascolamento minimo di 60 giorni quando la superficie a pascolo di un comune è posta ad un'altezza inferiore a 1.000 s.l.m.,
- b. Un periodo di pascolamento di 50 giorni quando la superficie a pascolo di un comune è posta ad un'altezza superiore a 1.000 s.l.m.,
- c. Un periodo di pascolamento di 30 giorni quando la superficie a pascolo di un comune è posta ad un'altezza superiore a 2.000 m s.l.m.

**6. Calcola Carico UBA Stalla (CUS):** Il Carico Uba Stalla è calcolato sulla base delle informazioni ricavate dalle interrogazioni 1 e 2. La procedura verifica se nei comuni a premio pascoli sono presenti stalle per le quali il beneficiario della pratica risulta essere il detentore. Se non risultano stalle, il Carico Uba stalla è uguale a zero. Altrimenti l'algoritmo calcola la densità animale come:

$$CUS_i = \frac{\sum_{j=1}^n NC_j \times FC_j}{SPC_i}$$

Dove CUS<sub>i</sub>=Carico Uba Stalla del comune i-esimo;

NC<sub>j</sub>=Numero dei capi di una specie animale in una certa fascia-età;

FC<sub>j</sub>= Fattore di conversione in UBA

SPC<sub>i</sub> =Superficie a Pascolo del Comune i-esimo.

**7. Calcola i Giorni di Pascolamento (GP):** i giorni di pascolamento sono calcolati sulla base delle informazioni ricavate dalle interrogazioni 3 e 4. Se l'interrogazione Anagrafica del Pascolo non ha restituito nessuno Codice pascolo situato nel comune che l'algoritmo sta processando, allora i Giorni di pascolamento saranno uguali a zero. Altrimenti l'algoritmo calcola i Giorni di Pascolamento come:

$$GP_i = \frac{\sum_{i=1}^n (Dt\_rientro - dt\_ingresso)_i}{NC_i}$$

dove:

GP<sub>i</sub>= Media dei Giorni Pascolamento sul comune i-esimo;

Dt\_rientro= data di uscita dal Pascolo del capo i-esimo proveniente da una stalla in cui il beneficiario della pratica è anche detentore;

Dt\_ingresso= data di ingresso dal Pascolo del capo i-esimo proveniente da una stalla in cui il beneficiario della pratica è anche detentore;

NC=Numero dei capi totali che hanno pascolato sul comune in esame e che provengono da una stalla in cui il beneficiario della pratica è anche detentore;

Saranno esclusi, ai fini del calcolo dei Giorni di pascolamento, i capi che non rispettano i limiti del periodo di pascolamento a seconda della quota, presentati nel paragrafo 6.3.

**8. Calcolo carico UBA Pascolo (CUP):** i giorni di pascolamento sono calcolati sulla base delle informazioni ricavate dalle interrogazioni 3 e 4. Se l'interrogazione Anagrafica del Pascolo non ha restituito nessuno Codice pascolo situato nel comune che l'algoritmo sta processando, allora il CUP sul comune sarà uguale a zero. Altrimenti l'algoritmo calcola il CUP come:

$$CUP_i = \frac{\sum_{j=1}^n (Dt\_rientro - dt\_ingresso)_j \times FC_j}{365 \times SPC_i}$$

Saranno esclusi, ai fini del Calcolo carico UBA pascolo, i capi che non rispettano i limiti del periodo di pascolamento a seconda della quota, presentati nel paragrafo 6.3.

**9. Salva le informazioni in una tabella precisa:** Dopo aver memorizzato i parametri calcolati nei punti precedenti in una tabella specifica la procedura del Calcolo Carico Uba sulla singola pratica termina.

Dopo che la procedura del Calcolo Carico Uba termina per tutte le pratiche, allora si può affermare che la seconda fase dell'istruttoria delle pratiche con premio pascolo sia terminata.

### **6.3.2.1 – Esempio di calcolo del Carico UBA di stalla**

Supponiamo di voler calcolare il Carico UBA di stalla di un certo beneficiario in un certo comune. Supponiamo che il beneficiario sia il detentore di due stalle sul comune di Torino e nelle stalle durante tutto l'anno sono presenti gli animali mostrati nella seguente tabella.

Codice Stalla	Specie	Fascia d'età	Fattore di conversione	Capi animali	UBA
000TO001	0121-Bovini	Oltre-24	1	5	5
000TO001	0121-Bovini	6-24 mesi	0.6	10	6
000TO001	0124-Ovini	-	0.15	20	3
<b>Totale UBA 1° Stalla</b>					<b>15</b>
000TO111	0125-Caprini	-	0.15	30	4.5
000TO111	0121-Bovini	0-6 mesi	0.4	20	8
<b>Totale UBA 2° Stalla</b>					<b>12.5</b>
<b>Totale UBA Stalla</b>					<b>27.5</b>
<b>Superficie richiesta a premio sul comune di Torino (ha)</b>					<b>100</b>
<b>Carico Uba Stalla (uba/ha)</b>					<b>0.275</b>

Tabella 12- Esempio di Calcolo del Carico Uba Stalla

Applicando la formula del punto del punto 6 del paragrafo 6.3.2, il carico Uba stalla sarà uguale a 0.275 uba/ha. Nella tabella sono mostrati tutti i calcoli per arrivare al risultato.

### 6.3.2.2 – Esempio di calcolo del Carico Uba al Pascolo

Supponiamo di voler calcolare le UBA al Pascolo di un certo beneficiario in un certo comune. Supponiamo che sul pascolo presente nel comune di Torino abbiano pascolato capi provenienti da stalle il cui detentore di stalla è il nostro beneficiario. I dati relativi alle informazioni dei capi al pascolo sono mostrati nella seguente tabella

Codice Pascolo	Capo	Specie	Fascia d'età	Coeff. Conv	Data di ingresso pascolo	Data di uscita pascolo	GG di Pascola.	UBA
000TO001	A	0121	Oltre-24	1	01-06-18	13-09-18	73	0.2
000TO001	B	0121	Oltre-24	0.6	01-06-18	13-09-18	73	0.2
000TO001	C	0121	Oltre-24	0.15	01-06-18	13-09-18	73	0.2
000TO001	D	0121	Oltre-24	0.15	01-06-18	13-09-18	73	0.2
000TO001	E	0121	Oltre-24	0.15	01-06-18	13-09-18	73	0.2
<b>Totale UBA 1° Pascolo</b>								<b>1</b>
<b>Totale UBA Pascolo</b>								<b>1</b>

<b>Superficie richiesta a premio sul comune di Torino (ha)</b>	<b>5</b>
<b>Carico Uba Pascolo (uba/ha)</b>	<b>0.2</b>

Tabella 13 – Esempi di Calcolo del Carico UBA al Pascolo

Applicando la formula del punto del punto 9 del paragrafo 6.3.2, il carico Uba al Pascolo sarà uguale a 0.2uba/ha. Nella tabella sono mostrati tutti i passaggi dei calcoli per arrivare al risultato.

### 6.3.3 – I controlli Pascoli

Una volta che per tutte le pratiche a premio pascolo è stata eseguita la procedura ‘Calcolo Carico Uba’ è possibile eseguire i controlli pascoli ed istruire la pratica. In quest’ultima fase del processo di istruttoria delle pratiche con premi pascolo si stabilisce se i beneficiari sono meritevoli del pagamento e quindi hanno rispettato tutti i requisiti necessari. L’organismo pagatore regionale ha previsto 4 diversi controlli bloccanti:

<b>Codice del controllo</b>	<b>Ordine</b>	<b>Descrizione</b>
P01	1	Carico Uba Stalla e Carico Uba del Pascolo entrambi uguali a zero.
P02	2	Carico uba di Stalla è minore del Coefficiente Minimo UBA.
P03	3	Giorni di Pascolamento inferiore al periodo minimo di pascolamento.
P04	4	Carico del Pascolo è minore del Coefficiente Minimo UBA.

Tabella 14- Classificazione dei controlli Pascoli

Se il comune a premio Pascolo non supera i controlli P01, P03 e P04 il sistema non effettuerà ulteriori controlli. Il beneficiario non ha rispettato i requisiti

previsti dalla normativa su quel comune. Ciò comporta la generazione di una penalità che porterà ad una riduzione del calcolo del premio.

Se, invece, il comune a premio pascolo supera il controllo P02 allora il beneficiario ha rispettato i requisiti previsti dalla normativa sul comune e non sono necessarie ulteriori verifiche.

### **6.3.4- Adeguamenti del Software effettuati per l'anno 2018**

Ogni anno dopo il termine ultimo di presentazione delle domande, l'ente OPR fornisce l'elenco dei controlli da effettuare nella fase di istruttoria. In questa fase l'ente fornisce sia requisiti di nuovi controlli da implementare sia i requisiti per modificare la logica dei controlli già implementati per adeguarsi al cambiamento della normativa.

Nel caso specifico dell'istruttoria delle pratiche a premio pascolo ciò si traduce nel modificare, nella maggior parte delle volte, la sola logica di calcolo seguita dall'algoritmo 'Calcolo Carico Uba'. Nel paragrafo 6.3.2 abbiamo presentato l'algoritmo già con gli adeguamenti per l'anno campagna 2018. In questo paragrafo ci concentreremo nell'analizzare quali sono state le modifiche ed in particolare su quali controlli queste modifiche hanno avuto impatto.

#### **6.3.4.1 – Adeguamento del calcolo dell'altitudine media dei Comuni**

Prima dell'adeguamento del 2018, la normativa prevedeva che il calcolo dell'altitudine media non fosse calcolata per Comune, ma sull'intera pratica. In altre parole veniva calcolato l'Altitudine Media delle Particelle (AMP) a premio pascolo senza fare distinzioni sull'ubicazione della particella.

#### **6.3.4.2 – Adeguamento della determinazione del Coefficiente Minimo**

##### **Uba**

Prima dell'adeguamento del 2018, la normativa prevedeva un unico coefficiente per tutti i comuni della pratica, anche se questi erano ubicati ad altitudini diverse. La normativa assegnava:

- a. Un carico minimo UBA di 0,2 UBA/ettaro quando l'altitudine media delle particelle è inferiore a 1.000 s.l.m.,
- b. Un carico minimo UBA di 0,1 UBA/ettaro quando l'altitudine media delle particelle è superiore a 1.000 s.l.m.,

L'adeguamento effettuato per l'anno campagna 2018 ha previsto un coefficiente per ogni comune ed ha introdotto sia una nuova fascia altimetrica che un nuovo coefficiente.

#### **6.3.4.3 – Adeguamento sulla determinazione del periodo minimo di**

##### **Pascolamento**

Prima dell'adeguamento del 2018, la normativa prevedeva un unico periodo minimo di pascolamento di 60 giorni, senza prevedere deroghe nel caso in cui il comune fosse ubicato ad una certa altitudine.

#### **6.3.4.4 – Adeguamento sul calcolo dei giorni di pascolamento e del Carico**

##### **Uba Pascolo.**

Prima dell'adeguamento del 2018, la normativa non limitava il periodo di pascolamento a seconda dell'altitudine. Pertanto erano considerate corrette tutte le movimentazioni al pascolo che fossero caratterizzate da una data di ingresso e di uscita dal pascolo avvenute nell'anno di presentazione della richiesta.

Con l'adeguamento previsto per l'anno campagna 2018 la normativa pone dei limiti al periodo di pascolamento (vedi paragrafo 6.3). Pertanto l'algoritmo del



Calcolo Carico Uba dovrà considerare valide, ai fini del calcolo dei GP, le movimentazioni che rispettano suddetti limiti.

## **Capitolo 7 - Costruzione di indicatori di Performance per il monitoraggio e controllo della fase di Testing**

In questo capitolo l'autore si concentrerà sulla definizione di indicatori di performance per valutare la fase di test condotta sugli adeguamenti apportati all'istruttoria pascoli del software PUA per l'anno campagna 2018. Il Capitolo è diviso in due parti. Nella prima parte l'autore analizzerà:

- L'esecuzione della fase di test del progetto PUA;
- Il funzionamento del Software 'TSD' (Tracking Software Development, Software per il), in uso dall'azienda per la gestione ed il monitoraggio degli sviluppi software, per chiarire l'origine e la qualità dei dati utilizzati per la costruzione degli indicatori;

Nella seconda parte l'autore si occuperà di:

- Definire e misurare indicatori specifici per la valutazione della fase di test;
- Suggestire un piano di intervento concreto per migliorare le misure degli indicatori;

### **7.1 – Fase di Test del progetto PUA**

La fase di test del progetto PUA consta di tre passi distinti:

- 1° Definizione dei casi di test;
- 2° Esecuzione dei test da una risorsa interna (Tester);
- 3° Esecuzione dei test di accettazione da parte del cliente;

I primi due passi sono eseguiti dal tester, mentre l'ultimo passo è eseguito dal cliente che si assicura che il software o le modifiche effettuate rispettano i requisiti.

Il tester inizia a definire i casi di test parallelamente alla fase di sviluppo grazie ai documenti redatti dall'analista e dal progettista. Il tester dovrà costruire casi per condurre i test a tre livelli di progettazione diversi ed in particolare per testare:

- I singoli sviluppi (Test dei moduli),
- L'integrazione degli sviluppi (Test di integrazione),
- L'intero sistema (Test di sistema).

Una volta terminata la fase di sviluppo, il tester potrà incominciare a testare il software con i casi di test predefiniti. In alcuni casi la fase di test può incominciare anche prima della fine della fase di sviluppo. Ciò può verificarsi quando lo sviluppo di alcuni moduli è già terminato ed il tester ha definito di tutti i casi di test.

Al tester è richiesto di indicare, tramite l'applicativo TSD, l'inizio e la fine di ciascuna fase del processo di test ed indicare i casi di test che sono associati a ciascuna fase.

Una volta che il Tester conclude i Test, il cliente effettua i propri test, Test di accettazione. Il cliente si occupa in primo luogo di verificare gli stessi casi di test condotti dal tester e se ne riterrà opportuno ne effettuerà degli altri.

Ogni qualvolta che il tester o il cliente individuano un errore, essi segnaleranno l'errore tramite il software TSD, che aprirà in automatico una richiesta di risoluzione allo sviluppatore di competenza.

La fase di test si conclude quando tutti gli errori individuati sono stati sanati. È possibile tuttavia, in casi particolari, che si decida di concludere la fase di test nonostante ci siano errori non risolti, per proseguire con la fase di rilascio. Gli errori saranno poi gestiti in un secondo momento.

## 7.2 – Software di tracciamento delle fasi di sviluppo

TSD è un software di supporto allo sviluppo software, utilizzato per tracciare e monitorare in tutte le sue fasi lo sviluppo di un nuovo Software o lo sviluppo di nuovi requisiti. Il Software è accessibile a tutti gli stakeholder e permette di:

- Inserire una nuova richiesta di sviluppo da parte del cliente;
- Monitorare e conoscere lo stato di avanzamento dello sviluppo;
- Effettuare segnalazioni di errore dalle parte interessate;

L'utilizzo del software ci permette di salvare alcune informazioni riguardo lo sviluppo software che sono elencate di seguito:

1. Data di inserimento di una nuova richiesta di sviluppo;
2. Inizio e fine della fase di analisi;
3. Inizio e fine della fase di sviluppo;
4. Inizio e fine della fase di test interna: per questa fase è possibile conoscere:
  - a. Il numero dei casi di test;
  - b. I passi di ciascun caso;
  - c. I casi di test che hanno prodotto un errore;
  - d. L'inizio e la fine risoluzione di ciascun errore;
5. Inizio e fine della fase di test condotta dal cliente (test di accettazione): per questa fase è possibile conoscere:
  - a. Il numero dei casi di test effettuati;
  - b. I passi di ciascun caso;
  - c. I casi di test andati in errore;
  - d. L'inizio e la fine risoluzione di ciascun errore;
6. Data di consegna del software al cliente, ovvero data del rilascio;
7. Segnalazioni di errori dopo la consegna del software: per questa fase è possibile conoscere:
  - a. Il numero di errori;
  - b. I passi per ricreare ciascun errore;

c. L'inizio e la fine risoluzione di ciascun errore;

Per la costruzione degli indicatori di performance saranno utilizzati i dati che sono raccolti e salvati dal software TSD a partire dal 4 punto in poi.

Sono elencate di seguito le informazioni principali per la valutazione della fase di test che si possono recuperare grazie al Software TSD:

1. Numero di errori individuati in ogni fase di test:
  - a. Numero di errori individuati durante i test interni:
    - i. Numero di errori durante i test dei moduli;
    - ii. Numero di errori durante i test di integrazione;
    - iii. Numero di errori durante i test di sistema;
  - b. Numero di errori individuati durante i test di accettazione;
2. Numero di errori individuati dopo la fase di rilascio;
3. Numero di errori risolti:
  - a. Numero di errori risolti prima della fase di rilascio;
  - b. Numero di errori risolti individuati dopo la fase di rilascio;
4. Durata totale della fase di test:
  - a. Durata dei test di dei moduli;
  - b. Durata dei test di integrazione;
  - c. Durata dei test di sistema;
  - d. Durata dei test di accettazione;
5. Durata totale della risoluzione di errori:
  - a. Durata totale della risoluzione di errori individuati prima della fase di rilascio;
  - b. Durata totale della risoluzione di errori individuati dopo la fase di rilascio;
6. Numero dei casi di Test effettuati:
  - a. Numero di casi di test per i test dei moduli;
  - b. Numero di casi di test per i test di integrazione;
  - c. Numero di casi di test per i test di sistema;
  - d. Numero di casi di test per i test di accettazione;

Nella seguente tabella è possibile trovare i valori dei dati appena elencati limitatamente agli sviluppi effettuati per gli adeguamenti dell'istruttoria pascoli delle RUF. La raccolta dei dati è stata effettuata il 16 febbraio 2019, data in cui si è concluso il processo di istruttoria e data dopo la quale le anomalie che sono state rilevate non possono essere più risolte.

<b>Nome Parametro</b>	<b>Valore del parametro</b>
Numero di errori durante i test dei moduli	5
Numero di errori durante i test di integrazione	7
Numero di errori durante i test di sistema	6
Numero di errori individuati prima dei test di accettazione	15
Numero di errori individuati durante i test di accettazione	6
Numero di errori individuati prima del rilascio	21
Numero di errori individuati dopo il rilascio	4
Numero totale di Errori individuati	25
Numero di errori risolti prima del rilascio	21
Numero di errori risolti dopo il rilascio	4
Numero di errori risolti totali	25
Durata dei test di dei moduli;	15h
Durata dei test di integrazione;	23h
Durata dei test di sistema;	21h
Durata dei test di accettazione;	28h
Durata totale dei test	87h
Durate totale di risoluzione di errori individuati prima del rilascio	11h

Durata di risoluzione di errori individuati dopo il rilascio	4h
Durata di risoluzione totale	15h
Numero di casi di test per i test dei moduli	21
Numero di casi di test per i test di integrazione	34
Numero di casi di test per i test di sistema	12
Numero di casi di test per i test di accettazione	76
Numero totale di casi di test	143

Tabella 15 – Elenco dei parametri messi a disposizione dal Software TSD

### 7.3 – Definizione degli Indicatore di Performance per la fase di Testing

In questo paragrafo, facendo riferimento alla teoria presentata nel capitolo 5, sono definiti indicatori di performance che appartengono a categorie diverse. Ogni indicatore sarà descritto attraverso una scheda di definizione che sarà utile per specificare al meglio tutte le caratteristiche principali.

#### PEPTA –Percentuale di Errori individuati dal tester in ambiente di Test

<b>Codice/Nome Completo</b>	PEPTA – PERCENTUALE di ERRORI individuati PRIMA dei TEST di ACCETTAZIONE
<b>Categoria</b>	Indicatore di qualità
<b>Descrizione</b>	Fornisce la percentuale di errori individuati dal tester prima di consegnare il software al cliente per i test di accettazione
<b>Metodo</b>	Le voci che compongono l'indicatore sono misurate grazie al software TSD
<b>Formula</b>	$\frac{\text{Errori individuati prima dei test di accettazione}}{\text{Errori individuati prima del rilascio}}$
<b>Interpretazione</b>	L'indicatore può assumere valori compresi tra 0 e 1. Il valore ideale è uguale a 1, ciò vuol dire che la fase di test condotta dal tester è di ottima qualità, poiché il cliente nonostante una conoscenza migliore dei requisiti e dell'intero processo, non è riuscito ad individuare errori nel software.

<b>Obiettivo</b>	PEPTA $\geq$ 0.8
<b>Scala di misura</b>	Scala di Rapporto
<b>Unità di misura</b>	Nessuna
<b>Fonte</b>	Software TSD

### PERPR – Percentuale di Errori Risolti Prima del Rilascio

<b>Codice e Nome Completo</b>	PERPR – PERCENTUALE di ERRORI RISOLTI PRIMA del RILASCIO
<b>Categoria</b>	Indicatore di qualità
<b>Descrizione</b>	Fornisce la percentuale di errori risolti rispetto agli errori totali individuati prima del rilascio
<b>Metodo</b>	Le voci che compongono l'indicatore sono misurate grazie al software TSD
<b>Formula</b>	$\frac{\text{Errori risolti prima della consegna finale del Software}}{\text{Errori individuati prima della consegna finale del Software}}$
<b>Interpretazione</b>	L'indicatore può assumere valori compresi tra 0 e 1. Il valore ideale è uguale a 1, ciò vuol dire che gli errori individuati dal tester e dal cliente sono stati completamente risolti.
<b>Target</b>	PERPR $\geq$ 0.9
<b>Scala di misura</b>	Scala di Rapporto
<b>Unità di misura</b>	Nessuna
<b>Fonte</b>	Software TSD

### PEP- Percentuali di Errori individuati nell'ambiente di Produzione

<b>Codice e Nome Completo</b>	PEP- Percentuali di Errori individuati nell'ambiente di Produzione
<b>Categoria</b>	Indicatore di qualità
<b>Descrizione</b>	Fornisce la percentuale di errori individuati dopo la consegna finale del software
<b>Metodo</b>	Le voci che compongono l'indicatore sono misurate grazie al software TSD.
<b>Formula</b>	$\frac{\text{Errori individuati dopo la consegna finale del Software}}{\text{Errori individuati totali}}$



<b>Interpretazione</b>	L'indicatore può assumere valori compresi tra 0 e 1. Il valore ideale è uguale a 0; ciò vuol dire che non esistono errori individuati dopo la consegna finale del Software
<b>Target</b>	$PEP \leq 0.2$
<b>Scala di misura</b>	Scala di Rapporto
<b>Unità di misura</b>	Nessuna
<b>Fonte</b>	Software TSD

### **PENR – Percentuale di Errori non risolti**

<b>Codice e Nome Completo</b>	PENR – PERCENTUALE di ERRORI NON RISOLTI
<b>Categoria</b>	Indicatore di qualità
<b>Descrizione</b>	Fornisce la percentuale di errori conosciuti ma non risolti rispetto a tutti gli errori conosciuti.
<b>Metodo</b>	Le voci che compongono l'indicatore sono misurate grazie al software TSD.
<b>Formula</b>	$\frac{\text{Errori individuati non risolti}}{\text{Errori individuati Totali}}$
<b>Interpretazione</b>	L'indicatore può assumere valori compresi tra 0 e 1. Il valore ideale è uguale a 0, ciò vuol dire che non esistono errori individuati, ma non risolti.
<b>Target</b>	$PENR \leq 0.1$
<b>Scala di misura</b>	Scala di Rapporto
<b>Unità di misura</b>	Nessuna
<b>Fonte</b>	Software TSD

### **TMIE – Tempo medio di individuazione di un errore**

<b>Codice e Nome Completo</b>	TMIE – Tempo medio di individuazione di un errore nella fase di test
<b>Categoria</b>	Indicatore di servizio
<b>Descrizione</b>	Fornisce il tempo medio di individuazione di un errore prima della consegna finale del Software.
<b>Metodo</b>	Le voci che compongono l'indicatore sono misurate grazie al software TSD.

<b>Formula</b>	$\frac{\text{Durata totale della fase di test}}{\text{Errori individuati prima della consegna finale del Software}}$
<b>Interpretazione</b>	L'indicatore può assumere valori maggiori di zero. Valori bassi dell'indicatore sono da preferire, perché vuol dire che gli errori sono individuati rapidamente. Attenzione al valore zero che può assumere l'indicatore, può significare che la fase di test non abbia funzionato bene.
<b>Target</b>	$TMIE \leq 3 \text{ h/errore}$
<b>Scala di misura</b>	Scala di Rapporto
<b>Unità di misura</b>	Minuti/errore
<b>Fonte</b>	Software TSD

### TMRE – Tempo medio di risoluzione di un errore

<b>Codice e Nome Completo</b>	TMIE – Tempo medio di risoluzione di un errore
<b>Categoria</b>	Indicatore di servizio
<b>Descrizione</b>	Fornisce il tempo medio di risoluzione di un errore.
<b>Metodo</b>	Le voci che compongono l'indicatore sono misurate grazie al software TSD.
<b>Formula</b>	$\frac{\text{Durata totale di risoluzione di Errore}}{\text{Errori risolti}}$
<b>Interpretazione</b>	L'indicatore può assumere valori maggiori di zero. Valori bassi dell'indicatore sono da preferire, perché vuol dire che gli errori sono stati risolti in breve tempo.
<b>Target</b>	1.5 h/per errore
<b>Scala di misura</b>	Scala di Rapporto
<b>Unità di misura</b>	Minuti/errore
<b>Fonte</b>	Software TSD

### PCT- Produttività dei Casi Di Test

<b>Codice e Nome Completo</b>	PCT- Produttività dei Casi Di Test
<b>Categoria</b>	Indicatore di Efficienza
<b>Descrizione</b>	Indica mediamente quanti test bisogna effettuare prima di individuare un errore.

<b>Metodo</b>	Le voci che compongono l'indicatore sono misurate grazie al software TSD.
<b>Formula</b>	$\frac{\text{Errori individuati prima della consegna finale del software}}{\text{Numero dei casi di test effettuati}}$
<b>Interpretazione</b>	L'indicatore può assumere valori maggiori di zero. Valori bassi dell'indicatore sono da preferire, perché vuol dire che per individuare errori nel software sono necessari pochi casi di test.
<b>Target</b>	0.2 errori/caso di test
<b>Scala di misura</b>	Scala di Rapporto
<b>Unità di misura</b>	Errore /Casi di test
<b>Fonte</b>	Software TSD

## 7.4 – Misurazione e Valutazione dei degli indicatori

Gli indicatori che andremo a misurare valuteranno la sola fase di test inerente gli adeguamenti effettuati per l'anno campagna 2018 al processo di istruttoria delle pratiche con premi pascolo. Per ulteriori approfondimenti sul progetto del caso di studio e sul software 'PUA', si rimanda al capitolo 6. La misura degli indicatori è stata effettuata il 16 febbraio 2019, data in cui si è concluso il processo di istruttoria e data dopo la quale le anomalie che sono state rilevate non possono essere più risolte.

Nella seguente tabella è possibile, oltre alle misure dei singoli indicatori, visualizzare i valori target e di quanto l'indicatore si discosta da esso.

Nel caso in cui l'indicatore deve essere maggiore di un certo valore (Target) lo scostamento è calcolato come la differenza tra il valore assunto dall'indicatore (Misura) ed il valore Target; altrimenti se l'indicatore deve essere inferiore del valore Target lo scostamento è misurato come la differenza del valore Target e la misura. Calcolato in questo modo se lo scostamento assume un valore positivo, vuol dire che la misura ha rispettato il valore target, altrimenti no.

<b>Indicatore</b>	<b>Misura</b>	<b>Target</b>	<b>Scostamento dal valore Target</b>
PEPTA	0.71	$\geq 0.8$	-0.09
PERPR	1	$\geq 0.9$	+0.1
PEP	0.16	$\leq 0.15$	-0.01
PENR	0	$\leq 0.1$	+0.1
TMIE	4.14	$\leq 3$	-1.14
TMRE	0.6	$\leq 1.5$	+0.9
PCT	0.14	$\geq 0.2$	-0.06

*Tabella 16 – Risultati della misurazione degli Indicatori*

Dalla tabella si evince che gli indicatori PER, PENR e TMRE rispettano ampiamente i valori target, mentre gli indicatori PEPTA, PEP, TMIE e PCT hanno mostrato delle criticità.

L'indicatore PEP, nonostante non abbia raggiunto l'obiettivo prestabilito, non desta preoccupazioni perché supera di poco il limite fissato.

Gli indicatori PEPTA, TMIE e PCT, invece, si distaccano molto dai valori target e la causa unica può essere ricercata in casi di test poco efficaci, dovuti ad una scarsa conoscenza del prodotto e del sistema. Dagli indicatori, in esame, si evince infatti che:

- Il software consegnato al cliente per la fase dei test di accettazione contiene un numero di errori elevato; il cliente riesce infatti ad individuare ulteriori 6 errori rispetto ai 15 individuati dal tester;
- Il tempo medio di individuazione di un errore è di circa un'ora e venti minuti superiore al valore limite;
- Un errore viene individuato mediamente dopo 7 casi di test, rispetto ai 5 richiesti dal valore obiettivo.

## 7.5 – Piano di intervento

L'autore dell'elaborato consiglia un piano di intervento atto a modificare l'organizzazione della fase di test. Nel paragrafo 7.1 è stata analizzata la fase di test del progetto PUA ed è stato detto che la fase di test inizia in parallelo con la fase di sviluppo dopo la fase di progettazione e che è composta da tre passi principali:

- 1° Definizione dei casi di test;
- 2° Esecuzione dei test da una risorsa interna (Tester);
- 3° Esecuzione dei test di accettazione;

L'autore suggerisce di aggiungere un ulteriore passo prima della definizione dei casi di test, 'Raccolta dei requisiti da Testare', che consiste nel coinvolgere anche il tester nella raccolta dei nuovi requisiti dal cliente. In questo modo il tester potrà:

1. Avere una conoscenza maggiore del processo che dovrà testare;
2. Raccogliere personalmente i nuovi requisiti;
3. Confrontarsi con l'analista, azione che può avere effetti positivi sia sulla fase di analisi che sulla fase di test;

L'autore suggerisce, inoltre, una partecipazione attiva del cliente durante la definizione dei casi di Test. In particolare durante e dopo la definizione dei piani di test suggerisce un continuo e duraturo confronto tra il tester ed il cliente, in modo tale da poter individuare e testare il maggior numero di casi d'uso differenti.

Nella figura seguente è mostrato come si evolverebbe la fase di Test con il Piano di intervento appena proposto.

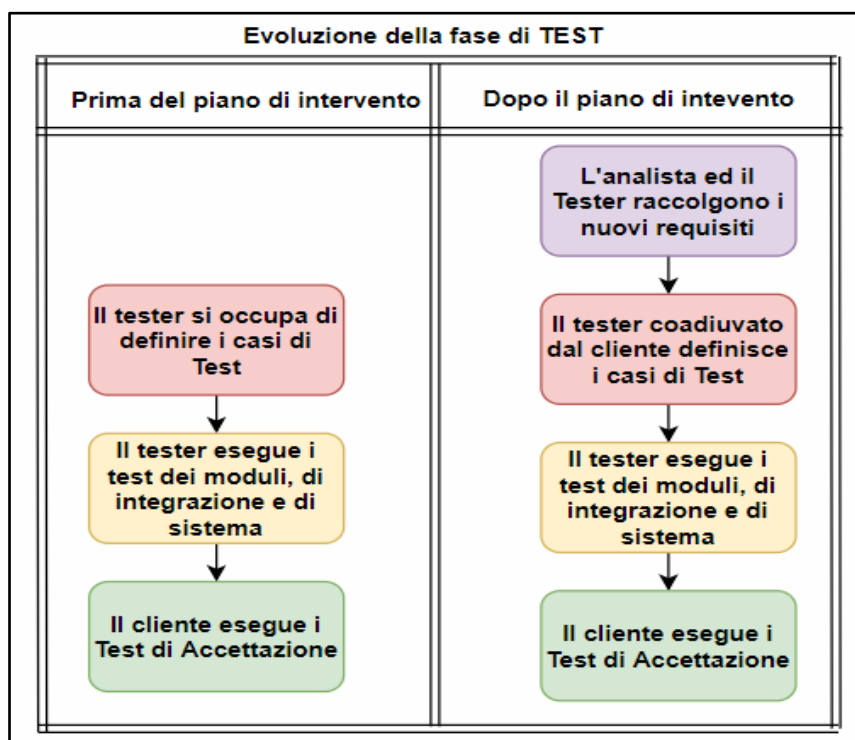


Figura 17 – Proposta di Evoluzione della fase di test

Gli obiettivi del piano di intervento, oltre a migliorare le misure degli indicatori che non hanno rispettato il valore Target, sono:

1. Ridurre il numero di dei casi di test: attraverso una migliore conoscenza del processo e dei requisiti e coadiuvato dal cliente, il tester sarà in grado di costruire casi di test più efficaci, evitando casi di test superflui o ripetitivi;
2. Ridurre la durata totale della fase di Test con una diminuzione anche dei costi;
3. Favorire l'individuazione precoce di errori nel software;
4. Ridurre gli errori individuati dal cliente durante i Test di Accettazione;
5. Ridurre gli errori individuati dopo il rilascio.

## **Conclusioni**

L'elaborato si è prefissato l'obiettivo di valutare le prestazioni della fase di testing per lo sviluppo di nuovi requisiti per un applicativo software impiegato dalla pubblica amministrazione (PA) utile per la gestione del fondo FEAGA (Fondo europeo agricolo di garanzia) in ambito di Richiesta di Finanziamento Unico (RFU).

L'analisi condotta nel capitolo finale della tesi ha rivelato alcune criticità della fase di Test. In particolare, gli indicatori hanno evidenziato una scarsa efficacia dei casi di Test interni la cui causa è da ricercare in una conoscenza limitata del processo esaminato. Nel capitolo finale è presentato un piano di intervento concreto e specifico per il progetto atto a modificare la fase di Test con l'obiettivo di correggere le misure degli indicatori che presentano anomalie.

# Bibliografia

1. (s.d.). Tratto da <http://www.opentextbooks.org.hk/zh-hant/ditatopic/26017>
2. (s.d.). Tratto da <https://www.managementstudyguide.com/what-is-public-administration.htm>
3. (s.d.). Tratto da <https://www.emotori.com/conosci-la-differenza-tra-software-orizzontale-e-verticale/>
4. (s.d.). Tratto da <https://www.techopedia.com/definition/4356/software>
5. (s.d.). Tratto da <https://en.wikipedia.org/wiki/Software>
6. (s.d.). Tratto da <https://www.pmgloss.com/about/>
7. (s.d.). Tratto da <https://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc>
8. (s.d.). Tratto da <https://standards.ieee.org/findstds/standard/610.12-1990.html>
9. (s.d.). Tratto da <https://www.seguetech.com/the-four-levels-of-software-testing/>
10. (s.d.). Tratto da [http://sse.tno.nl/IEEE\\_STDS/SESC/1517-1999.pdf](http://sse.tno.nl/IEEE_STDS/SESC/1517-1999.pdf)
11. (s.d.). Tratto da <http://ecomputernotes.com/software-engineering/types-of-software-maintenance>
12. (s.d.). Tratto da <https://reqtest.com/requirements-blog/understanding-the-difference-between-functional-and-non-functional-requirements/>;
13. (s.d.). Tratto da <https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Web/Requirements/FeasibilityStudy.html>;
14. (s.d.). Tratto da <https://www.oecd.org/dac/evaluation/2754804.pdf>
15. (s.d.). Tratto da [https://en.wikiversity.org/wiki/Information\\_Systems/Hardware](https://en.wikiversity.org/wiki/Information_Systems/Hardware)
16. (s.d.). Tratto da <https://www.britannica.com/topic/information-system>
17. (s.d.). Tratto da <http://arianna.cr.piemonte.it/regolaf0/dettaglioRegolamento.do?urnRegolamento=urn:nir:regione.piemonte:regolamento:2011-09-20;8@2018-08-23&tornaIndietro=true>
18. (s.d.). Tratto da <https://safebytes.com/7-types-software-errors-every-tester-know/>
19. (s.d.). Tratto da [https://www.researchgate.net/publication/46280097\\_Software\\_Testing\\_-\\_Goals\\_Principles\\_and\\_Limitations](https://www.researchgate.net/publication/46280097_Software_Testing_-_Goals_Principles_and_Limitations)
20. (s.d.). Tratto da [http://thklein.com/en\\_US/cost-of-defect/](http://thklein.com/en_US/cost-of-defect/)
21. (s.d.). Tratto da [https://it.wikipedia.org/wiki/Politica\\_agricola\\_comune](https://it.wikipedia.org/wiki/Politica_agricola_comune)
22. (s.d.). Tratto da [https://en.wikipedia.org/wiki/Common\\_Agricultural\\_Policy](https://en.wikipedia.org/wiki/Common_Agricultural_Policy)
23. (s.d.). Tratto da [http://europedirect.055055.it/sites/europedirect.055055.it/files/documenti/la\\_pac\\_spiegata.pdf](http://europedirect.055055.it/sites/europedirect.055055.it/files/documenti/la_pac_spiegata.pdf)
24. (s.d.). Tratto da <https://www.terrainnova.it/cose-lo-sviluppo-rurale/>
25. (s.d.). Tratto da <https://statistica.regione.emilia-romagna.it/metadati/glossario/u/uba-n>



26. (s.d.). Tratto da <https://www.toppr.com/guides/accountancy/application-of-computers-in-accounting/meaning-and-elements-of-computer-system/>
27. (s.d.). Tratto da <http://www.isaonline.it/mag/UE-Funzionamento.html>
28. B. P. Lientz, E. B. (s.d.). *Software Maintenance Management*.
29. Carlo Ghezzi, D. M. (s.d.). *Ingegneria del software: fondamenti e principi*. Pearson.
30. Carlo Ghezzi, M. J. (s.d.). *Ingegneria del software. Fondamenti e principi*.
31. Damiani, E. (s.d.). *Ingegneria del software con UML e Unified Process*. McGraw-Hill.
32. Fiorenzo Franceschini, M. G. (2007). *Indicatori e misure di prestazione per la gestione dei processi*. Il Sole 24 Ore .
33. Fiorenzo Franceschini, M. G. (2007). *Management by Measurement: Designing Key Indicators and Performance Measurement Systems*. SPRINGER.
34. Franceschini, F. (2001). *Dai prodotti ai servizi. Le nuove frontiere per la misura della qualità* . UTET Università.
35. Fritz Morstein Marx, J. W. (s.d.). *Elements of Public Administration*.
36. Fulvio Corno, M. T. (s.d.). Tratto da <https://softeng.polito.it/courses/02CIX/Dispense/cap10-standalone.pdf>
37. G. Bracchi, C. F. (s.d.). *Sistemi informativi d'impresa*.
38. <https://bus206.pressbooks.com/chapter/chapter-1/>. (s.d.).
39. [https://www.cs.csustan.edu/~lamie/cps603/glossary\\_of\\_terms.htm](https://www.cs.csustan.edu/~lamie/cps603/glossary_of_terms.htm). (s.d.).
40. *IEEE Standard Glossary of Software Engineering Terminology*. (s.d.). 2010.
41. Jane Price Laudon, K. C. (2014). *Management Information Systems: Managing the Digital Firm* . PEARSON.
42. Jeong, C. H. (s.d.). *Fundamental of development administration*.
43. Kenneth C. Laudon, C. G. (2012). *Management Information Systems*. Prentice-Hall.
44. L., F. (1982). *Handbook of measurement science, Theoretical Fundamentals*.
45. Massot, A. (s.d.). Tratto da <http://www.europarl.europa.eu/factsheets/it/sheet/106/il-finanziamento-della-pac>
46. Massot, A. (s.d.). Tratto da <http://www.europarl.europa.eu/factsheets/it/sheet/107/gli-strumenti-della-pac-e-le-loro-riforme>
47. Massot, A. (s.d.). Tratto da <http://www.europarl.europa.eu/factsheets/it/sheet/108/il-primopilastro-della-pac-i-l-organizzazione-comune-dei-mercati-ocm-dei-prodot>
48. Massot, A. (s.d.). Tratto da <http://www.europarl.europa.eu/factsheets/it/sheet/109/primopilastro-della-politica-agricola-comune-pac-ii-pagamenti-diretti-agli-agri>
49. Massot, A. (s.d.). Tratto da <http://www.europarl.europa.eu/factsheets/it/sheet/110/il-secondo-pilastro-della-pac-la-politica-di-sviluppo-rurale>
50. Massot, A. (s.d.). Tratto da <http://www.arpea.piemonte.it/site/interventi-finanziati/regime-di-pagamento-unico/681-l-aiuto-disaccoppiato-o-regime-di-pagamento-unico>

51. Massot, A. (2018). Tratto da <http://www.europarl.europa.eu/factsheets/it/sheet/103/la-politica-agricola-comune-pac-e-il-trattato>
52. Nigro, F. A. (s.d.). *Modern Public Administration*.
53. Parmenter, D. (2015). *Key Performance Indicators Developing, Implementing, and Using Winning KPIs*. John Wiley & Sons Inc;.
54. Paul R. Gamble, J. B. (s.d.). *Knowledge Management: A State of the Art Guide*. Kogan Page Business Books .
55. R. Kelly Rainer, C. G. (s.d.). *Introduction to Information Systems*. Wiley.
56. Reema Khurana, V. V. (2009). Business process modeling with information integrity. *Business Process Management Journal*.
57. S.M.K Quadri, S. U. (2010). Software Testing – Goals, Principles, and Limitations. *International Journal of Computer Applications* .
58. Schneider, J. V.-C. (2010). *Information Systems Today - Managing in the Digital World, fourth edition*. . Prentice-Hall.
59. Simon, H. A. (s.d.). *The New Science of Management Decision*. Prentice Hall.
60. Sommerville, I. (s.d.). *Software Engineering*. Pearson .
61. Stair Ralph, G. R. (2006). *Fundamentals of Information Systems*. Cengage Learning.
62. Stephens, R. (s.d.). *Beginning Software Engineering*.
63. *Systems and software engineering - Vocabulary, ISO/IEC/IEEE std 24765*. (2010).
64. White, L. D. (s.d.). *Introduction to the Study of Public Administration*.
65. *Wikipedia*. (s.d.). Tratto da [https://it.wikipedia.org/wiki/Modello\\_a\\_spirale](https://it.wikipedia.org/wiki/Modello_a_spirale)
66. *Wikipedia*. (s.d.). [http://en.wikipedia.org/wiki/Information\\_systems\\_\(discipline\)](http://en.wikipedia.org/wiki/Information_systems_(discipline)).
67. Wilson, W. (s.d.). *The Study of Administration*.