

# POLITECNICO DI TORINO

Corso di Laurea Magistrale

in Ingegneria Gestionale – Percorso Informatica

Tesi di Laurea Magistrale

*Analisi e implementazione dell'Hybrid Agile Framework*

*nella gestione dei progetti*



Relatore:

Prof.ssa Giulia Bruno

Candidato:

Gianluca Ficarra

Anno Accademico 2020/2021

*Un guerriero non può abbassare la testa, altrimenti perde di vista l'orizzonte dei suoi sogni.*

Paulo Coelho

## Indice

Introduzione	4
1. Project Management	6
1.1 Origini e caratteristiche del Project Management	6
1.2 Definizione del progetto	10
1.3 Ciclo di vita di un progetto	12
2. Metodologia Tradizionale Waterfall	23
2.1 Origini e caratteristiche del modello Waterfall	23
2.2 Fasi della metodologia Waterfall	25
2.3 Contesti applicativi	27
2.4 Vantaggi e svantaggi del metodo	28
3. Metodologia Agile	32
3.1 Origini del Manifesto e del metodo Agile	32
3.2 Struttura del metodo	37
3.2.1 Fasi	38
3.2.2 Proprietà	41
3.3 Contesti applicativi	42
3.4 Vantaggi e svantaggi del metodo	43
3.5 Metodo SCRUM	46
3.5.1 Ruoli	48
3.5.2 Eventi	49
3.5.3 Artefatti	51
3.5.4 Vantaggi e svantaggi di Scrum	53
4. Approccio “Ibrido”	55
4.1 Waterfall vs Agile	55
4.2 Ibrido	59
5. Caso di studio aziendale – Altea UP	61
5.1 Altea UP: panoramica dell’azienda	61
5.2 Metodologia Waterfall di Altea UP	65
5.2.1 Organizzazione progettuale	66
5.2.2 Ciclo di vita del progetto secondo il Waterfall di Altea UP	67
5.3 Hybrid Agile Framework Altea UP	73
5.3.1 Caratteristiche dell’Hybrid Agile Framework	76
5.3.2 Ciclo di vita del progetto secondo l’Hybrid di Altea UP	80
5.3.3 Ruoli nel progetto Hybrid	85
5.3.4 Principi di governo della dimensione temporale ed economica	87
5.3.5 Componenti della metodologia tradizionale	88
5.3.6 Tipologie di test	88
5.3.7 Reporting real-time	89
Bibliografia e Sitografia	92
Ringraziamenti	94

## Introduzione

Al giorno d'oggi le imprese stanno attraversando un periodo storico contraddistinto da continui cambiamenti, in un contesto estremamente dinamico e incerto. Faticano a trovare dei vantaggi competitivi sostenibili nel lungo periodo, fatto che ha condotto all'evoluzione delle metodologie di Project Management: da metodologie lineari ad approcci iterativi e ibridi, inizialmente pensati per la fase di programmazione ma poi applicati all'intera organizzazione aziendale.

Il presente lavoro di tesi è frutto di una stretta collaborazione con l'azienda Altea Up, società di consulenza SAP Golden Partner presso la quale ho svolto le attività di tirocinio curricolare. Altea Up mi ha proposto di indagare le metodologie di gestione dei progetti utilizzate, dando maggiore rilievo all'Hybrid Agile Framework da loro implementato. Quest'ultimo è un approccio versatile che può essere adattato alle mutevoli e sempre diverse esigenze di un progetto. Come richiesto da Altea Up, mi sono occupato di studiare e realizzare una documentazione relativa all'approccio ibrido utilizzato, alla luce della mia esperienza lavorativa in un progetto in ambito farmaceutico in cui è attualmente in uso, confrontandolo anche con analoghe applicazioni in altri progetti. Il risultato è una sorta di manuale di riferimento di tale Framework per poterlo replicare con maggiore uniformità in progetti futuri.

L'elaborato è suddiviso in cinque capitoli. Nella prima parte è trattata la letteratura relativa al Project Management e allo sviluppo di un progetto; nei capitoli successivi si analizzano nel dettaglio le tre metodologie Waterfall, Agile e Ibrido, sia singolarmente sia poste a confronto; nell'ultimo capitolo, infine, sono presentati dei casi di studio aziendali, frutto dell'analisi dell'applicazione di queste metodologie da parte di Altea Up nei suoi progetti. In settori tradizionali come quello chimico e delle costruzioni è spesso adottato un metodo Waterfall, mentre in progetti complessi, in contesti più dinamici o soggetti a specifiche normative come quello farmaceutico, si predilige l'Hybrid Agile Framework. Entrambi sono oggetto dei casi di studio di questa tesi, ma il secondo risulta essere attualmente il più utilizzato e fruttuoso per l'azienda. Al contrario, per la metodologia Pure Agile non è proposto un caso applicativo reale, dal momento che questa non è adottata da Altea Up a

causa di caratteristiche specifiche della metodologia stessa (ridotta documentazione e una fase di pianificazione e analisi meno approfondita).

## Capitolo primo

### Project Management

#### 1.1 Origini e caratteristiche del Project Management

Tra il XX e il XXI secolo, Frederick Winslow Taylor, ingegnere meccanico americano, analizzò le tecniche di gestione aziendale, elaborando una teoria basata sull'esistenza di una via migliore (*one best way*) che i lavoratori avrebbero potuto percorrere per ottimizzare i tempi di lavoro. Durante gli anni della prima guerra mondiale, Henry Laurance Gantt, un associato di Taylor, studiò un modo per ottimizzare la sequenza di attività richieste per la costruzione di una flotta, realizzando nel 1917 l'omonimo Diagramma di Gantt, un grafico utilizzato come supporto alla pianificazione dei progetti, tuttora alla base delle attività di pianificazione. La struttura del diagramma cartesiano<sup>1</sup> prevede nelle ascisse una scala temporale che si estende dall'inizio alla fine del progetto, mentre nelle ordinate sono riportate le attività da svolgere, garantendo di poter rappresentare, pianificare e controllare l'avanzamento delle attività (*Figura 1*).

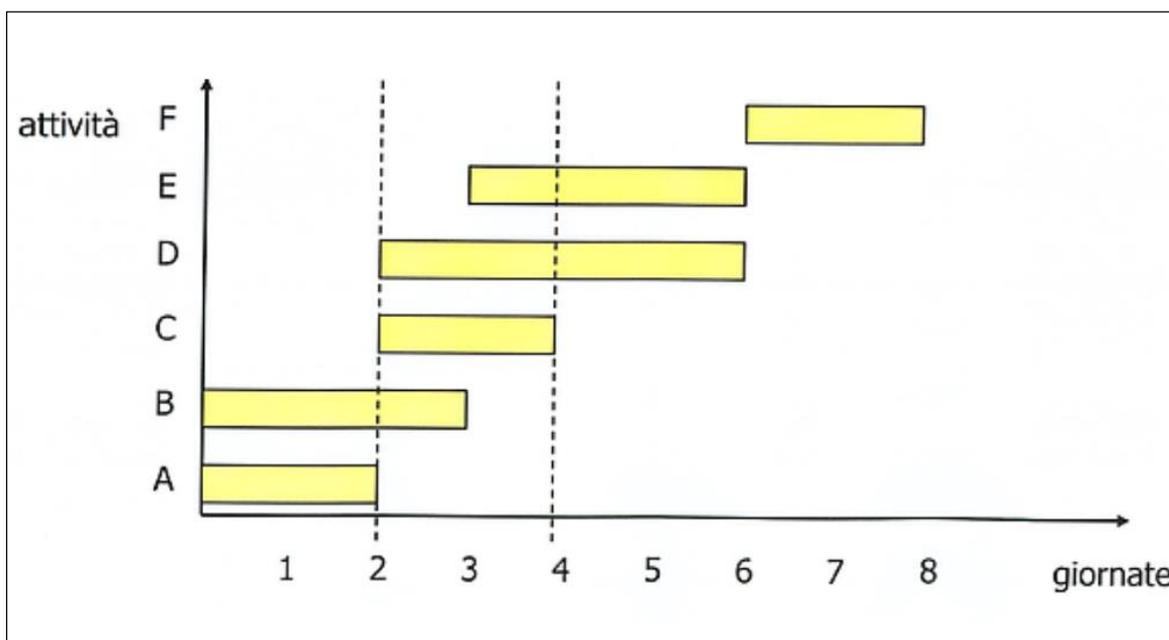


Figura 1. Esempio di Diagramma di Gantt.

<sup>1</sup> Diagramma di PERT e di GANTT. Strumenti per la gestione/controllo dei progetti, <https://www.unieda.it/modulo-e4-improving>.

Dopo la seconda guerra mondiale, si avvertì sempre di più l'esigenza di realizzare un piano di ricostruzione generale e di progredire nella gestione dei progetti; pertanto, quest'ultima divenne una tematica sempre più rilevante e in continua evoluzione, con un conseguente sviluppo anche di nuove metodologie come i diagrammi di PERT (*Program Evaluation and Review Technique*)<sup>2</sup> e del metodo del CPM (*Critical Path Method*) (Figura 2).

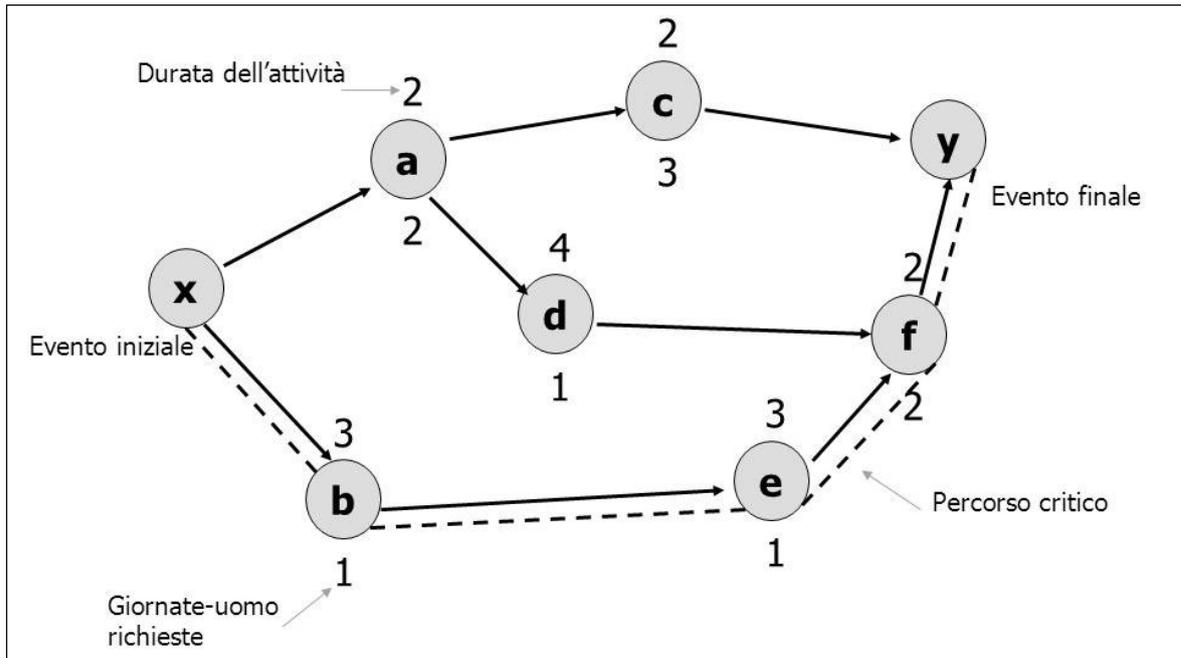


Figura 2. Esempio di diagramma PERT – CPM.

A partire dagli anni Sessanta, nella cultura aziendale iniziò ad affermarsi la disciplina del Project Management, per merito di imprese che suddivisero i loro lavori per progetti, riscontrando così decisivi miglioramenti in termini di efficienza. Si diffusero sempre più massicciamente sistemi gestionali e di pianificazione, in quanto era ormai oggettivo il vantaggio derivante dall'osservazione dell'avanzamento temporale ed economico di un progetto. Dagli anni Settanta in poi, tale disciplina fu considerata ormai consolidata nell'edilizia, e dagli anni Ottanta anche in altri contesti, grazie all'avvento di internet e all'evoluzione della componentistica hardware e software. Dato il successo ottenuto, furono fondate molte associazioni internazionali, tra cui le principali sono IPMA (*International Project Management Association*) e PMI (*Project Management Institute*). Questi enti si

<sup>2</sup> Vedi nota 1.

occupano della gestione e del rilascio delle certificazioni internazionali e della diffusione della cultura del Project Management.

In generale, con l'espressione Project Management si intende la gestione sistematica di un'impresa complessa, unica e dalla durata limitata, rivolta al raggiungimento di un obiettivo. Quest'ultimo è predeterminato attraverso un processo continuo di pianificazione e controllo delle risorse, con vincoli di costi-tempi-qualità. Ne esistono quattro modelli:

- AD HOC

Modelli "fai da te" utilizzati nelle piccole/medie imprese che hanno da sempre lavorato per processi e che hanno dovuto iniziare a lavorare per progetti. È caratterizzato da: regole locali e approccio informale; una formazione tecnica e non manageriale del PM; scarso metodo di monitoraggio; disallineamenti dovuti a errori di persone; elevata conoscenza del contesto applicativo.

- BUROCRATICO

Modello utilizzato in ambito pubblico, un contesto piuttosto stabile in cui sono richieste una particolare trasparenza degli atti di utilizzo delle risorse e una correttezza formale.

- NORMATIVO

Approccio più diffuso, basato su standard internazionali che forniscono le regole. È caratterizzato da un'impostazione razionale, con ampio uso di norme e manuali, scarsa reattività ai cambiamenti di contesto e gestione di un'alta quantità di dati. Quella del PM è considerata una vera e propria professione.

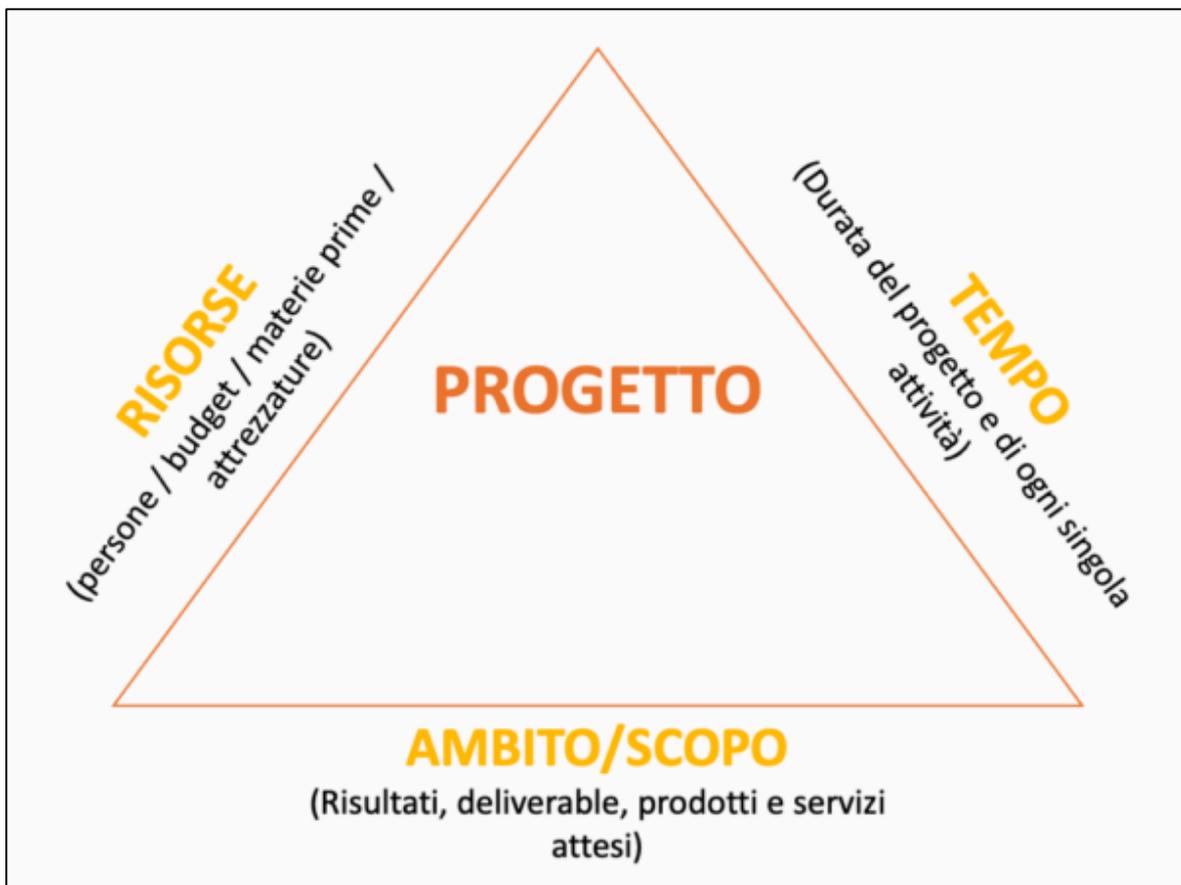
- CREATIVO

Modello più adatto a contesti dinamici poiché si adegua alla complessità attraverso una predisposizione all'auto-organizzazione e un'impostazione flessibile sui deliverable e sui rapporti con il cliente.

Con il passare degli anni, il ruolo del Project Manager ha avuto una sostanziale evoluzione, assumendo sempre più importanza fino a diventare una figura cruciale per il successo di un progetto. Negli ultimi tempi è emersa l'insufficienza delle sole capacità tecniche per gestire

progetti dinamici in contesti innovativi e competitivi: oggi giorno risultano fondamentali anche le competenze manageriali. Quindi, il Project Manager ha il compito di: definire la visione globale, gli obiettivi, le responsabilità e i risultati del progetto; pianificare e organizzare gli step necessari; controllare costantemente se gli obiettivi sono raggiunti nei tempi e nei costi previsti; guidare e ispirare il team; concludere il progetto in modo efficiente ed efficace.

Dunque, quotidianamente al Project Manager è richiesto di prendere decisioni critiche che riguardano il “triplice vincolo”<sup>3</sup> del Project Management secondo cui tempi, costi e qualità si condizionano e vincolano reciprocamente: infatti, ogni variazione di un lato del triangolo ha ripercussioni sugli altri due, andando a determinare diverse configurazioni qualitative della soluzione finale (*Figura 3*).



*Figura 3.* “Triplice vincolo” del Project Management.

<sup>3</sup> Cristian Boin, *Project management – 7 consigli utili a tutti*, in “Cristian Boin – Manager in affitto”, <https://cristianboin.com/consigli-di-project-management/>.

- **Tempo**  
Ogni progetto ha un ciclo di vita specifico e predefinito, così la riduzione delle tempistiche di un progetto comporterebbe un aumento del budget per avere a disposizione un maggior numero di risorse, a discapito della qualità, con l'unico scopo di consegnare il progetto entro le nuove scadenze.
- **Risorse**  
Il budget necessario per ultimare un progetto viene sempre definito prima dell'inizio delle attività; quindi ridurlo in corso d'opera comporterebbe un'inevitabile riduzione della qualità dell'output, oltre che un'estensione dei tempi di consegna.
- **Ambito/Scopo**  
Lo scopo di un progetto fa riferimento ai risultati attesi con un livello di qualità che rispetti i requisiti richiesti. Ricercare un maggiore livello di qualità per un progetto già avviato comporta dover aumentare anche i costi e le tempistiche legate al ciclo di produzione.

Secondo questa prospettiva, il Project Manager bilancia il triplice vincolo al fine di soddisfare le esigenze del cliente, il quale deve essere consapevole dei rischi che derivano dal modificare una delle tre componenti. Nella realtà, almeno un lato del triangolo è fisso, di conseguenza sarà possibile modificare solo i restanti per rispettare i requisiti del cliente e gli obiettivi del progetto.

### *1.2 Definizione del progetto*

Il PMI definisce un progetto come la combinazione di risorse umane e non, riunite in un'organizzazione temporanea per raggiungere un obiettivo definito con risorse limitate.

Un progetto è perciò considerato come una combinazione di risorse tra le quali incentivare la collaborazione per massimizzare la soluzione finale, che deve essere definita sin dal

principio. È importante anche sottolineare che la durata di un progetto è limitata, così come le risorse a esso allocate, quindi risulta centrale il concetto della transitorietà.

Dunque, le principali caratteristiche di un progetto possono essere riassunte in:

- **Transitorietà**  
Ogni progetto ha una durata limitata nel tempo.
- **Unicità del Deliverable**  
Gli output di un progetto sono unici perché frutto della combinazione di molteplici fattori caratteristici di ogni caso.
- **Elaborazione progressiva**  
Un progetto viene sviluppato tramite fasi successive o incrementali.

Nel processo di definizione degli obiettivi di un progetto nella maggior parte dei casi è adottato il metodo SMART<sup>4</sup> per verificare se questi siano validi (*Figura 4*):



*Figura 4. Metodo SMART.*

- **Specific:** gli obiettivi devono essere ben definiti, tangibili e comprensibili;
- **Measurable:** deve essere possibile la misurazione di un obiettivo, deve cioè essere quantificabile numericamente;
- **Achievable:** l'obiettivo deve poter essere raggiungibile in base alle risorse e al contesto in cui è sviluppato; invece, dopo averlo tragguardato è richiesto che sia stabile e affidabile nel tempo;

---

<sup>4</sup> Virginia Fabris, *Come impostare obiettivi SMART? Il metodo per raggiungere la vetta*, in "Appvizer", <https://www.appvizer.it/rivista/organizzazione/project-management/obiettivi-smart>.

- **Realistic:** è necessario individuare i margini di realizzazione del progetto, analizzando attentamente il rapporto costi/benefici per capire se gli obiettivi siano realistici e di interesse. Inoltre, deve essere un progetto sfidante ma allo stesso tempo alla portata del team;
- **Time based:** i progetti devono essere pianificati e terminati entro una certa data; sono inoltre soggetti a un costante monitoraggio nel loro stato di avanzamento.

### *1.3 Ciclo di vita di un progetto*

Nel ciclo di vita di un progetto è possibile distinguere cinque fasi che costituiscono l'intero periodo di svolgimento delle attività, dall'inizio alla fine.

#### 1) AVVIO

L'avvio di un progetto nasce con un'offerta da parte del contraente verso il cliente, a partire da un suo bisogno o problema, all'interno della quale vengono già definiti alcuni elementi strategici di base. Il Proposal Manager è il responsabile della fase di offerta, considerata anche come la fase di ideazione e fattibilità; ogni obiettivo deve essere ritenuto valido dal Test Smart.

Questa fase comprende:

- Definizione del progetto e dei suoi requisiti;
- Formulazione di un'offerta da proporre al cliente da parte del contraente: se ne occupa il Proposal Manager, con il supporto del Project Manager, per ridurre il gap tra la fase di offerta e quella esecutiva;
- Perfezionamento del contratto: un momento in cui tutti gli attori coinvolti devono comprendere il contratto definendo con cura la sezione tecnica e quella economico-normativa. La firma indica che le parti hanno già concordato cosa fare, con quali tempistiche e budget;
- Gestione delle varianti in corso d'opera richieste dal committente: è importante per la commessa a lui associata perché aggiunge valore, ma anche margine economico;

L'avvio si basa sul kick-off meeting iniziale, una riunione in cui il capo progetto e i funzionali si riuniscono per scambiarsi informazioni utili per lo sviluppo. L'obiettivo è quello di informare gli enti preposti alla realizzazione della commessa e di fornire informazioni guida sulle caratteristiche del nuovo progetto. Gli elementi necessari a questa riunione sono l'invio della documentazione contrattuale richiesta per sviluppare le attività, le valutazioni economiche, l'individuazione di eventuali criticità e la stesura dei piani di lavoro specifici.

Tra i documenti di riferimento il principale è il Project Charter, che autorizza ufficialmente il progetto e che deve includere informazioni come i requisiti, il proposito, il PM assegnato, lo schedule delle principali milestone, l'influenza degli stakeholder, l'organizzazione funzionale e dei vincoli organizzativi e un budget riassuntivo.

## 2) PIANIFICAZIONE

Nella seconda fase avviene la pianificazione del progetto, dunque se ne prevede lo sviluppo per ottimizzare i tempi della realizzazione, l'impiego delle risorse e i costi da sostenere. Per tali ragioni, il planning viene inteso come un "modello della realtà futura" che corrisponde alla realtà se il contesto è stabile e c'è una disponibilità di informazioni.

Le attività di pianificazione consistono sostanzialmente in tre cose da fare:

CHE COSA: Identificare gli elementi fisici costituenti del progetto;

COME: Identificare i processi necessari all'esecuzione;

CHI: Assegnare le responsabilità esecutive.

Gli strumenti utilizzati nella pianificazione sono:

- WORK BREAKDOWN STRUCTURE

La WBS<sup>5</sup> è una rappresentazione analitica del progetto di tipo gerarchico, in cui questo è suddiviso in parti di lavoro con le attività distribuite su più livelli dal diverso grado di dettaglio. Possono essere utilizzate varie logiche di scomposizione: funzionale, spaziale, dei processi di lavoro, scomposizione fisica e degli obiettivi. Tale struttura gerarchica del progetto deve però comprendere tutti gli elementi necessari alla sua formazione e i principali compiti funzionali per realizzarli. Quindi, la WBS consente di stabilire una relazione significativa fra i diversi elementi d'informazione e di codifica, facilitando il lavoro di coordinamento, di monitoraggio e l'organizzazione delle responsabilità (Figura 5).

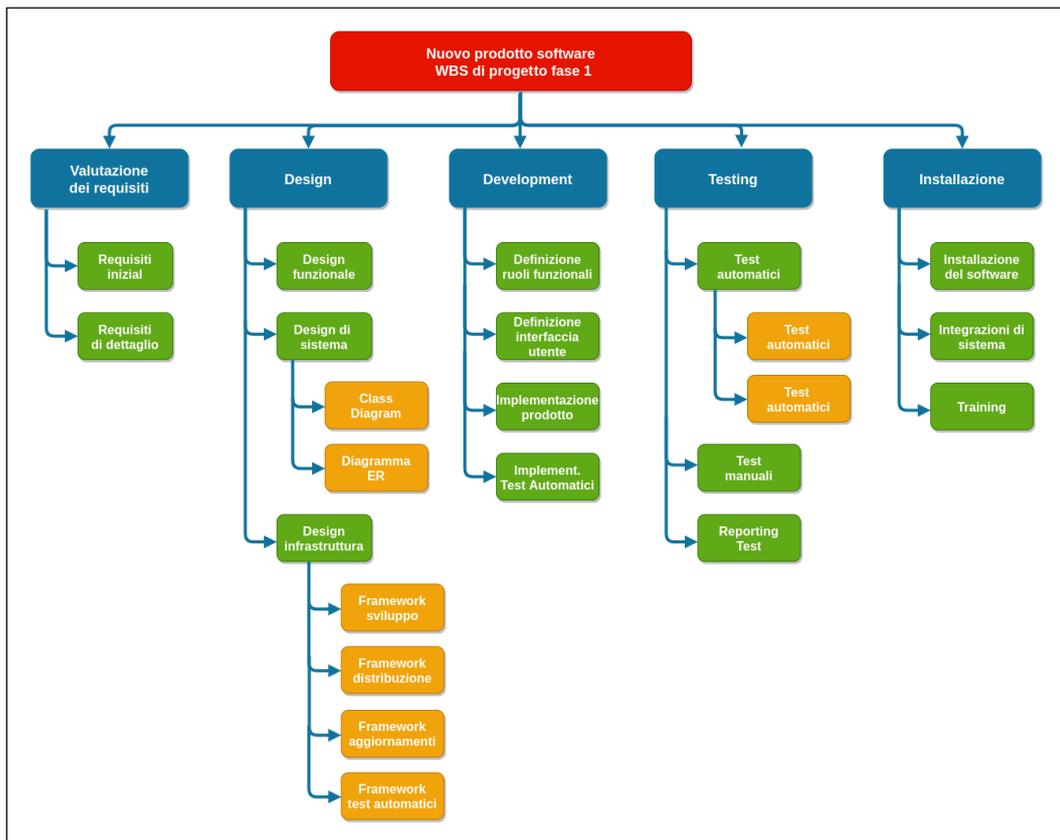
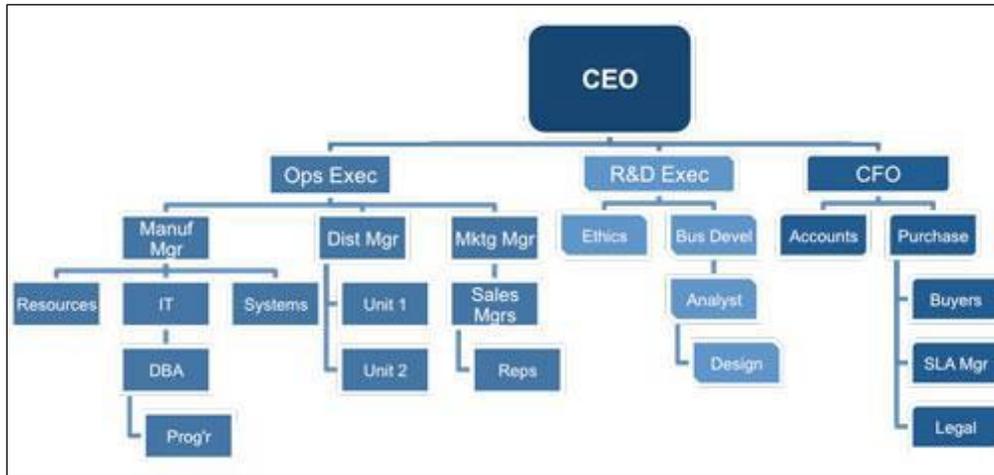


Figura 5. Esempio di WBS.

<sup>5</sup> Stefano M., *Un esempio di Work Breakdown Structure di progetto*, in “TeraNet PPM”, <https://www.tppm.it/esempio-wbs-di-progetto-software>.

- ORGANIZATION BREAKDOWN STRUCTURE

Dopo aver realizzato la WBS, è necessario collegare le singole attività all'esecutore, operazione possibile attraverso la OBS<sup>6</sup>. Si tratta di un organigramma di progetto in cui si associano le attività alle risorse; occorre perciò a identificare i partecipanti o le funzioni aziendali in uno specifico progetto (*Figura 6*).



*Figura 6.* Esempio di OBS.

- RESOURCE BREAKDOWN STRUCTURE

L'RBS<sup>7</sup> rappresenta, in una specifica struttura organizzativa, l'identificazione di tutte le risorse fisiche necessarie per la realizzazione di un progetto. Con il termine "risorse fisiche" fa riferimento a tutto ciò che è necessario per realizzare il progetto in termini di risorse umane, economiche e materiali (attrezzature e impianti) (*Figura 7*).

<sup>6</sup> Vito Lavecchia, *Che cos'è l'OBS (Organization Breakdown Structure) e la matrice delle responsabilità*, in "Informatica e ingegneria online", <https://vitolavecchia.altervista.org/che-cose-lobb-organization-breakdown-structure-e-la-matrice-delle-responsabilita/>.

<sup>7</sup> *Resource Breakdown Structure*, <http://resourcebreakdownstructure.blogspot.com/2013/06/resource-breakdown-structure.html>.

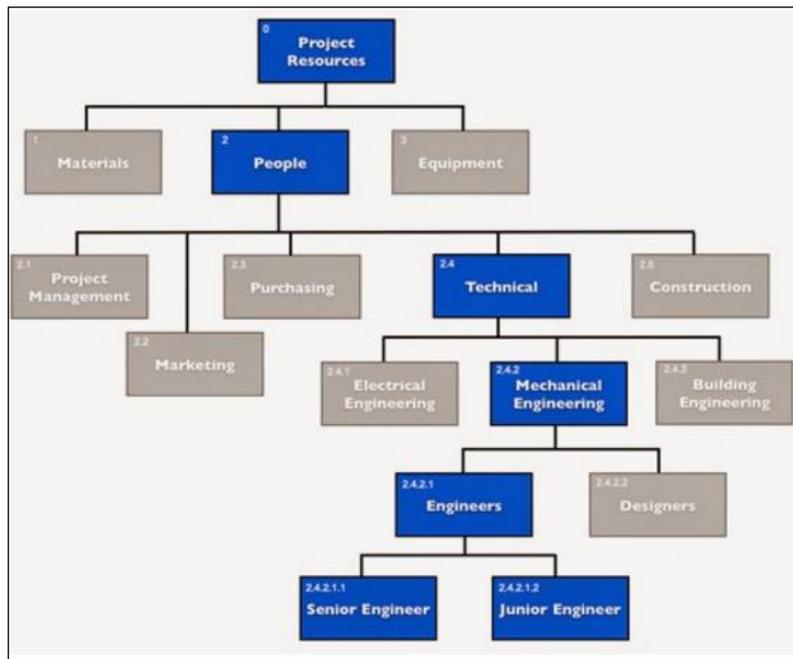


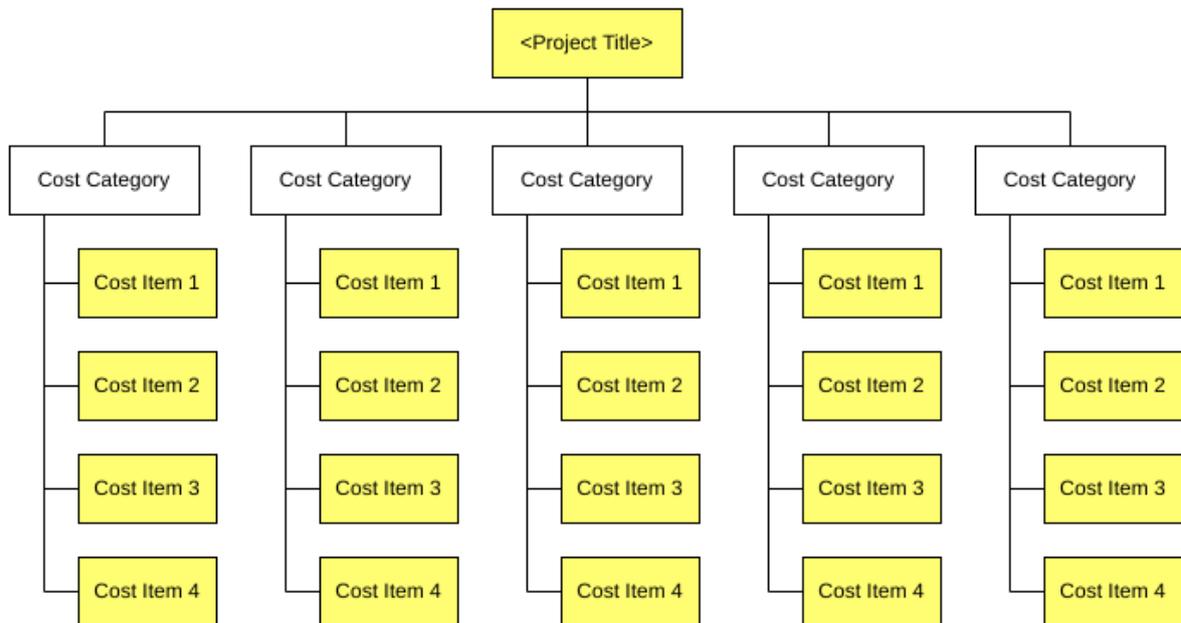
Figura 7. Esempio di RBS.

- COST BREAKDOWN STRUCTURE

La scomposizione del progetto si può estendere alla valorizzazione economica delle singole attività, noti i costi unitari e l'entità delle risorse assegnate. Nasce così questa nuova struttura ad albero relativa ai costi di commessa, le cui componenti si riferiscono a voci direttamente legate al progetto o a esso assimilabile. La CBS<sup>8</sup> è usata per definire il costo di un progetto assegnando ai Work Package i costi unitari (Figura 8). Tuttavia, in questa fase è possibile parlare di costi solo in termini di stime, distinguendo due approcci:

- 1) Bottom Up: si parte dal costo delle singole attività per ogni Work Package e si costruiscono per aggregazione i costi totali.
- 2) Top Down: si valuta il costo complessivo del progetto e lo si scompone attraverso iterazioni nelle varie componenti. Questo avviene tramite stime parametriche; spesso, una volta individuato il valore di riferimento, si deve restare all'interno di tale budget.

<sup>8</sup> Cost Breakdown Structure Template, in "Visual Paradigm Online", <https://online.visual-paradigm.com/de/diagrams/templates/work-breakdown-structure/cost-breakdown-structure-template/#>.



*Figura 8. Esempio di CBS.*

Individuate le voci componenti del progetto, dopo la pianificazione vi è la necessità di calendarizzare le attività che sono state individuate e temporalmente definite. Si stabiliscono delle logiche consequenziali fra le varie attività, ossia i vincoli di precedenza, andando a creare un network fra esse.

In una commessa sono richiesti tre documenti, con un diverso livello di dettaglio, al fine di mettere in relazione committente e fornitore:

- Overall Master Schedule

È il documento più generico definito in fase di offerta, incorporato nel contratto e verificato all'avvio. Il suo scopo è quello di fissare gli obiettivi del progetto, riportare le milestone contrattuali e fornire le linee guida del progetto. Inoltre, deve riportare almeno una serie di date come quelle di inizio/fine lavori, intermedie, di consegna, di durata delle attività, di avviamento, di collaudo, di fine garanzia e fatturazione. Quindi, l'OMS è un insieme di date e informazioni che fissano sostanzialmente i rapporti col committente, definito come "baseline contrattuale" perché è la base del contratto che non può essere modificata a meno di ripercussioni sul progetto.

- Project Schedule

Il Project Schedule è redatto dal Project Manager insieme ai funzionali durante l'avvio del progetto. Il suo scopo è quello di stabilire il programma, riportare gli eventi del progetto, definire priorità e sequenze delle attività fornendo le date per l'elaborazione dei programmi di dettaglio sviluppati dai singoli enti. Si tratta, dunque, di un insieme di informazioni che cambiano nel tempo. Questo rappresenta la "Timeline del progetto", un documento operativo di programmazione che serve all'impresa per comunicare, monitorare e verificare l'avanzamento dei lavori. Il requisito fondamentale è la dinamicità, che consente di effettuare modifiche al percorso prestabilito rispettando comunque le milestone previste.

- Detailed Schedule

Dal Project Schedule derivano i Detailed Schedule delle singole funzioni aziendali, così ogni soggetto ha la possibilità di creare la propria programmazione di dettaglio.

Le principali tecniche di Scheduling appartengono a tre famiglie: la prima è quella dell'Elenco delle attività; la seconda fa riferimento ai Diagrammi a barre tra cui spicca il celebre Diagramma di Gantt; la terza comprende varie Tecniche reticolari come PERT, CPM, PDM, CCM e CRD, le quali, attraverso nodi e archi, consentono di realizzare e rappresentare graficamente i collegamenti logici tra le attività al fine di creare un reticolo che possa definire il piano dei tempi e delle risorse del soggetto.

- Elenco delle attività

Questo elenco è il metodo più semplice per programmare un progetto; consiste in un elenco delle attività con le date di inizio, scadenza e revisione che permette di verificarne gli avanzamenti. Si utilizza quando si ha poca esperienza o quando l'interlocutore non ha familiarità con le altre tecniche di Scheduling.

- Diagramma di Gannt

Questo metodo è adatto alla programmazione di progetti di dimensioni ridotte o standard, attraverso la rappresentazione della sua evoluzione su una scala temporale. A partire dall'analisi della WBS, il diagramma consente di assegnare a ogni attività una stima della sua durata, andando a rappresentarla graficamente tramite rettangoli di lunghezza proporzionale alla loro durata. Tali diagrammi si basano su: scomposizione nelle operazioni principali corrispondenti alle attività delle WBS; assegnazione della durata stimata a ogni attività; rappresentazione delle azioni con figure rettangolari. La sequenza dei rettangolini deve seguire il reale sviluppo delle attività.

- PERT - *Program Evaluation and Review Technique*

Il PERT è un metodo statistico di determinazione dei tempi delle attività di un progetto che si basa su dei valori di stima. La tecnica è stata sviluppata per l'applicazione in contesti in cui è necessario valutare e controllare lo sviluppo di progetti in ambiti di incertezza, in quanto è uno strumento dinamico che garantisce un'ottima programmazione, comunicazione, controllo e informazione. La sua finalità è quella di ordinare le attività, determinare il minor tempo possibile per raggiungere l'obiettivo e individuare le operazioni critiche.

- CPM - *Critical Path Method*

Metodo attraverso il quale si migliora la pianificazione dei programmi di costruzione, che realizza un migliore controllo del binomio tempi-costi. Il CPM permette di fare un trade-off tra i tempi di sviluppo e costi relativi al nuovo prodotto; di eliminare l'incertezza sui tempi considerando valori deterministici; di individuare la durata di ogni attività collegandola al costo; di definire la durata che rende minimo il costo totale del progetto.

- PDM - *Precedence Diagram Method*

Modalità di gestione dei reticoli in termini di legami. Nei casi in cui c'è la necessità di effettuare una sovrapposizione delle attività, il PDM permette di creare quattro legami differenti:

- 1) Legame FS (*Finish to start*): la fine dell'attività precedente si lega con l'inizio dell'attività seguente;
- 2) Legame FF (*Finish to finish*): la fine dell'attività seguente dipende dal completamento di quella precedente;
- 3) Legame SS (*Start to start*): l'attività seguente non può iniziare se non è già iniziata la precedente;
- 4) Legame SF (*Start to finish*): la fine dell'attività seguente dipende dall'inizio dell'attività precedente;

- CCM - *Critical Chain Method*

Il metodo della catena critica tiene in considerazione il fatto che quando si lavora su un progetto non ci sono solo le motivazioni tecniche, ma anche quelle relative agli aspetti umani e al modo in cui si lavora. Quindi, è una tecnica di programmazione che tiene conto dell'influenza umana sui tempi, al fine di consentire il completamento del progetto entro le scadenze previste e ridurre la durata totale. Per garantire un certo grado di flessibilità, è necessario assicurare la presenza di buffer, cioè di riserve di tempo e risorse, all'interno del progetto. Dopo aver creato il reticolo, questo meccanismo prevede di iniziare a definire i tempi dal fondo procedendo a ritroso; per tale motivo è infatti definito "Scheduling Backwards".

- CRD - *Critical Resource Diagram*

Questo metodo è un'estensione del CPM che consente di individuare la distribuzione delle risorse nel tempo. A differenza del CPM, si scambia la denominazione delle attività con quella delle risorse, pur mantenendo invariato il reticolo; per tale motivo viene detto "Gantt delle risorse", a cui si associa alla durata delle attività il numero di risorse impiegate.

### 3) ESECUZIONE

Fase in cui avvengono lo sviluppo, la costruzione, l'installazione e i test del progetto.

### 4) MONITORAGGIO E CONTROLLO

Il monitoraggio e controllo è una fase che si sviluppa nel momento in cui parte l'esecuzione del progetto; il monitoraggio è orientato a misurare i lavori, mentre il controllo a prendere delle decisioni sulla base delle osservazioni fatte. Nello specifico, la prima attività consiste nel misurare l'andamento del progetto, definire le metriche di misura e un set di procedure usate per raccogliere informazioni e misurare le performance, confrontare le misure e calcolare gli scostamenti rispetto a quanto programmato, delineando il trend del progetto. Invece, la seconda permette di contenere il progetto all'interno degli obiettivi iniziali, analizzando e prevedendo le cause di eventuali variazioni; definisce inoltre nuovi obiettivi in base ai quali riprogrammare il progetto. Lo scopo di questa fase è quello di misurare regolarmente le prestazioni del progetto, individuare e correggere la variazione rispetto a quanto previsto a budget e nello Scheduling.

### 5) CHIUSURA

Le attività che segnano la chiusura del progetto hanno luogo solo dopo l'approvazione del deliverable finale. In questa fase avviene la risoluzione del contratto, il passaggio di responsabilità al cliente e la sua accettazione della soluzione consegnata. Un'altra azione fondamentale è la catalogazione della documentazione utilizzata durante il progetto, che potrebbe servire anche per la gestione di futuri progetti in cui utilizzare parti comuni o simili; successivamente, avviene la raccolta di tutti gli aspetti positivi e negativi riscontrati nell'esecuzione, che costituiscono la "Lesson Learned", che andrà ad arricchire la cultura aziendale. Nei grandi progetti può rivelarsi utile l'organizzazione di un incontro, detto Close-out meeting, in cui il Project Manager presenta i risultati del progetto, i problemi riscontrati e le sue valutazioni. Tuttavia, nonostante la fine del progetto, nel contratto potrebbe anche essere previsto un servizio di manutenzione e di assistenza post-vendita.

Il ciclo di vita di un progetto può essere gestito con diverse metodologie di Project Management. Gli approcci metodologici più utilizzati per la gestione delle attività di un progetto comprendono metodi agili, interattivi, incrementali o basati sulla successione di fasi predefinite. A prescindere dall'approccio scelto, sarà sempre richiesto un particolare accorgimento per la definizione degli obiettivi del progetto e per la definizione delle responsabilità di tutti gli stakeholder coinvolti.

Nei capitoli seguenti del presente lavoro saranno approfondite le principali metodologie utilizzate per la gestione di un progetto; seguirà la presentazione di casi applicativi in cui tali metodi sono stati usati nel contesto aziendale di Altea UP, società Golden Partner SAP, che ha permesso lo studio del proprio "Verticale" al fine di delineare e definire i metodi usati. Tra questi, il principale risulta l'approccio Hybrid Framework, approfondito per conto dell'azienda per poterlo replicare con maggiore uniformità anche in progetti futuri.

## Capitolo secondo

### Metodologia Tradizionale Waterfall

#### 2.1 Origini e caratteristiche del modello Waterfall

La metodologia di gestione tradizionale dei progetti Waterfall, anche nota come *Linear Sequential Life Cycle Model*, è stata teorizzata nella seconda metà del Novecento da Winston W. Royce nel *paper Managing the development of large software systems* (1970). Per la nascita del modello si guardò all'industria manifatturiera e al settore delle costruzioni, per poi applicare questa metodologia nei processi di realizzazione del software quando questi iniziarono a essere percepiti come un'attività industriale.

Sulla base di tale metodologia, il processo di sviluppo del software si struttura in una sequenza lineare di fasi ben definite e non sovrapponibili, ognuna delle quali prevede differenti attività. Ogni fase può essere eseguita solo una volta e portarla a termine in maniera corretta è il prerequisito essenziale per accedere alla successiva. I deliverables di ciascuna fase sono utilizzati come input per il passo seguente e in tal modo non è consentito tornare indietro per apportare modifiche retroattive. Royce sosteneva che:

l'ordinamento degli step è basato sul concetto che, a mano a mano che uno step procede e la progettazione si fa più dettagliata, si crea un'iterazione di uno step con il precedente e il successivo, ma raramente con gli step più lontani nella sequenza.<sup>9</sup>

Pertanto, le attività sono svolte con una certa sequenzialità e non possono essere parallelizzate (*Figura 9*);<sup>10</sup> tuttavia, sono prevedibili e gestibili sulla base di un piano già definito sul quale si basa il progetto stesso. Da qui deriva la definizione “a cascata”, non direttamente coniata da Royce (che se ne servì solo come paragone), apparsa ufficialmente per la prima volta in un *paper* del 1976 di T.E. Bell e T.A. Thayer.

---

<sup>9</sup> Winston W. Royce, *Managing the development of large software systems* (1970).

<sup>10</sup> Massimo Ragazzi, *Tutte le differenze tra Agile Project Management e il PM tradizionale*, in “Agile School”, <https://www.agile-school.com/blog/agile-project-management-vs-pm-waterfall>.

Così come in una cascata, l'acqua fluisce da un'altitudine più alta ad una più bassa, il ciclo produttivo procede sequenzialmente, da uno stage ad un altro.<sup>11</sup>



*Figura 9. Serial Lifecycle*

La rigidità della metodologia Waterfall dipende dal contesto in cui è nata e si è sviluppata: come si è detto, affonda le sue radici nei settori tradizionali dell'ingegneria edile e dei cicli produttivi, all'interno dei quali gli scostamenti dal progetto originario sono onerosi in termini di costi e di tempi, e le singole fasi si svolgono necessariamente in sequenza, secondo un ordine prestabilito e senza la possibilità di ripetizioni o modifiche dopo la loro conclusione. Questi vincoli rendono necessaria una pianificazione accurata del progetto, che preveda l'individuazione e l'analisi dei requisiti con un certo anticipo rispetto all'inizio del progetto stesso ed è importante che sia effettuata con la massima precisione, al fine di evitare qualunque possibilità di cambiamento in corso d'opera. La pianificazione deve inoltre essere resa nota a tutti i membri dei teams di lavoro coinvolti e opportunamente documentata. Una documentazione completa ed esaustiva diventa l'elemento cardine dell'intero progetto, in quanto precisa gli obiettivi e le procedure da effettuare per raggiungerli.

Tuttavia, tale scelta progettuale di prevenzione dei cambiamenti potrebbe determinare il rischio che il prodotto diventi già obsoleto ancor prima del suo rilascio, o che non sia conforme alle aspettative del cliente, dal momento che ne esclude il contributo nelle fasi di progettazione e sviluppo. La consegna avviene sempre al termine di tutte le fasi.

Dunque, nonostante l'obiettivo del modello Waterfall sia quello di realizzare un progetto nel migliore dei modi mediante un'accurata pianificazione, sul piano pratico quest'ultima non si rivela sempre e comunque corretta, principalmente a causa della difficoltà nel raccogliere reali requisiti sin dagli albori del progetto stesso. Ne derivano diverse conseguenze: il parallelizzarsi di alcune fasi del metodo – eventualità esclusa sul piano teorico – e la comparsa di errori nelle fasi finali del progetto, con uno spreco di tempo e denaro. Per

---

<sup>11</sup> M. McCormick, *Waterfall vs Agile Methodology* (2012).

un'analisi più accurata dei vantaggi e degli svantaggi del modello Waterfall, si rimanda al paragrafo 2.4 del presente capitolo.

## 2.2 Fasi della metodologia Waterfall

Come si è già detto, la metodologia Waterfall prevede fasi non sovrapponibili<sup>12</sup>, strutturate in una sequenza lineare, ognuna delle quali produce un output utilizzato come input per la successiva.

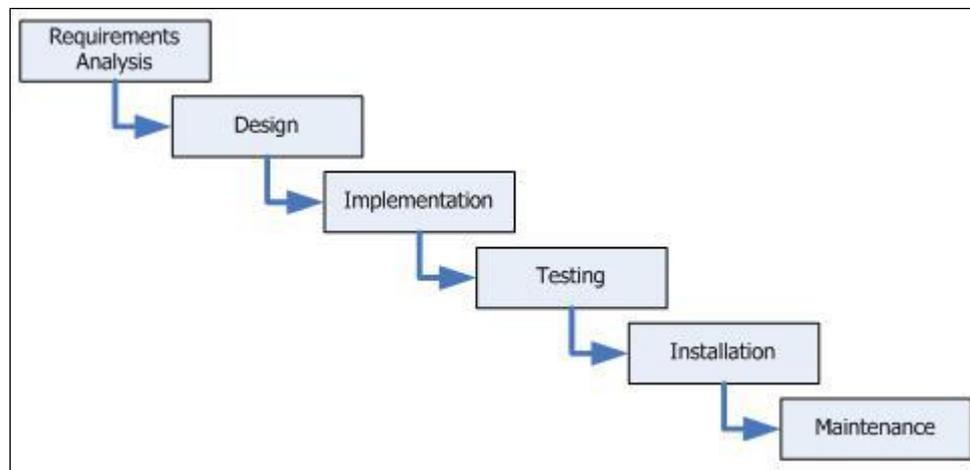


Figura 10. Fasi della metodologia Waterfall.

Si esaminano ora nel dettaglio le singole fasi della metodologia (Figura 10):

### 1) Studio di fattibilità e Analisi dei requisiti

In questa fase preliminare si analizzano le condizioni necessarie all'avvio del progetto. Lo scopo di tale analisi è quello di identificare i requisiti necessari al software per soddisfare le richieste del cliente, in anticipo rispetto all'inizio del progetto. Si raccolgono inoltre tutte le informazioni utili alla stesura di un documento di specifica dei requisiti, che dovrà essere condiviso con il team di lavoro e nel quale è riportata un'analisi di costi, risorse e benefici attesi, metriche di successo e tempistiche. In particolare, occorrerà esplicitare sia le

---

<sup>12</sup> Metodologia Agile e PMBOK: Waterfall e metodo iterativo, in "Humanware – Project Media Center", <https://www.humanwareonline.com/project-management/center/pmbok-agile/>.

Specifiche Funzionali con cui indicare cosa farà il prodotto (ottenibili attraverso un confronto con il Business dell'azienda cliente), sia la User Interface con il dettaglio di come agirà il prodotto.

## **2) Progettazione**

In questa fase l'architettura complessiva del progetto è definita a partire dalle specifiche funzionali, dalle informazioni e dai documenti redatti nella fase precedente. Tali documenti sono studiati e valutati dagli sviluppatori software, con il fine di capire in che modo elaborare la soluzione più adatta alle esigenze del cliente.

## **3) Implementazione**

La pianificazione definita nel corso delle fasi precedenti inizia a concretizzarsi con l'effettiva codifica del software. Gli sviluppatori utilizzano i documenti di progettazione ricevuti per creare un prodotto funzionale, inizialmente implementando il codice in piccoli pezzi successivamente integrati al termine di questa fase. Se gli step precedenti sono stati eseguiti in modo corretto e minuzioso, l'implementazione risulta forse la fase più breve dell'intero processo.

## **4) Test**

Si tratta della fase di collaudo, durante la quale il sistema realizzato è oggetto di vari controlli al fine di verificarne la correttezza e la corrispondenza con le specifiche del cliente, con segnalazione agli sviluppatori di eventuali errori o anomalie prima che avvenga la consegna. È prevista la possibilità che l'utente esegua un test di accettazione (UAT) al fine di verificare che i requisiti espressi nelle specifiche siano stati implementati correttamente.

## **5) Sviluppo**

In questa fase avviene il rilascio del sistema che ha superato il collaudo, con la relativa installazione nell'ambiente di produzione del cliente. Sono incluse varie attività, a partire dalla predisposizione dell'ambiente di lavoro fino allo svolgimento di percorsi di formazione degli utenti per di trasmettere le conoscenze necessarie per un corretto utilizzo.

## **6) Manutenzione**

Dopo il rilascio del software e la consegna al cliente, si eseguono le attività che costituiscono la fase di manutenzione, per far fronte a possibili difetti o eventuali malfunzionamenti. Il

software è oggetto di attento monitoraggio e potrebbero essere richieste necessarie modifiche attraverso nuove integrazioni. A riguardo, si individuano tre tipi di manutenzione:

- manutenzione evolutiva: aggiunta di funzionalità al prodotto;
- manutenzione correttiva: interventi che hanno lo scopo di correggere eventuali errori o anomalie del software, non rilevate durante la fase di collaudo o emerse con l'utilizzo da parte degli utenti;
- manutenzione adattiva: modifiche eseguite sul software per adattarlo ai cambiamenti dell'ambiente in cui viene utilizzato.

### *2.3 Contesti applicativi*

La metodologia Waterfall è utilizzata principalmente in progetti caratterizzati da attività, budget e tempi ben definiti, oltre che da requisiti e processi perfettamente descrivibili in anticipo rispetto alla fase iniziale del progetto. Questo è possibile solo in contesti applicativi stabili, nei quali la pianificazione risulta appunto prevedibile e non incline a cambiamenti in corso d'opera. Eventuali anomalie devono essere facilmente rilevabili già nelle prime fasi, attraverso una pianificazione e accurate analisi, dal momento che il prodotto sarà sottoposto a collaudo solo dopo la fase di implementazione ed eventuali requisiti errati risulterebbero difficilmente modificabili a metà del processo.

Requisiti descrivibili, tempistiche predefinite e non mutabili, deliverables standard da realizzare e sequenzialità delle fasi: sono questi gli elementi fondamentali del contesto stabile richiesto dal modello. Solo rispettando tali condizioni si garantisce che il prodotto rilasciato sia proprio quello pianificato nelle prime fasi in base alle indicazioni fornite. Per queste motivazioni, spesso la metodologia Waterfall è utilizzata per replicare prodotti e servizi già noti, che presentano un'incertezza progettuale minima.

Se i requisiti del progetto sono ambigui o poco chiari – scenario che si verifica quando il cliente ha solo un'idea generica di ciò che gli occorre, senza riuscire a fornire delle specifiche adeguate – il modello a cascata risulta inadatto e pertanto sconsigliato. La sua struttura rigida e sequenziale non permette di fronteggiare facilmente il cambiamento e non è dunque indicata per contesti in continua evoluzione.

## 2.4 Vantaggi e svantaggi del metodo

Dopo aver analizzato la struttura della metodologia Waterfall e i suoi contesti applicativi, è opportuno tracciare una panoramica generale dei vantaggi e degli svantaggi che la caratterizzano.

### Vantaggi

- **Pianificazione e progettazione semplificate**

A partire dalle prime fasi del progetto, il team di sviluppo e il Business dell'azienda cliente identificheranno i requisiti e la soluzione finale che dovrà essere rilasciata alla fine del progetto. Questo permetterà di stabilire anticipatamente le tempistiche di ogni fase, con una riduzione dei tempi di consegna, e di agevolare il monitoraggio dell'avanzamento del progetto, evidenziando eventuali anomalie sin dal principio. In tal modo, sarà possibile ottimizzare anche la gestione del budget, con un significativo risparmio di denaro. Inoltre, come già sottolineato, il requisito fondamentale per ogni fase è che quella precedente sia già stata conclusa, garantendo che al termine del progetto non ci siano elementi lasciati in sospeso.

- **Documentazione accurata**

Ogni fase della metodologia Waterfall è minuziosamente documentata (si parla di metodologia *document driven*), a partire dall'analisi dei requisiti, delle tempistiche e delle attività necessarie per consegnare la soluzione finale entro la data prevista. Al termine di ogni fase, si effettua una revisione della documentazione che sarà poi presa in esame nel momento successivo, con una semplificazione dello scambio di informazioni sia con gli stakeholders coinvolti, sia con i membri del team, per offrire un'accurata panoramica delle attività svolte nelle fasi precedenti.

La documentazione ben dettagliata, che non lascia spazio a fraintendimenti, consente perciò di gestire il progetto anche con personale che non possiede necessariamente skills elevate.

- **Facilità di gestione del progetto**

Al termine di ogni fase, si produce un deliverable che sarà sottoposto a un processo di revisione prima di essere dato in input alla fase successiva, consentendone l'avvio. Il flusso lineare e cadenzato seguito dalle fasi rende il progetto a cascata facile da gestire, poiché

consente di capire in ogni istante in quale fase si trova il progetto, favorendone così il controllo e il monitoraggio.

- **Gestione del team di lavoro**

Ogni team è a conoscenza delle attività da svolgere e delle tempistiche da rispettare, sulla base delle indicazioni riportate nella documentazione. Questo agevola la gestione del team di lavoro anche nei casi di sostituzioni o ampliamento dei membri, in quanto è necessario solo fare riferimento alla documentazione affinché tutti siano allineati e consapevoli delle attività svolte e mancanti.

- **Approccio Hands-off**

La presenza del cliente risulta necessaria solo nelle prime fasi della metodologia (studio di fattibilità e analisi dei requisiti), che dovranno essere discusse congiuntamente dalle parti. A partire dai momenti di progettazione e implementazione, la figura del cliente non è più richiesta, fino al momento di consegna del prodotto finito.

### Svantaggi

- **Rigidità**

Il modello Waterfall, come già osservato, tende a essere piuttosto rigido sin dalla prima fase; una volta definiti i requisiti, questi possono essere modificati solo tramite azioni specifiche, che richiedono una maggiore quantità di tempo al momento dell'analisi, con un conseguente ritardo dell'inizio della fase di progettazione. In casi estremi, se i requisiti non sono chiari e precisi, il metodo può rivelarsi anche del tutto inefficace.

Se nel corso dello sviluppo del progetto emergono nuove funzionalità o mutano le condizioni a contorno, tale approccio poco flessibile potrebbe far risultare già obsoleta la soluzione tecnica rilasciata al termine del progetto, o anche non soddisfare le aspettative del cliente. Un altro elemento di rigidità consiste nella sequenzialità delle fasi, poiché nella realtà raramente i progetti seguono lo stesso flusso lineare, e ritornare a una fase precedente per apportare modifiche può rivelarsi un'operazione onerosa.

Pertanto, flessibilità e cambiamento sono caratteristiche non compatibili con la natura del metodo Waterfall e, invece che apportare benefici, rischierebbero di causare importanti perdite di tempo e di denaro.

- **Difficoltà nel determinare i requisiti**

La fase iniziale del progetto consiste in un'analisi dei requisiti effettuata dagli analisti funzionali insieme al Business del cliente, in un momento preliminare all'inizio del progetto stesso. Se nel corso di tale analisi dovessero presentarsi particolari criticità, si potrebbero verificare rallentamenti o errori, con ripercussioni sull'intera sequenza di fasi. È irrealistico dare per scontato che tutti i requisiti siano definibili sin dal principio, e bisogna tenere conto del fatto che il cliente potrebbe non formularli immediatamente in modo esaustivo o richiedere modifiche in corso d'opera.

- **Assenza di comunicazione con il cliente**

Come si è detto, il confronto con il cliente è limitato alle fasi iniziali e finali del progetto, quindi ai momenti di analisi dei requisiti e a quelli di test e rilascio della soluzione finale. Restano escluse dalla comunicazione le fasi centrali e questo potrebbe comportare che il cliente riceva un prodotto diverso dalle sue aspettative, in un momento terminale in cui eventuali cambiamenti si rivelerebbero assai difficili e costosi, o in cui la comparsa di nuove funzionalità potrebbe generare problematiche importanti. Dunque, l'approccio hands-off – che per alcuni progetti può risultare vantaggioso – in generale rischia di rivelarsi controproducente, non portando a risultati pianificati a causa della mancanza di un riscontro da parte del cliente nelle fasi centrali del processo.

- **Fase di test**

La fase di test segue quella di implementazione del prodotto, e questo può comportare la possibilità di errori o anomalie non evidenti e rilevabili fino alle ultime fasi del processo, quando apportare modifiche risulterebbe estremamente oneroso e una rilavorazione potrebbe essere spesso infattibile. Tale limite rende il metodo Waterfall poco efficiente nei settori dinamici e in continua evoluzione.

- **Tempi di consegna più lunghi**

La metodologia Waterfall non risulta ideale per progetti di grandi dimensioni, poiché eventuali ritardi riscontrati in una singola fase potrebbero avere delle ripercussioni su tutte le successive, posticipando di conseguenza la data di consegna del prodotto finito. Tale effetto è dovuto al meccanismo sul quale si basa il metodo: ogni fase può iniziare solo dopo il completamento della fase precedente.

- **Poco spazio per la creatività**

Documentazione iniziale e processi rigorosi sono due caratteristiche essenziali del modello Waterfall. In tal modo, risulta difficile che si generino nuove idee all'interno del team, non essendo previsto uno spazio per soluzioni innovative e creative.

Al di là degli svantaggi, i punti di forza della metodologia Waterfall restano comunque significativi e il modello continua a essere un punto di riferimento canonico per comprendere le sue evoluzioni e derivazioni moderne.

## Capitolo terzo

### Metodologia Agile

#### 3.1 Origini del Manifesto e del metodo Agile

A partire dai primi anni Duemila si registra un significativo cambiamento del mercato: il modello Waterfall e, in generale, le metodologie di sviluppo software tradizionali iniziano a non essere più adeguate per gestire progetti complessi, in contesti applicativi dinamici ad alto livello di incertezza e con difficoltà nell'identificare i requisiti della soluzione fin dal principio del progetto stesso. Emerge così la necessità di sviluppare nuove metodologie più flessibili, che prevedano una riduzione dei tempi e della documentazione solitamente richiesta.

In risposta alle nuove esigenze, è stata teorizzata una nuova filosofia di sviluppo software, l'*Agile Software Development* (ASD), fondata su principi e valori che furono discussi nel febbraio 2001 in Utah (USA) da diciassette progettisti software ed esperti informatici, che formarono l'*Agile Alliance*. Dalle loro valutazioni emerse con maggiore forza l'inadeguatezza del metodo ingegneristico per la codifica di nuovi software, attività non produttiva ma creativa, in cui l'interazione umana è fortemente necessaria. Un articolo<sup>13</sup> del 2001 mostra infatti come solo il 28% dei progetti IT completati con il metodo Waterfall rispettasse tempi e budget previsti (*Figura 11*).

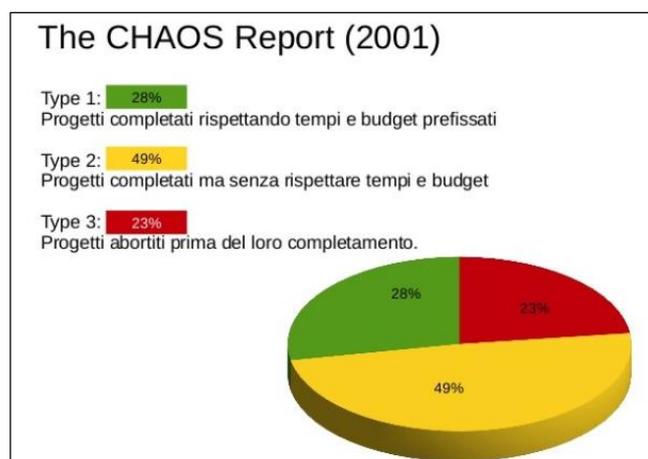


Figura 11. Percentuale di progetti Waterfall completati secondo le previsioni.

<sup>13</sup> *Extreme Chaos*, The Standish Group, [https://www.cin.ufpe.br/~gmp/docs/papers/extreme\\_chaos2001.pdf](https://www.cin.ufpe.br/~gmp/docs/papers/extreme_chaos2001.pdf).

La nuova metodologia Agile sopperisce alle mancanze di quella Waterfall basandosi su un approccio agile, rapido e leggero, caratteristiche fortemente richieste per creare processi snelli, iterativi, flessibili, meno ancorati alla pianificazione e a un'accurata documentazione e pienamente rispondenti alle richieste e alle necessità del committente.

Pertanto, non si parla più di una procedura rigida e lineare, bensì di una snella e adattabile, basata su singole fasi iterative (dette sprint), che garantisce rapidità nel rilascio di nuovi programmi e flessibilità nella loro realizzazione grazie anche a una maggiore interazione fra gli stakeholders coinvolti nel progetto.

Affinché un metodo possa definirsi Agile deve rispettare quattro principi fondamentali:

- **Gli individui e le interazioni più che i processi e gli strumenti**

I membri del team di lavoro assumono maggiore importanza rispetto ai processi e agli strumenti da utilizzare, poiché lo sviluppo del software deve essere guidato dalla creatività della squadra e dalle esigenze del cliente. Il raggiungimento dell'obiettivo finale risulta fortemente legato alla coesione e all'affiatamento del team, che ha la possibilità di organizzarsi e adattarsi al meglio ai cambiamenti.

- **Il software funzionante più che la documentazione esaustiva**

Come osservato, la metodologia Waterfall è vincolata alla presenza di un'accurata documentazione, la stesura della quale richiede sforzi notevoli e tempi talvolta molto lunghi. L'Agile, al contrario, richiede unicamente le informazioni necessarie agli sviluppatori per la realizzazione del progetto: avere un software efficiente, o piccole funzionalità pronte a essere rilasciate al termine di ogni iterazione, è più utile di una documentazione dettagliata.

- **La collaborazione col cliente più che la negoziazione dei contratti**

È fondamentale che il cliente sia coinvolto in tutte le attività progettuali, dal momento che è spesso impossibile raccogliere in anticipo i requisiti e sono necessari feedback costanti per sviluppare la soluzione più adeguata alle richieste e alle esigenze del cliente stesso. Non occorrono, dunque, negoziazioni di contratti a scadenze e intervalli prestabiliti.

- **Rispondere al cambiamento più che seguire un piano**

La struttura iterativa del metodo Agile consente di apportare modifiche in risposta a cambiamenti di esigenze o condizioni particolari, favorendo sempre e comunque uno sviluppo costante. La rapidità è una caratteristica essenziale che garantisce efficacia nel reagire a un mutamento anche improvviso.

Tali valori hanno trovato una consacrazione formale nel *Manifesto per lo Sviluppo Agile del Software*<sup>14</sup>, elaborato dai diciassette esperti dell'*Agile Alliance*<sup>15</sup> e articolato in dodici punti, che aggiungono un ulteriore livello di completezza e chiarezza:

- 1) La nostra massima priorità è soddisfare il cliente rilasciando software di valore, fin da subito e in maniera continua.
- 2) Accogliamo i cambiamenti nei requisiti, anche a stadi avanzati dello sviluppo. I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente.
- 3) Consegniamo frequentemente software funzionante, con cadenza variabile da un paio di settimane a un paio di mesi, preferendo i periodi brevi.
- 4) Committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto.
- 5) Fondiamo i progetti su individui motivati. Diamo loro l'ambiente e il supporto di cui hanno bisogno e confidiamo nella loro capacità di portare il lavoro a termine.
- 6) Una conversazione faccia a faccia è il modo più efficiente e più efficace per comunicare con il team ed all'interno del team.
- 7) Il software funzionante è il principale metro di misura di progresso.
- 8) I processi agili promuovono uno sviluppo sostenibile. Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante.
- 9) La continua attenzione all'eccellenza tecnica e alla buona progettazione esaltano l'agilità.
- 10) La semplicità - l'arte di massimizzare la quantità di lavoro non svolto - è essenziale.
- 11) Le architetture, i requisiti e la progettazione migliori emergono da team che si auto-organizzano.

---

<sup>14</sup> *Manifesto per lo sviluppo Agile Software*, <https://agilemanifesto.org/iso/it/manifesto.html>.

<sup>15</sup> *Agile Alliance*, <https://www.agilealliance.org/>.

12) A intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza.

Generalizzando e astruendo i principi dal contesto di sviluppo di un software, è possibile individuare gli elementi sui quali si basa il metodo:

- Team snelli, capaci di autogestirsi e costituiti da figure cross-funzionali che congiuntamente possedano tutte le competenze necessarie al completamento del deliverable. Le comunicazioni interne devono essere frequenti, al fine di allineare tutti i membri e di favorire discussioni costruttive e funzionali al raggiungimento degli obiettivi.
- Attività gestite temporalmente tramite iterazioni, con lo svolgimento delle fasi di analisi, progettazione, implementazione e test. A ogni iterazione è possibile apportare nuove funzionalità alla soluzione o eventuali correzioni, sulla base dei feedback forniti nell'iterazione precedente, per migliorare la soluzione in quelle successive.
- Possibilità di effettuare dei test a ogni iterazione per verificare il corretto funzionamento della soluzione software.
- Le interazioni con gli stakeholders e il riscontro da parte del mercato permettono al team di lavoro di migliorare le proprie strategie al fine di trarre le consegne. A ogni iterazione il cliente fornisce dei feedback, testando ed eventualmente approvando il risultato ottenuto. Le sue impressioni saranno integrate nelle iterazioni successive, al fine di ridurre i rischi e di fornire un deliverable il più aderente possibile alle sue aspettative.

Questi principi, tuttavia, non richiedono solo implicazioni tecniche per essere applicati: è necessario anche un cambiamento nell'approccio delle persone coinvolte, in particolare modo riguardo alla leadership. I team hanno ormai la facoltà di autogestirsi: si parla di *servant leadership* perché il comando non è più una prerogativa esclusiva del manager.

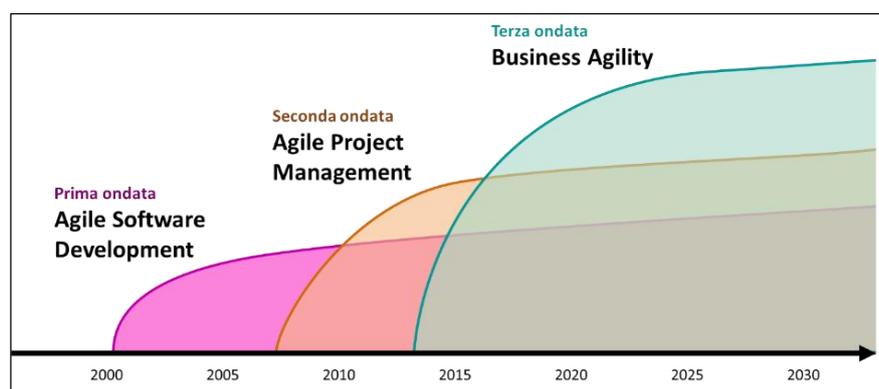
Questo fa comprendere in che modo l'avvento dell'Agile sia stato rivoluzionario e abbia apportato una vera e propria trasformazione: ha influenzato l'intera struttura delle aziende e le loro strategie competitive, modificando il modo di lavorare e anche le pratiche di Project Management; queste ultime, grazie all'introduzione di nuovi concetti e approcci strettamente

legati all'ambito IT, hanno tratto grandi benefici dalla nuova metodologia di Agile Project Management.

Occorre, a questo punto, mettere in evidenza le principali differenze fra la nuova gestione di project management e la tradizionale:

- Soddisfare il cliente è più importante che soddisfare specifiche predefinite.
- Si diffonde una gestione dei progetti adattativa, non più restia ai cambiamenti, bensì capace di affrontarli.
- La pianificazione de progetto non avviene più in modo preciso e minuzioso: diventa strategica e propensa alle modifiche in corso d'opera.
- Le fasi di lavoro non sono più serializzate ma iterate, con deliverables testabili prodotti alla fine di ogni sprint.
- Team in grado di auto-organizzarsi, limitando l'influenza del project manager.
- I processi burocratici tra le fasi diventano più snelli.

Per potenziare e sfruttare al massimo i benefici di questo modello, è necessario che l'intera azienda ne condivida i valori, favorendo così un processo di innovazione continua e l'affermarsi della Business Agility<sup>16</sup>: con questa espressione si fa riferimento alla capacità di riconfigurare strategie, processi operativi, competenze e tutto ciò che garantisce all'azienda la possibilità di adattarsi rapidamente in caso di modifiche del contesto circostante (*Figura 12*).



*Figura 12. Agile: dallo sviluppo software alla trasformazione del business.*

<sup>16</sup> Marco Mazzucco, *Agile transformation: come cambia il modo di lavorare per affrontare la trasformazione digitale*, <https://www.zerounoweb.it/cio-innovation/metodologie/agile-transformation-come-cambia-il-modo-di-lavorare-per-affrontare-la-trasformazione-digitale/>.

Con la diffusione del nuovo modello, sono nati molti frameworks riconducibili alla famiglia Agile, poiché consentono di modificare le specifiche del progetto in base alle interazioni costanti con il cliente e con l'ambiente circostante. La più utilizzata è la metodologia SCRUM, che sarà analizzata più avanti nel presente capitolo.

In ogni caso, data la diversità dei contesti applicativi reali, le aziende tendono ad adattare questi frameworks in modo peculiare, in base al tipo di progetto di cui si occupano.

Tuttavia, nonostante le diversità applicative delle metodologie appartenenti alla famiglia Agile, queste presentano una struttura e delle caratteristiche comuni che si rifanno certamente ai principi e ai valori contenuti nel *Manifesto*, come l'iterazione e la ricerca di un riscontro costante da parte del cliente.

### 3.2 *Struttura del metodo*

Lo scopo dei nuovi metodi agili è quello di ottimizzare i tempi e le risorse destinate al progetto, garantendo, come si è detto, una maggiore flessibilità nella gestione progettuale attraverso la costante comunicazione fra il cliente e un team cross-funzionale dotato di potere decisionale e capace di organizzarsi da sé. Perciò, l'intero progetto è adattato alle esigenze del cliente, che deve essere coinvolto in ogni istante e fornire di volta in volta un feedback positivo o negativo.

La metodologia si basa su una pianificazione a breve termine, che consente di far fronte ai cambiamenti, con la modifica degli obiettivi effettuata periodicamente e della priorità delle fasi in base alle necessità. Così è impossibile che il progetto fallisca: in qualsiasi momento, anche su iniziativa del cliente, è possibile intervenire per verificare il soddisfacimento delle aspettative e definire eventualmente nuovi requisiti e nuove funzionalità.

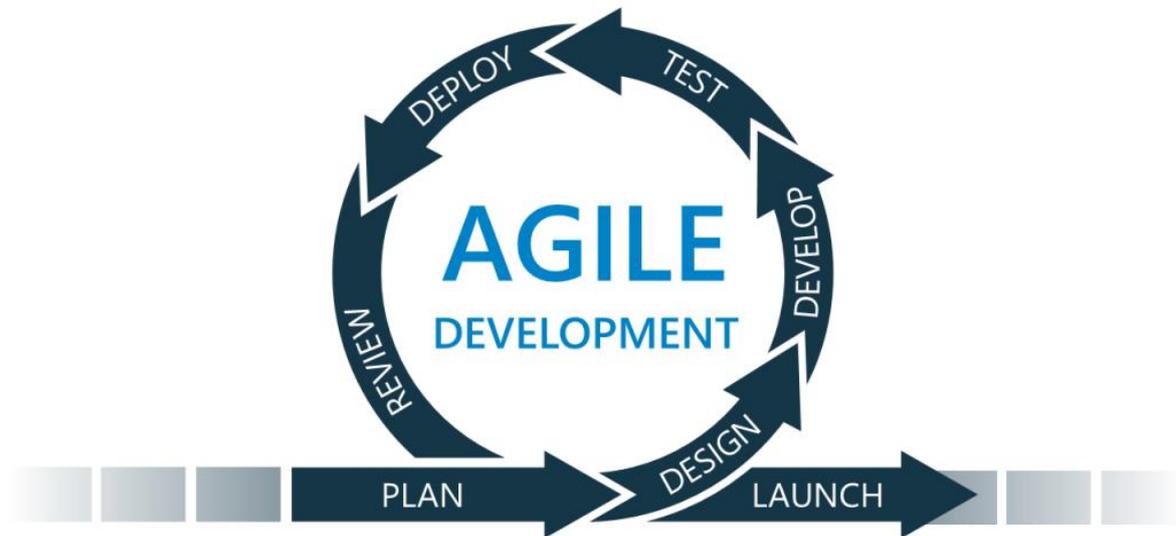
Rilasciare soluzioni software funzionanti il più rapidamente possibile è una caratteristica naturale del metodo Agile, poiché si sono ridotti i tempi destinati alle fasi di analisi dei requisiti e di stesura della documentazione.

Da un punto di vista tecnico, si può affermare che le fasi del nuovo metodo sono le medesime del modello tradizionale, ma con una rivisitazione in chiave iterativa:<sup>17</sup> in ogni iterazione

---

<sup>17</sup> Jason Doll, *Effective Agile sprint cycles for data consulting projects*, in "Neal Analytics", <https://nealanalytics.com/blog/effective-agile-sprint-cycles-for-data-consulting-projects/>.

sono inclusi tutti gli step previsti in un processo di sviluppo software tradizionale, tuttavia si lavora sulle singole funzionalità o sui frammenti della soluzione finale invece di concentrarsi sulla sua interezza (*Figura 13*).



*Figura 13.* Struttura di uno sprint.

Nel corso di ogni iterazione si realizza una parte della soluzione finale, che prende forma al termine di tutti i cicli. Inoltre, a differenza del modello Waterfall, le fasi si evolvono con il progetto stesso e in certi casi è possibile che alcune di esse – come quelle di sviluppo e di test – siano svolte in parallelo a ogni iterazione.

### 3.2.1 Fasi

- **Analisi dei requisiti e pianificazione**

Nella prima fase di ogni iterazione sono definiti i requisiti necessari alla realizzazione delle funzionalità o dei frammenti che saranno rilasciati al suo termine. Nelle iterazioni successive sarà possibile prendere in considerazione nuovi requisiti e sfruttare i feedback delle precedenti per ottenere un risultato il più possibile conforme alle esigenze e alle aspettative del cliente. Si effettua, inoltre, un calcolo del budget previsto, si creano i team di lavoro e si attribuisce un ordine di priorità alle attività.

- **Progettazione**

I requisiti formulati nella fase precedente sono oggetto di studio del team di sviluppo, con lo scopo di individuare gli strumenti più adeguati per ottenere la soluzione desiderata. Nel corso delle varie iterazioni sono inoltre discusse le metodologie utilizzate e la struttura della soluzione software.

- **Implementazione**

Si tratta di una fase di sviluppo e codifica, attraverso la quale si mira a fornire un software funzionante in base ai requisiti dell'iterazione e a feedback precedenti.

- **Test**

A ogni iterazione, le funzionalità aggiunte sono oggetto di test.

- **Sviluppo**

Al termine dell'iterazione, avviene il rilascio della soluzione pianificata al cliente, affinché questa possa essere utilizzata e valutata.

Con le metodologie tradizionali la pianificazione delle attività è eseguita una volta sola (così come tutte le altre fasi e attività) e riguarda l'intero processo produttivo. Al contrario, nel metodo Agile l'asse temporale si suddivide negli sprint,<sup>18</sup> intervalli di tempo prestabiliti (con una durata definita in settimane), nei quali si ripetono le fasi precedentemente descritte (*Figura 14*).

---

<sup>18</sup> *Scrum: come funziona la metodologia Agile più diffusa*, in "Mònade", <https://monade.io/it/scrum-come-funziona-la-metodologia-agile-piu-diffusa/>.

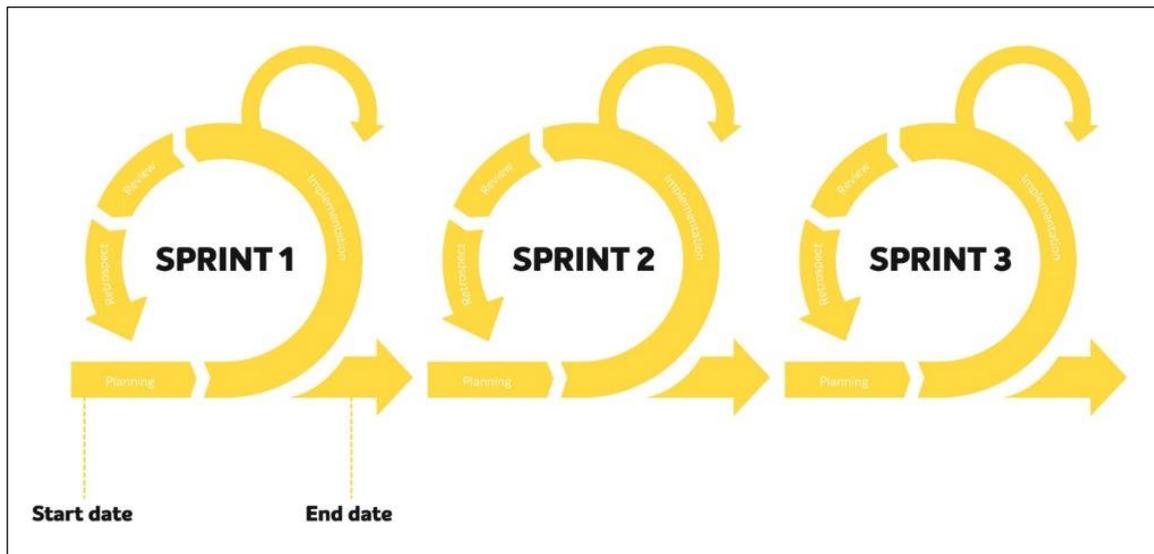


Figura 14. Struttura del processo Agile.

All'inizio di ogni sprint si pianifica con il cliente una soluzione che è sottoposta a processi di valutazione e test, sia da parte del team di progetto, sia da parte del cliente stesso. Solo al termine dell'intero processo la soluzione è consegnata al cliente, senza il rischio di anomalie o errori.

Ogni iterazione ha una durata predefinita generalmente di quattro settimane (si parla di timeboxed), che permette di rientrare nei costi e nei tempi concordati con il cliente, oltre che facilitare il monitoraggio dell'intero processo. In caso di mancato completamento delle attività entro la durata prevista, queste sono riallocate con nuovi livelli di priorità, al fine di ottenere risultati di alta qualità nei tempi inizialmente previsti.

Dunque, si può dire che il metodo Agile è costituito da cicli di consegna a breve termine e che in ogni iterazione sono ripercorse tutte le fasi del modello tradizionale Waterfall, con due importanti differenze: il coinvolgimento continuo del cliente e la possibilità di apportare modifiche tra un'iterazione e l'altra. Ogni sprint è perciò paragonabile a un piccolo progetto che, una volta concluso, rilascia una parte della soluzione software finale.

### 3.2.2 Proprietà

Il processo di sviluppo software Agile è caratterizzato dalle seguenti proprietà:

- **Struttura iterativa e incrementale**

La soluzione software è realizzata in modo incrementale attraverso la conclusione di fasi iterative; ogni iterazione ha una durata fissa che deve essere necessariamente rispettata. Al termine del processo, avviene il rilascio di una parte della soluzione finale.

- **Coinvolgimento del cliente**

Come si è detto, il cliente, per tutta la durata del processo, fornisce dei feedback e testa il risultato di ogni iterazione, al fine di ridurre i rischi e garantire il raggiungimento delle sue aspettative.

- **Gestione delle priorità**

Al termine di ogni iterazione, si rivalutano le priorità date alle varie azioni in base ai rischi di sviluppo e a nuove potenziali esigenze del cliente, chiaramente dando precedenza a quelle con priorità maggiore.

- **Adattabilità**

Considerando che i requisiti necessari al raggiungimento della soluzione finale sono difficilmente reperibili a priori, il metodo Agile predilige l'adattabilità alle modifiche, con la conseguente capacità di integrare nel processo produttivo gli eventuali cambiamenti delle condizioni a contorno.

- **Team autogestisti**

Generalmente, i team coinvolti nel progetto sono di piccole dimensioni e hanno la facoltà di autogestirsi; si parla di *servant leadership*, dal momento che il comando non è più una prerogativa esclusiva del manager.

- **Centralità delle persone**

I singoli individui sono più importanti del processo stesso. La documentazione e le altre attività non propriamente di sviluppo e test sono ridotte al minimo.

- **Sviluppo e rilasci rapidi e frequenti**

I tempi della fase di sviluppo sono molto rapidi, con cicli ridotti e feedback più frequenti. In tal modo, anche i rilasci del software o di sue funzionalità diventano altrettanto frequenti.

- **Test**

Nel corso di ogni iterazione, il team di lavoro e il cliente effettuano frequentemente test e controlli.

- **Semplicità**

Fondamentale è mantenere ogni cosa il più semplice possibile, pur restando propensi al cambiamento.

### *3.3 Contesti applicativi*

Alla luce di quanto detto, è chiaro che la metodologia Agile non è la scelta consigliata in caso di progetti dall'impianto più tradizionale. Nata nel contesto di piccole aziende flessibili di sviluppo software, ha avuto una notevole influenza anche in altri ambiti in cui tendenzialmente si applicavano modelli Waterfall, come l'industria manifatturiera o l'edilizia. L'applicazione di metodi Agile in questi contesti si è però rivelata spesso complessa e controproducente: gestire lo sviluppo incrementale di un software è totalmente diverso da operazioni come la gestione della costruzione di un edificio.

Al contrario, le virtù dell'Agile sono apprezzabili in casi in cui non è possibile effettuare pianificazioni o stime accurate, quando i tempi di consegna sono brevi e se esiste la necessità di un processo con fasi iterative che possono essere facilmente modificate in base alle esigenze del cliente. Tali caratteristiche si possono riscontrare:

- in progetti dai requisiti incerti o potenzialmente mutevoli in corso d'opera, senza perciò la necessità di un'accurata pianificazione e con la previsione di modifiche;
- in caso di clienti con l'esigenza di rilasciare sul mercato prodotti funzionanti in tempi molto ridotti;

- in ambito di Start Up, all'interno delle quali non è possibile fare previsioni di budget o definire tempistiche di consegna.

### 3.4 *Vantaggi e svantaggi del metodo*

Dopo aver analizzato la struttura della metodologia Agile e i suoi contesti applicativi, è opportuno tracciare una panoramica generale dei vantaggi e degli svantaggi che la caratterizzano.

#### Vantaggi

- **Client focalizzato**

Il cliente è coinvolto in ogni fase del progetto e interagisce con il team durante l'intero svolgimento, fornendo feedback a ogni iterazione per introdurre eventuali modifiche.

- **Struttura iterativa**

La struttura del modello Agile è iterativa e incrementale e permette al progetto di cambiare in parallelo ai suoi requisiti, garantendo così flessibilità di fronte ai cambiamenti.

- **Team agili**

Le squadre di lavoro sono autorganizzate e cross-funzionali.

- **Obiettivi periodici**

L'approccio Agile prevede fasi di rilascio incrementale del prodotto. Questo consente al cliente di ricevere frammenti della soluzione finale nel corso dell'intero svolgimento del processo e non più al termine senza possibilità di modifiche.

- **Tempistiche ridotte**

La modalità di sviluppo incrementale (che consente di concentrarsi su parti ridotte di grandi progetti), la minore importanza attribuita alla pianificazione e alla stesura di una documentazione completa permettono la riduzione significativa dei tempi di consegna.

- **Test e revisioni frequenti**

La fase di test non è più eseguita un'unica volta al termine del processo di sviluppo, ma viene ripetuta ad ogni iterazione, a garanzia di una soluzione sempre efficiente.

- **Nessun rischio di obsolescenza**

La soluzione rilasciata non è più considerabile vetusta, dal momento che i tempi del processo si sono drasticamente ridotti e anche grazie alle continue modifiche e revisioni a cui è sottoposta la soluzione a ogni iterazione.

- **Predisposizione ai cambiamenti**

A ogni iterazione, i requisiti o le funzionalità della soluzione possono essere modificati e adattati a nuove necessità. Attraverso i test effettuati e i feedback del cliente dopo il rilascio delle soluzioni parziali, è più semplice intercettare eventuali anomalie e correggerle modificando il sistema.

### Svantaggi

- **Coinvolgimento del cliente**

Il cliente è costantemente coinvolto in ogni fase e iterazione del progetto e i suoi feedback potrebbero portare a continue rilavorazioni o aggiunte di funzionalità inizialmente non definite, determinando così un aumento dei costi e dei tempi.

- **Time-box**

Il metodo prevede rilasci continui di nuove funzionalità al termine di ogni sprint. Tali consegne programmate e la possibilità di variare le priorità delle attività potrebbero portare a posticipare alcune attività nelle iterazioni successive. Inoltre, l'eventuale creazione di sprint aggiuntivi può comportare un aumento dei costi e delle tempistiche richieste dal progetto. Ancora, con il rilascio di soluzioni parziali nel breve termine, si corre il rischio di perdere la prospettiva di lungo periodo o la visione di insieme.

- **Team**

Per rendere le attività il più efficienti possibile, è richiesto un impegno totale ai membri del team, che devono dedicarsi a un solo progetto. Pertanto, per favorire le relazioni e

l'operatività di ognuno, è consigliato al team di trovarsi nello stesso ambiente lavorativo, anche se questo non è sempre possibile. Inoltre, l'alto livello di qualificazione richiesto potrebbe costituire uno svantaggio.

- **Pianificazione e documentazioni ridotte**

La documentazione prodotta durante le fasi è poco dettagliata: questo crea problemi ai futuri utilizzatori, siano essi clienti o altri membri del team. Nel caso di progetti complessi, le sole conoscenze tacite del gruppo difficilmente si rivelano sufficienti al completamento della soluzione finale senza una documentazione condivisa ed esaustiva che guidi lo svolgimento delle attività.

Con la mancanza di una pianificazione completa e concreta fin dall'inizio del progetto, si corre il rischio di non individuare le reali esigenze del cliente, oltre a lasciare spazio a fraintendimenti se neanche il committente ha le idee chiare sulla soluzione più adeguata a rispondere alle sue necessità. L'assenza di un piano iniziale può diventare la causa del rilascio di un prodotto finale diverso da quello proposto o desiderato.

Dunque, lo snellimento delle fasi di pianificazione e stesura della documentazione non deve sfociare in assenza di comunicazione o nel mancato rispetto delle regole e della disciplina.

- **Costi**

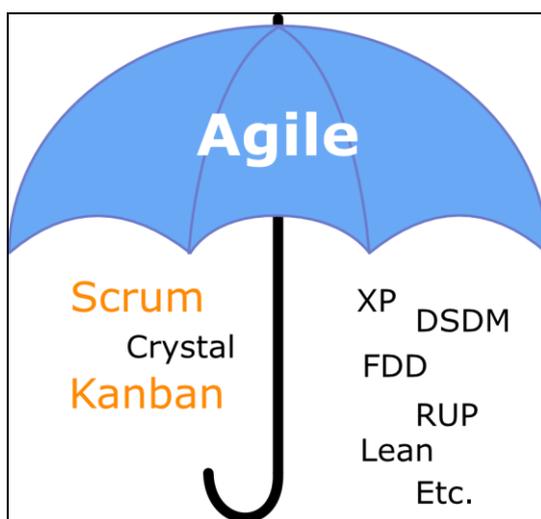
A causa di eventuali modifiche e rilavorazioni tra un'iterazione e l'altra, i costi di progetto sono più difficili da stimare, e in ogni caso potrebbero non essere inferiori rispetto ad altre metodologie tradizionali.

- **Iterazioni**

A ogni iterazione è sempre possibile attuare modifiche in base ai feedback degli sprint precedenti. Le frequenti rilavorazioni possono comportare tempistiche più lunghe e costi più elevati. Inoltre, l'analisi delle funzionalità effettuata a ogni iterazione rende difficile la previsione della data di consegna del progetto.

### 3.5 Metodo SCRUM

La metodologia Agile è spesso paragonata a un ombrello<sup>19</sup> sotto il quale confluiscono diversi metodi che condividono principi e valori del *Manifesto Agile*. Tra questi si ricordano metodologie come Scrum, Extreme Programming, Crystal, DsDm e molte altre definite agili proprio perché, a differenza di quelle tradizionali, permettono di apportare modifiche nelle fasi di sviluppo attraverso una stretta collaborazione con il cliente (*Figura 15*).



*Figura 15. Agile Umbrella.*

Scrum è attualmente il framework Agile più diffuso. Fu teorizzato nel 1986 da Hirotaka Takeuchi e Ikujiro Nonaka, con l'obiettivo di rendere più veloce e flessibile il processo di sviluppo software. Con le dovute modifiche, è stato utilizzato anche per la realizzazione di progetti di tipo differente. Nel 1995, Jeff Sutherland e Ken Schwaber, due dei sottoscrittori del *Manifesto Agile*, definirono per la prima volta questa metodologia in un loro articolo, ribattezzandola Scrum, letteralmente "mischia". Tale termine è utilizzato nel gioco del rugby per descrivere la strategia usata per riprendere la partita dopo un'infrazione, quando tutti i membri della squadra si impegnano, uniti e compatti, a spingere nella stessa direzione in modo coordinato per ottenere il possesso della palla.

Il parallelismo con le procedure di sviluppo di soluzioni software trova le sue ragioni nel fatto che il metodo è spesso utilizzato in progetti complessi con grandi margini di incertezza, dove è necessaria una grande coordinazione tra i membri del team.

<sup>19</sup> Hans Bruins, *Naked Scrums*, <https://medium.com/@hans.bruins/naked-scrum-81218f6f4833>.

È una strategia flessibile e olistica allo sviluppo del prodotto, dove il team di sviluppo lavora come un'unica entità per raggiungere un obiettivo comune, in opposizione all'approccio sequenziale delle metodologie tradizionali.<sup>20</sup>

Considerando l'appartenenza alla famiglia Agile, anche la metodologia Scrum si basa su processi di tipo incrementale e iterativo per gestire le incertezze progettuali. Il flusso produttivo è scomposto in più iterazioni dalla durata predefinita, gli sprint, al termine dei quali saranno rilasciate nuove funzionalità e incrementi che andranno a costituire la soluzione finale.

In ogni caso, non si tratta di un processo standard, ma di un framework, un insieme di prassi, ruoli e strumenti per la pianificazione e gestione di un processo produttivo, che sarà personalizzato per adattarlo ai modelli delle organizzazioni che lo utilizzano.

Nonostante tale metodo venga applicato con caratteristiche diverse in ogni organizzazione, è necessario che siano sempre osservati i suoi valori e principi fondamentali:

- **Trasparenza**

In ogni applicazione del metodo Scrum è necessario che siano riscontrabili gli aspetti fondamentali su cui si basa il modello Agile – aspetti che non possono essere modificati in quanto standard comuni. Gli obiettivi e gli elementi del progetto devono essere chiari e sempre osservabili da tutti gli stakeholders coinvolti

- **Ispezione**

È necessaria una supervisione in ogni fase del progetto, insieme all'esecuzione di collaudi a ogni iterazione, al fine di intercettare immediatamente eventuali anomalie o errori. Inoltre, anche la documentazione, deve essere oggetto di verifica.

- **Adattamento**

Modifiche o aggiunta di nuove funzionalità devono sempre essere possibili, anche per superare eventuali errori riscontrati nelle ispezioni. L'adattamento deve essere effettuato in breve tempo per evitare ulteriori problemi.

---

<sup>20</sup> Hirotaka Takeuchi, Nonaka Ikujiro, *The New Product Development Game*, in "Harvard Business Review", gennaio-febbraio 1986.

Il framework Scrum<sup>21</sup> si basa su Ruoli, Eventi e Artefatti, nonché su regole a essi associati che li legano insieme (Figura 16).

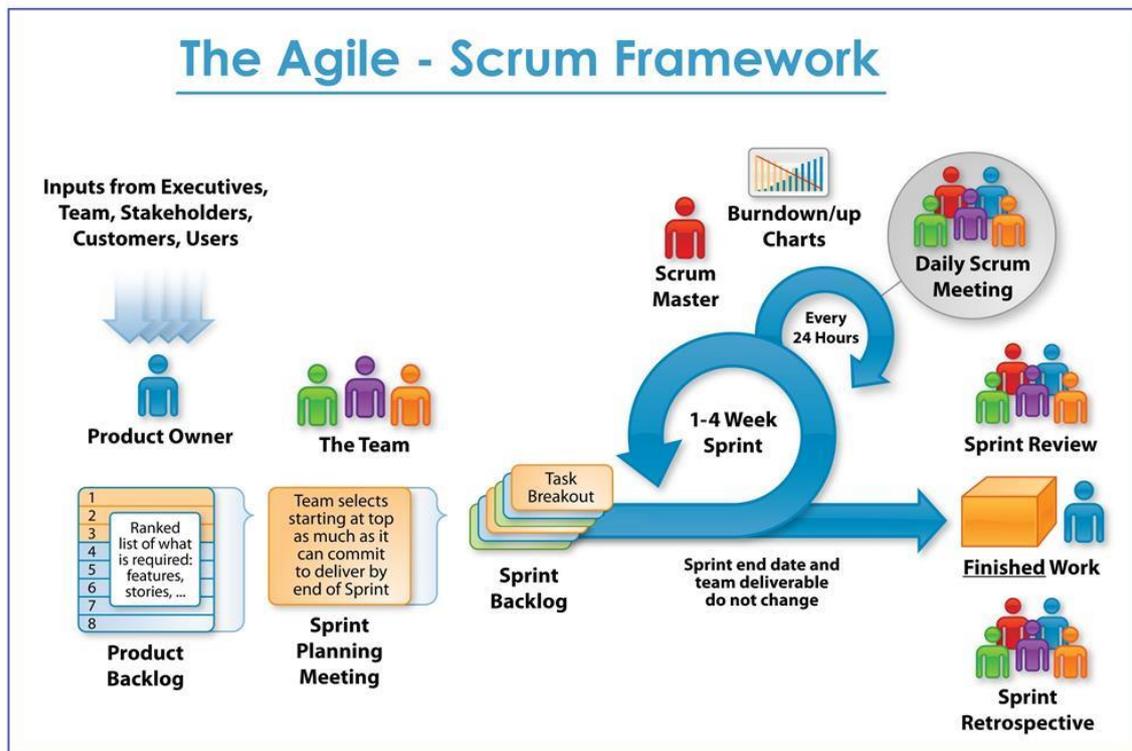


Figura 16. Scrum: flusso di processo.

### 3.5.1 Ruoli

Scrum, rispetto alle altre metodologie agili, è caratterizzato da una struttura gerarchica definita che prevede diversi ruoli, al fine di facilitare la sua applicazione in un contesto aziendale. Un'altra peculiarità è la scalabilità: nonostante sia stato creato per gruppi di piccole dimensioni può essere applicato in certi contesti con team di lavoro più ampi.

I principali ruoli sono:

- **Product Owner**

Il proprietario del prodotto rappresenta gli interessi del cliente e degli stakeholders: ha pertanto un ruolo decisivo nella riuscita del progetto, in quanto è il responsabile del Product Backlog, un documento che contiene i requisiti e le caratteristiche della soluzione che si

<sup>21</sup> Marco Mazzucco, *Agile transformation...*, cit.

intende sviluppare, indicando quali implementare durante i vari sprint, e che detta l'ordine delle attività in base alle priorità di sviluppo del prodotto finale. Il Product Owner deve assicurarsi che questo documento sia sempre aggiornato con i risultati ottenuti a ogni sprint, oltre a garantire che sia chiaro e trasparente e facilmente comprensibile da tutto il team. Un altro documento di cui deve occuparsi è la stesura della User Story, che contiene le specifiche da mostrare al team di sviluppo e sulle quali quest'ultimo dovrà lavorare, ossia informazioni legate all'utente, alle funzionalità e al motivo per cui dovranno essere sviluppate. Inoltre, come responsabile dell'approvazione finale del risultato, definisce le date e i contenuti dei rilasci, oltre ai criteri di accettazione per i deliverables prodotti in ogni sprint.

- **Development Team**

Si tratta di un team di sviluppo capace di auto-organizzarsi, che comprende membri altamente qualificati e dalle competenze cross-funzionali, il cui numero può variare da un minimo di tre a un massimo di nove persone. Il compito è quello di eseguire lo sviluppo tecnico del prodotto e di testarne le funzionalità, rilasciando un incremento della soluzione finale a ogni iterazione in base agli obiettivi assegnati a quello sprint. Un'altra mansione è quella di comunicare allo Scrum Master lo stato di avanzamento delle attività ed eventuali anomalie riscontrate, mentre al Product Owner occorre dare evidenza delle azioni completate.

- **Scrum Master**

Chi riveste questo ruolo è il responsabile della gestione del progetto e della corretta applicazione di regole e processi, nonché leader dello Scrum Team, che tutela da ostacoli esterni e interferenze. Ha il compito di moderare e agevolare le relazioni all'interno della squadra. Infine, assiste il Product Owner nel trasmettere il Product Backlog al resto del team, aiutando a comprenderne il contenuto. In generale si occupa anche della stesura della documentazione richiesta nel processo.

### 3.5.2 *Eventi*

Il framework prevede una serie di eventi fissi, utilizzati in Scrum per creare regolarità e ridurre al minimo l'esigenza di incontri eccezionali, oltre che per monitorare l'andamento

del progetto garantendo trasparenza nelle attività. Ogni evento ha una durata massima ben precisa, per evitare di dedicare troppo tempo al processo di pianificazione.

In base alla struttura iterativa, è possibile individuare quattro eventi limitati temporalmente:

- **Sprint Planning Meeting**

Questo incontro si tiene all'inizio di ogni sprint e coinvolge tutti i membri dello Scrum Team, con l'obiettivo di pianificare le attività da svolgere nella corrente iterazione. Tra i prerequisiti di questa fase si riscontrano la disponibilità del Product Backlog e i risultati dell'iterazione precedente, necessari al team di sviluppo per valutare quali attività previste possano essere sviluppate durante il nuovo sprint, con l'attribuzione a ognuna di esse di un certo livello di priorità. È inoltre possibile individuare uno Sprint Goal, ossia un obiettivo da raggiungere entro la fine dello sprint immediatamente successivo tramite l'implementazione del Product Backlog. Quest'ultimo dovrà essere aggiornato al termine della fase.

- **Daily Scrum meeting**

Si tratta di un incontro giornaliero della durata massima di quindici minuti fra il team di sviluppo e lo Scrum Master, previsto di volta in volta allo stesso orario e nello stesso luogo. Nel corso del meeting, ogni componente del team condivide le attività personali previste per quel giorno con il resto dei membri. In tal modo, tutti sono informati sul programma odierno, con il conseguente miglioramento della comunicazione e l'individuazione di eventuali ostacoli allo sviluppo (che possono così essere eliminati sul nascere). L'incontro consente inoltre di controllare l'avanzamento delle attività necessarie al raggiungimento dello Sprint Goal e di valutare se queste siano allineate alle previsioni dello Sprint Backlog.

- **Sprint Review**

Si tratta di un meeting che si svolge al termine di ogni sprint con tutti i membri del team e i principali stakeholders, che discutono delle attività appena svolte. Si verificano i risultati ottenuti (eventualmente modificando la priorità delle attività non concluse), ci si focalizza su cosa fare e ottimizzare nello sprint successivo e su eventuali criticità già affrontate. Lo scopo principale è quello di capire se l'obiettivo prefissato è stato effettivamente raggiunto e con quali risultati; successivamente, al termine della riunione il Product Owner decide se è possibile rilasciare l'incremento in base ai feedback ricevuti.

- **Sprint Retrospective**

Si tratta di una riunione collocata tra lo Sprint Review di un'iterazione e lo Sprint Planning Meeting dell'iterazione successiva, alla quale partecipa l'intero Scrum Team. In questa sede si affrontano tutte le criticità già individuate negli eventi precedenti, sia se dovute a processi e strumenti utilizzati, sia se legate ad esperienze negative in merito alle relazioni con il resto del team o con il cliente. Lo scopo è quello di creare e attuare un piano di miglioramenti negli sprint successivi.

### 3.5.3 Artefatti

Le metodologie agili – e dunque anche Scrum – presentano una ridotta documentazione rispetto a quelle tradizionali, ma sempre e comunque finalizzata a garantire trasparenza e possibilità di controllo, oltre che ad allineare il team riguardo alle attività. Inoltre, si utilizza per tenere traccia del lavoro effettuato per il completamento di un progetto. Per convenzione, la documentazione prodotta da Scrum è definita con il termine Artefatti e si articola in:

- **Product Backlog**

Questo documento rappresenta l'unica lista di requisiti, caratteristiche e funzionalità da utilizzare per la realizzazione e le modifiche della soluzione finale. Il Product Owner è il responsabile della stesura e della cura del contenuto e dell'ordinamento delle parti, e ne garantisce sempre la disponibilità per i membri del team. Tale artefatto è definito “vivente” in quanto si evolve in parallelo al progetto stesso e all'ambiente di sviluppo ed è continuamente modificato per identificare in ogni iterazione ciò che occorre al raggiungimento dello Sprint Goal. Lo scopo è quello di fornire al team una lista di attività da svolgere durante tutta la durata del progetto, nonché anomalie da correggere, modifiche da apportare o aggiornamenti da effettuare.

È composto dalle user story, che consistono in funzionalità o incrementi previsti per la soluzione da sviluppare durante le varie iterazioni e sono realizzate dal punto di vista del cliente. Comprendono un dettaglio sulle specificità che l'utente desidera avere, secondo un formato predefinito: “Come <ruolo> voglio <requisito> così che <beneficio>”.

- **Incremento**

L'incremento è il risultato del completamento di tutti i punti previsti dal Product Backlog durante uno sprint, sommati a quelli già portati a termine nelle iterazioni precedenti. Deve essere utilizzabile indipendentemente dal fatto che il Product Owner decida di rilasciarlo o meno.

- **Sprint Backlog**

Lo Sprint Backlog è costituito da un insieme di elementi del Product Backlog selezionati per lo sprint corrente: sostanzialmente, fornisce una lista delle attività che lo Scrum Team dovrà ultimare, sulla base delle priorità, per il conseguimento del prossimo Incremento e la realizzazione dello Sprint Goal. Nel corso dello sprint non è possibile modificare la lista dei requisiti indicata nel documento; resta tuttavia possibile aggiungere nuove attività ed eliminarne alcune superflue. Il team di sviluppo si auto-organizza, prendendosi la responsabilità delle modifiche apportate, in funzione di nuovi elementi eventualmente emersi durante lo sviluppo e necessari al raggiungimento dello Sprint Goal.

- **Burndown Chart**

Il Burndown Chart è un grafico che indica quanto sforzo è necessario per ultimare le attività elencate nello Sprint Backlog, ed è utilizzato per valutare il lavoro già svolto e stimare quello che ancora rimane da svolgere. Graficamente, presenta sull'asse delle ascisse il tempo e su quello delle ordinate il lavoro indicato in ore, dato dalla sommatoria dei giorni stimati necessari al completamento dei lavori (*Figura 17*).

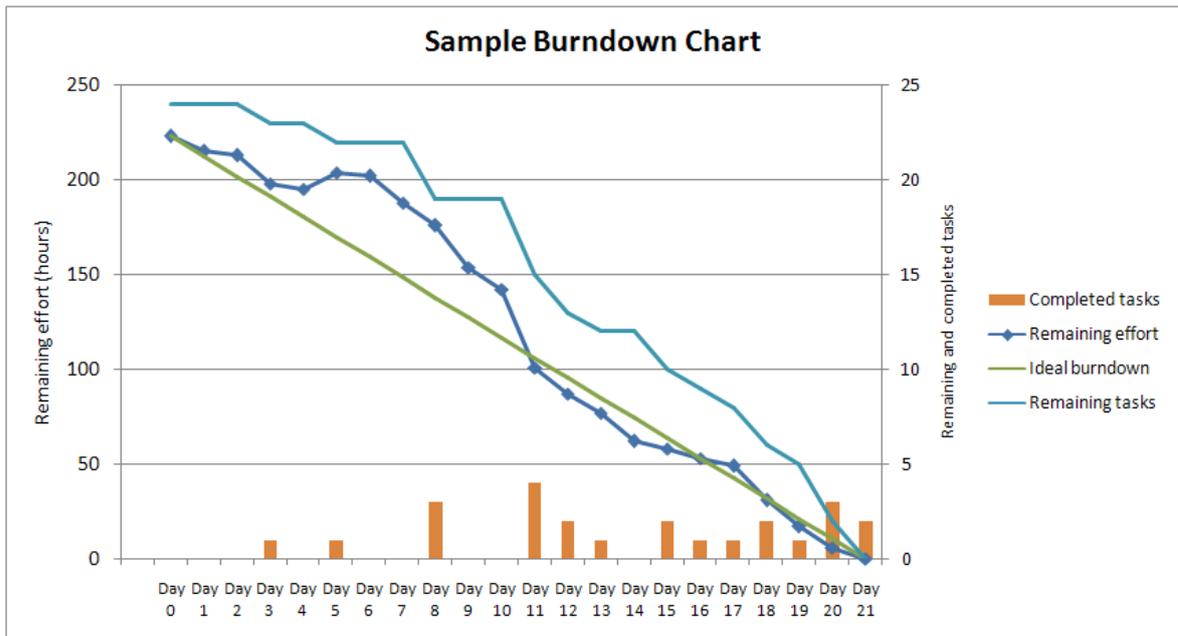


Figura 17. Esempio di Burndown Chart.<sup>22</sup>

### 3.5.4 Vantaggi e svantaggi di Scrum

Come osservato, Scrum è un framework dettagliato strutturato in Ruoli, Eventi e Artefatti ben precisi. Anche tale metodologia, come le altre, presenta vantaggi e svantaggi che saranno esaminati di seguito.

#### Vantaggi

- **Trasparenza**

Attraverso riunioni giornaliere e ruoli ben definiti, i team sono sempre allineati sul corso delle attività e su eventuali anomalie.

- **Responsabilità**

Il team si auto-organizza il lavoro, in base anche alle priorità delle azioni, distribuendo le attività tra i vari sprint. La figura del Project Manager che dirige il team perde potere a favore di una maggiore responsabilizzazione e collaborazione tra i membri.

<sup>22</sup> *Sample Burndown Chart*, da Wikipedia.

- **Libertà di modifica**

Le iterazioni dalla durata predefinita e limitata, insieme a feedback costanti sugli incrementi rilasciati, rendono possibile l'esecuzione di modifiche durante tutto il ciclo produttivo. Il Daily Scrum Meeting è l'occasione in cui i membri dello Scrum Team possono allinearsi sulle attività previste ed effettuate, oltre a segnalare eventuali ostacoli riscontrati in fase di sviluppo per pianificare le opportune modifiche.

- **Costi ridotti**

Grazie agli incontri programmati e frequenti, eventuali anomalie riscontrate diventano subito oggetto di discussione e modifiche, al fine di ridurre i costi e aumentare la qualità di tutte le fasi.

### Svantaggi

- **Scope Creep**

Si tratta di una situazione in cui gli obiettivi di un progetto tendono a espandersi rapidamente durante il suo sviluppo, allontanandosi da quanto era stato inizialmente previsto.

- **Grandi team**

Il framework Scrum è basato su Ruoli, Eventi e Artefatti, tutti elementi pensati per team di piccole o medie dimensioni. Scrum può perdere la sua efficacia se il gruppo di lavoro è troppo ampio.

- **Centralità dello Scrum Master**

Questa figura deve fidarsi del suo team limitandosi a guidarlo; in caso di abuso di potere e controllo da parte sua, la "mischia" potrebbe fallire.

- **Attività imprecise**

Le riunioni devono identificare attività precise, con relativi tempi e costi; in caso contrario, causerebbero danni a tutto il progetto. La pianificazione potrebbe rivelarsi errata e gli sprint potrebbero richiedere tempi aggiuntivi per il raggiungimento dello Sprint Goal.

# Capitolo quarto

## Approccio “Ibrido”

### 4.1 Waterfall vs Agile

Dopo aver analizzato le caratteristiche e il funzionamento delle metodologie Waterfall e Agile, in questa sede si prenderanno entrambe in esame, con un confronto<sup>23</sup> relativo ai loro aspetti e caratteristiche principali (Figura 18).

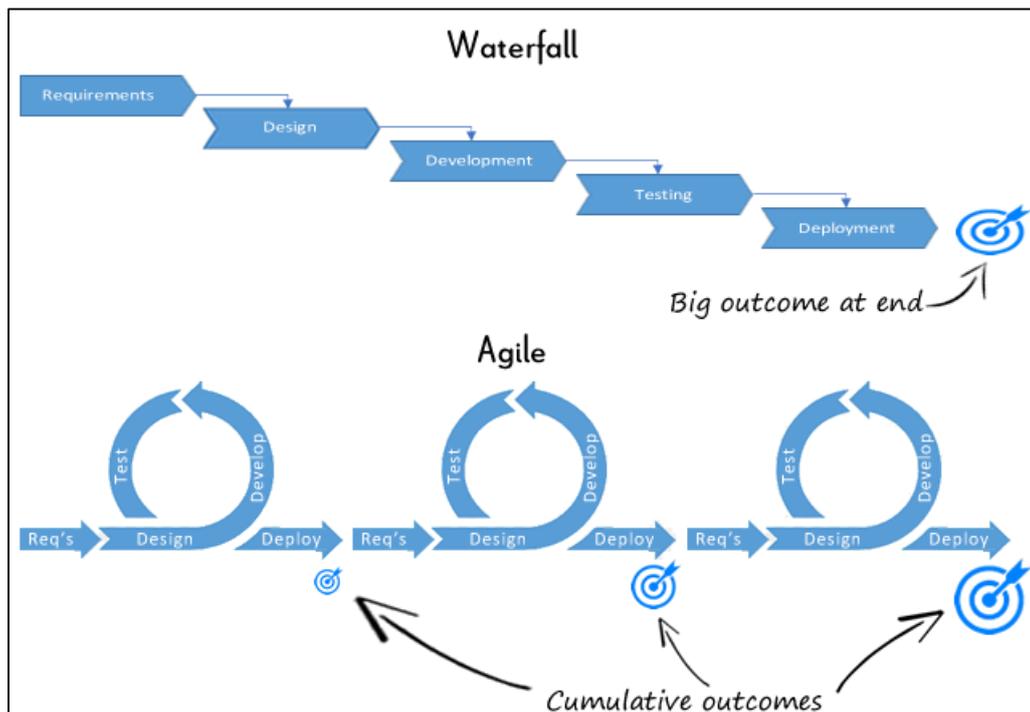
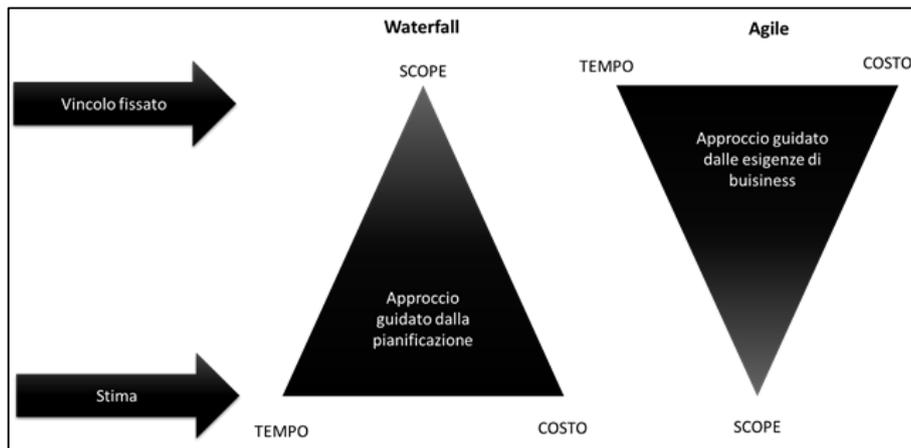


Figura 18. Metodologie Waterfall e Agile a confronto.

<sup>23</sup> Cfr. Daniele Federicis, *Change Management & Project Management*, in “Il blog di Management”, <https://www.blog-management.it/2018/04/10/change-management-project-management/>; Salameh H., “*What, When, Why, and How? A Comparison between Agile Project Management and Traditional Project Management Methods*” in “International Journal of Business and Management Review” (2014), v. 2, n. 5, pp. 52-74.

- Waterfall è una metodologia sequenziale, molto strutturata e rigida, suddivisa in fasi che si verificano una sola volta per tutta la durata del progetto, al termine delle quali sarà fornita un'unica soluzione. Il successo dell'intero progetto dipende dalla capacità di effettuare una pianificazione dettagliata da seguire attentamente. Al contrario, la metodologia Agile è un processo incrementale noto per la sua flessibilità, garantita dalla suddivisione del ciclo produttivo in sprint. Al termine di ogni iterazione, in cui si ripetono tutte le fasi previste già nel metodo Waterfall, si consegna un incremento rispetto alla soluzione fornita nello sprint precedente. Quindi la soluzione finale risulta essere la somma dei rilasci di tutte le iterazioni.
- L'analisi approfondita dei requisiti è alla base del metodo Waterfall per lo sviluppo di un software; non sono previste modifiche in corso d'opera ed è necessario conoscerli sin dall'inizio. La metodologia Agile, grazie alla sua flessibilità, consente sempre di apportare modifiche ai requisiti, anche dopo l'inizio del progetto e la fine della fase di pianificazione, in quanto è previsto che questi cambino e si evolvano tra le varie iterazioni.
- Mentre nella metodologia Waterfall l'intervento del cliente è previsto solo durante l'analisi dei requisiti, in quella *Agile* lo stesso riveste un ruolo principale, in quanto l'obiettivo di ottenere il suo completo soddisfacimento è raggiungibile solo con un coinvolgimento costante in tutti gli step produttivi.
- Una differenza importante tra le metodologie risiede nella centralità della fase di Test. Nei modelli Waterfall questa è effettuata una sola volta dopo la fase di implementazione, al termine del ciclo produttivo. Invece, in Agile sono eseguiti dei test a ogni iterazione, per garantire un livello di qualità maggiore e un rapido intervento in caso di errori.
- Nei metodi tradizionali la documentazione assume un ruolo centrale ai fini del progetto, al contrario dei metodi Agile che vengono soprannominati Lightweight, leggeri, perché presentano una produzione di documentazione nettamente minore.

Dall'analisi delle due metodologie è emerso un altro elemento di differenza<sup>24</sup>: entrambe fanno riferimento a paradigmi decisamente differenti (*Figura 19*). Il metodo tradizionale è di tipo Plan-driven, poiché privilegia lo scopo e mantiene fissi i requisiti andando a calcolare di conseguenza tempi e costi che possono subire modifiche; l'Agile è di tipo Value-driven, in quanto fissa sia il tempo che le risorse, gestendoli con una logica di tipo Timeboxing, lasciando variare l'insieme delle funzionalità.



*Figura 19. Paradigmi a confronto.*

Ancora, mentre Waterfall è molto sensibile ai cambiamenti dei requisiti, specialmente verso la fine del progetto, con Agile il team di sviluppo corre meno rischi e il cliente avrà sempre la possibilità di modificare i requisiti, aumentando il grado di qualità e di soddisfazione degli stakeholders coinvolti. Anche il modo di misurare le performance o il risultato del progetto cambiano radicalmente: se in Waterfall si ritiene che un processo di sviluppo è efficiente solo se sono stati rispettati il budget e i tempi previsti, in Agile il livello di successo è legato al grado di soddisfazione del cliente in base al fatto che il deliverable rispetti o meno le richieste e aspettative.

Nello specifico, analizzando i due metodi dal punto di vista dei soli costi riferiti alla fase di pianificazione e gestione delle modifiche, è possibile trarre le seguenti conclusioni.

Considerando la fase di pianificazione di un progetto, i costi sostenuti da un metodo tradizionale sono molto elevati nella prima fase, per poi ridursi drasticamente una volta

<sup>24</sup> Mirko Menecali, *Approccio Agile al Project Management: meglio del Waterfall?*, in "Sinfo online", <https://blog.sinfo-one.it/waterfall-e-agile-due-approcci-contrapposti-al-project-management>.

avviato il progetto. Al contrario, con i metodi Agili i costi restano costanti e sostenuti fino alla fine, a causa dei continui rilasci di funzionalità e versioni della soluzione (Figura 20).

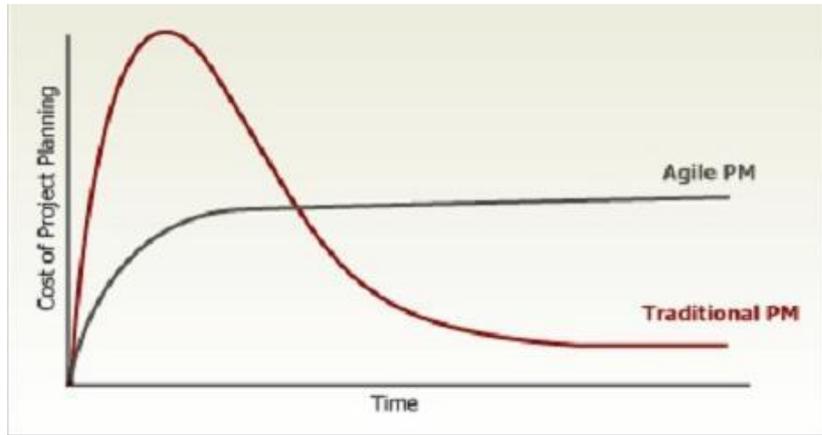


Figura 20. Andamento dei costi legati alla pianificazione di un progetto.

Considerando la gestione delle modifiche, nel caso delle metodologie tradizionali ogni cambiamento rispetto a ciò che è stato pianificato risulta estremamente oneroso e crescente verso le fasi finali, al contrario dei metodi agili dove il costo resta, ancora una volta, stabile. (Figura 21)



Figura 21. Andamento dei costi legati alle modifiche.

Le caratteristiche dei due metodi dipendono anche dai differenti contesti in cui sono utilizzati: Agile è adeguato anche a situazioni incerte, Waterfall si mostra più adatto in presenza di condizioni in cui si richiedono rigore e poca incertezza.

## 4.2 Ibrido

In questi anni, il dibattito su quale sia il metodo migliore tra Waterfall e Agile non è ancora giunto al termine. In molte aziende è stata percorsa una nuova via che porta all'integrazione dei loro aspetti più validi tramite un approccio "ibrido"<sup>25</sup> nella gestione dei progetti (Figura 22).

Un approccio ibrido potrebbe prevedere un'accurata fase di analisi dei requisiti seguita da sprint con consegne incrementali della soluzione finale, o, ancora, stabilire una pianificazione iterativa seguita da un'unica fase di sviluppo. Tale nuovo approccio nasce dal fatto che non tutte le aziende si trovano nelle condizioni di passare a una metodologia Agile a causa di incompatibilità con le politiche organizzative della società, come l'impossibilità di avere un budget fisso e delle tempistiche rigide. Dunque, molte aziende si stanno ponendo l'obiettivo di utilizzare i processi basati sui principi dell'Agile, sfruttando però gli elementi fondamentali delle metodologie tradizionali, andando così ad adottare un modello ibrido capace di gestire parti diverse di un progetto con metodi diversi. Tale obiettivo è tuttavia raggiungibile solo dopo aver individuato le modalità di utilizzo e le aree di integrazioni possibili tra i due modelli, in base al progetto specifico.

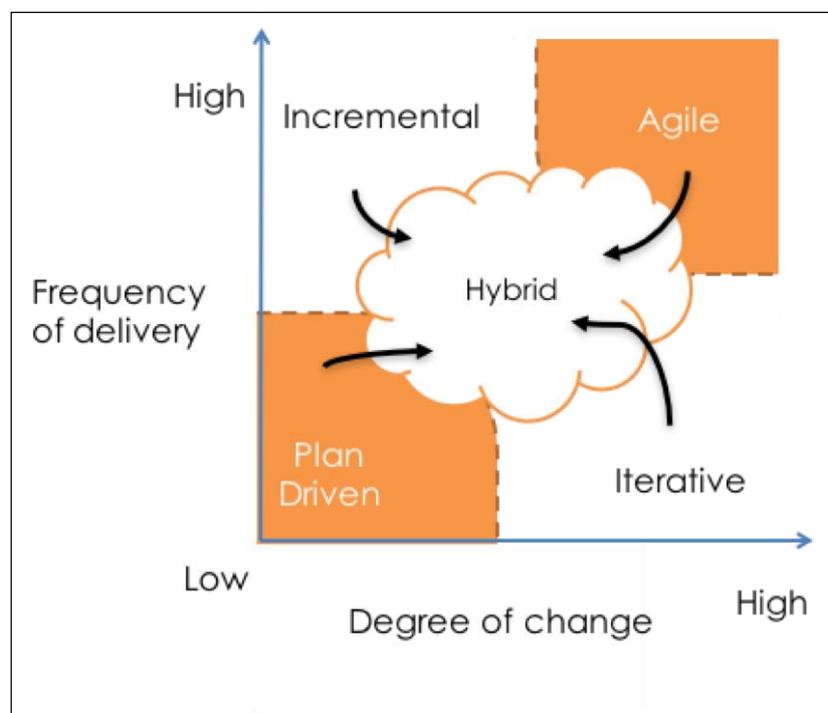


Figura 22. Combinazione di metodi Agile con altre tecniche tradizionali.

<sup>25</sup> *What is Hybrid Agile, Anyway?*, <https://www.agilealliance.org/what-is-hybrid-agile-anyway/>.

Nella realtà aziendale, è possibile riscontrare diverse tipologie di ibridazione. Nei progetti in cui è utilizzato un approccio Waterfall sono introdotti alcuni elementi tipici del modello Agile, ad esempio la presenza di incontri e riunioni di coordinamento o avanzamento, l'analisi delle attività già fatte e l'allocazione di un team a un solo progetto per tutta la sua durata.

Viceversa, nei progetti dove si usa Agile è introdotta una maggiore documentazione a sostegno dello scambio di informazioni, della coerenza dei dati e della tracciabilità di tutto il processo.

In generale, l'approccio ibrido più diffuso prevede lo svolgimento delle prime fasi di analisi dei requisiti e pianificazione mediante la metodologia Waterfall, mentre quelle di sviluppo e test sono eseguite con Agile.

Un altro dei possibili scenari in cui si potrebbe applicare l'ibrido prevede la presenza di aziende di grandi dimensioni, in cui i team non riescono ad adattarsi nel breve periodo alla metodologia Agile, optando per una soluzione intermedia.

In conclusione, non esiste una metodologia privilegiata, ma solo contesti applicativi in cui un determinato approccio risulta più efficace di altri. Ogni modello ha infatti dei punti di forza che, se adeguatamente sfruttati in sinergia, si trasformano in strumenti aggiuntivi che consentono di rispondere efficacemente anche alle situazioni più strane o complesse.

## Capitolo quinto

### Caso di studio aziendale – Altea UP

#### 5.1 Altea UP: panoramica dell'azienda

Altea UP è una società del gruppo Altea Federation, SAP Gold Partner e SAP Intelligent Enterprise Partner Ambassador.

La stretta collaborazione con tutte le società del gruppo, le competenze certificate sulle innovazioni *SAP* e la forte expertise in ambito ERP e gestionale consentono ad Altea UP di proporre al mercato una gamma di servizi completi per ogni area aziendale, garantendo continuità del business, supporto evolutivo, team dedicati, livelli di servizio costantemente monitorati ed elevati livelli di soddisfazione. L'azienda lavora così al fianco dei propri clienti per sostenere le loro sfide di business e guidarli nel percorso di trasformazione digitale verso l'Intelligent Enterprise. L'obiettivo primario dell'intervento di Altea UP consiste, quindi, nel rendere operativa presso il Cliente una soluzione ERP integrata che l'accompagna nella crescita in termini di volumi, aree di business ed evoluzione dei processi aziendali.

Altea UP implementa soluzioni software SAP e sviluppa nuove tecnologie per rendere efficiente l'azienda e l'intera supply chain, costruendo soluzioni innovative tramite strumenti tecnologici che semplificano e migliorano tanto la User Experience quanto i processi di business. Al centro della strategia aziendale ci sono la soddisfazione del cliente e il successo di ogni progetto di trasformazione digitale intrapreso, assicurato dalla metodologia Agile e dalle best practices certificate PC COE (Partner Center of Expertise).

Uno tra i fattori che assicurano il successo del progetto SAP è l'adozione delle Best practice SAP, intese come processi e procedure standard suffragati da innumerevoli implementazioni e anni di sviluppo delle soluzioni SAP. Utilizzare le Best Practice SAP consente di fruire di un ambiente prototipale in tempi rapidi, con la possibilità di utilizzare il sistema già nelle fasi di analisi di dettaglio. Inoltre, tale approccio permette di minimizzare le personalizzazioni e quindi le manutenzioni del sistema nel tempo.

Un'innovativa e articolata proposta di soluzioni (ERP SAP S/4 HANA, Customer Engagement & Commerce, Big data, Predictive Analytics) e il corretto mix di tecnologie (IoT, Intelligenza Artificiale, Machine Learning, Chatbot, piattaforma Cloud) consente ad

Altea UP di proporre al mercato una suite di soluzioni *SAP* per ogni area aziendale, in grado di semplificare e migliorare l'efficienza dei processi. L'azienda è attiva nei settori *manufacturing, automotive, engineering & construction, chemical, pharma, professional services, food & beverage e retail*. Settori in cui ha maturato negli anni una consolidata presenza e competenze professionali su base nazionale e internazionale.

Riguardo l'ambito farmaceutico, la gestione dei processi e delle attività sono caratterizzate da requisiti sempre più complessi. Da qui la consapevolezza di quanto l'elemento IT rappresenti, per le organizzazioni che operano in questo settore, un fattore concorrenziale ed evolutivo essenziale per lo sviluppo del business e delle performance. Altea UP offre un ottimo compromesso tra il bisogno di compliance dei clienti e la stringente necessità di efficienza attraverso una convalida dei processi garantita, integrata e controllata. Con la soluzione Intelligent Pharma di Altea UP le aziende del settore chimico e farmaceutico trovano una risposta concreta all'esigenza di ridurre i costi gestionali e aumentare la qualità dei processi aziendale, attraverso una soluzione verticale altamente referenziata che governa e supporta il processo di convalida dei sistemi informativi. Inoltre, la soluzione S/4 HANA Altea UP Intelligence Pharma per il settore chimico-farmaceutico rispetta le normative GMP, *Good Manufacturing Practices* e CFR 21 che certificano le buone norme per la fabbricazione del farmaco ad enti come AIFA e FDA.

I vantaggi principali che l'introduzione di S/4 HANA può comportare per un Cliente sono così sintetizzabili:

- Sistema integrato: tutte le informazioni gestionali aziendali saranno residenti su un'unica piattaforma applicativa, con una semplificazione del reperimento dei dati aziendali e con l'introduzione di una nuova integrazione dei processi;
- Utilizzando un approccio incentrato sul Cliente, è possibile usufruire di strumenti di analisi delle informazioni, garantendo l'integrazione con tutte le componenti di gestione aziendale;
- Ottimizzazione dei processi interni aziendali e adattamento del sistema informativo al business aziendale;
- Evidente facilità d'uso, che permette di offrire agli utenti modalità di accesso e strumenti di supporto mirati a rendere l'utilizzo del sistema assolutamente indipendente dalla specifica competenza informatica.

Inoltre, la soluzione è particolarmente scalabile in quanto consente a una azienda di dotarsi di volta in volta delle funzionalità ulteriori per rispondere a rinnovate esigenze di business e organizzative.

Con la realizzazione del progetto di implementazione S/4 HANA, il Cliente si prefigge fondamentalmente di raggiungere gli obiettivi di seguito elencati:

- Chiara definizione di un modello per la gestione dei processi di business all'interno dell'organizzazione;
- Centralizzare la gestione dei diversi master data e delle logiche sottese alla gestione dei principali processi di business;
- Armonizzare i processi e le procedure, cogliendo l'opportunità di accedere a best-practice riconosciute a livello mondiale, da promuovere all'interno del gruppo e adottare come facilitatore del processo di evoluzione e ottimizzazione della propria organizzazione;
- Sfruttare al massimo le funzionalità native (standard) di un sistema informativo supportato a livello worldwide;
- Avviare un percorso di evoluzione dei sistemi informativi aziendali, il cui fine sia quello di dotarsi di una piattaforma unica per la gestione dei principali processi di business;
- Avere garanzia di supporto e scalabilità della soluzione da parte di un Vendor riconosciuto come best-of-breed a livello mondiale.

Attraverso l'introduzione del sistema S/4 HANA, il Cliente potrà anche:

- Migliorare ulteriormente il rapporto con i propri clienti, attraverso l'accesso alle fonti di informazione raccolte dall'ERP;
- Controllare e porre le basi per razionalizzare e diminuire i costi interni:
  - Razionalizzazione dei prodotti gestiti (riduzione della varietà) e delle informazioni a essi collegate;
  - Diminuzione, ove possibile, degli "overhead", con un'opportuna programmazione dei costi;
  - Abbassamento dei costi tramite la riduzione dei tempi di attesa e la semplificazione o l'eliminazione dei processi a basso (o nullo) valore aggiunto;
  - Predisposizione al Management di adeguati strumenti di controllo economici e finanziari, attraverso sistemi di governance più efficaci ed efficienti.

- Migliorare il flusso interno delle informazioni mediante:
  - Rinnovamento tecnologico dei sistemi gestionali;
  - Maggiore copertura funzionale di alcune problematiche delicate e complesse non attualmente supportate;
  - Raggiungimento di adeguati livelli di trasparenza, integrità e unicità dei dati e delle informazioni scambiate, sia verso l'interno, sia verso l'esterno dell'azienda.

ALTEA UP, nell'approcciare il progetto di implementazione S/4 HANA, si prefigge come obiettivo quello di progettare il sistema informativo a partire dall'analisi dell'attuale situazione, con forte attenzione volta al futuro, arrivando a definirne uno basato sulle best-practice SAP che, oltre a garantire il supporto ai principali processi di business del Cliente, consente di perseguire contestualmente una razionalizzazione organizzativa, assicurando al contempo:

- Univocità e tempestività dell'informazione (soprattutto per il Management, nell'ottica di governare in tempo reale la situazione);
- Riduzione di costi per riduzione di complessità inutile;
- Maggiore efficienza e qualità nella gestione dei processi transazionali;
- Maggiore mobilità interna di risorse e conseguente ottimizzazione delle stesse;
- Possibilità di estensione e crescita.

Nel presente capitolo sarà preso in esame il “Verticale” di Altea UP, ossia le metodologie di gestione dei progetti più utilizzate dall'azienda. In settori tradizionali come quello chimico e delle costruzioni è spesso adottato un metodo Waterfall, mentre, in contesti più dinamici e soggetti a norme come quello farmaceutico si predilige l'Hybrid Agile Framework. Entrambi sono oggetto dei casi di studio di questa tesi. Al contrario, per la metodologia Pure Agile non sarà affrontato un caso applicativo reale, dal momento che questa non è adottata da Altea UP a causa della sua ridotta documentazione e di una fase di pianificazione e analisi meno approfondita.<sup>26</sup>

---

<sup>26</sup> Il materiale utilizzato per la stesura di questo capitolo è stato fornito dall'azienda Altea Up.

## 5.2 Metodologia Waterfall di Altea UP

La metodologia Waterfall è stata utilizzata da Altea UP per la realizzazione di progetti di implementazione di nuovi sistemi SAP. L'obiettivo è quello di trasferire all'azienda cliente tutte le conoscenze necessarie per controllare e gestire la crescita del sistema informativo, in modo non invasivo rispetto all'organizzazione e al modello di business già in uso.

Il documento di Business Blue Print (BBP) riveste un ruolo centrale perché riporta gli ambiti funzionali identificati e ha come obiettivo quello di definire un modello di implementazione. Tale documento costituisce parte integrante dell'allegato tecnico consegnato al cliente ed è il riferimento fondamentale in termini di contenuto. Inoltre, è un prerequisito all'avvio del progetto, poiché riporta l'esito della fase di analisi dei requisiti in base al quale sono pianificate le attività necessarie.

Le leve su cui opera la metodologia di Altea UP sono (*Figura 23*):



*Figura 23. Leve Altea UP.*

- **People**

Le persone, gli specialisti Altea UP, a vari livelli e con le relative responsabilità ben delineate e condivise, sono le principali risorse su cui fare leva per il successo delle diverse iniziative di intervento.

- **System**

Il sistema è una delle chiavi per il raggiungimento degli obiettivi prefissati, se ben governato e se sfruttate pienamente le funzionalità e potenzialità messe a disposizione dalle nuove tecnologie.

- **Organization**

La corretta identificazione di chi fa che cosa è l'elemento fondamentale per la reale gestione e per il controllo dei processi aziendali, nonché l'attuazione di una concreta razionalizzazione degli stessi e il loro costante e continuativo presidio.

### *5.2.1 Organizzazione progettuale*

La metodologia, prima di tutto, prevede una struttura delle risorse di progetto ben definita e chiara, con ruoli strategici, di controllo e operativi a vari livelli.

L'intervento progettuale di Altea UP relativo all'intero progetto è gestito da un Project Manager nominato da Altea UP e da un Project Manager nominato internamente dal Cliente. Il Project Manager Altea UP è affiancato da diverse tipologie di Consultant esperti, capaci di analizzare le necessità del Cliente e supportarlo nella ricerca di soluzioni idonee alle problematiche organizzative/applicative, scegliendo insieme al Cliente stesso le migliori metodologie di parametrizzazione in funzione delle problematiche da risolvere.

Il Project Manager interno del Cliente, oltre a selezionare e coinvolgere i Business Process Owner in funzione delle necessità di analisi, coordina e supervisiona tutte le attività, assicurando il collegamento e la "sponsorship" con il Management del Cliente.

Pertanto, deve essere opportunamente identificato un Gruppo di Lavoro Interno che prevede per ogni macro area un Referente di Processo (Business Process Owner), un Utente Chiave (Key User), uno o più Referenti Operativi (End User) e un Referente Tecnico IT interno.

Tutte le attività gestite all'interno delle fasi progettuali descritte nei successivi paragrafi saranno documentate attraverso opportuna documentazione progettuale, approvata sotto l'aspetto tecnico dai Business Process Owner e dal Project Manager del Cliente e

formalmente dallo Steering Committee, il comitato direttivo con compiti decisionali di tipo strategico (Figura 24).



Figura 24. Metodologia di progetto Altea UP.

Di seguito si propone una sintesi delle diverse attività e del loro contenuto.

### 5.2.2 Ciclo di vita del progetto secondo il Waterfall di Altea UP

Il metodo di lavoro proposto da Altea UP garantisce la massima aderenza del futuro sistema informativo integrato alle esigenze aziendali, con tempi e costi continuamente tenuti sotto controllo. Tale metodo si sviluppa in diverse attività (Figura 25).

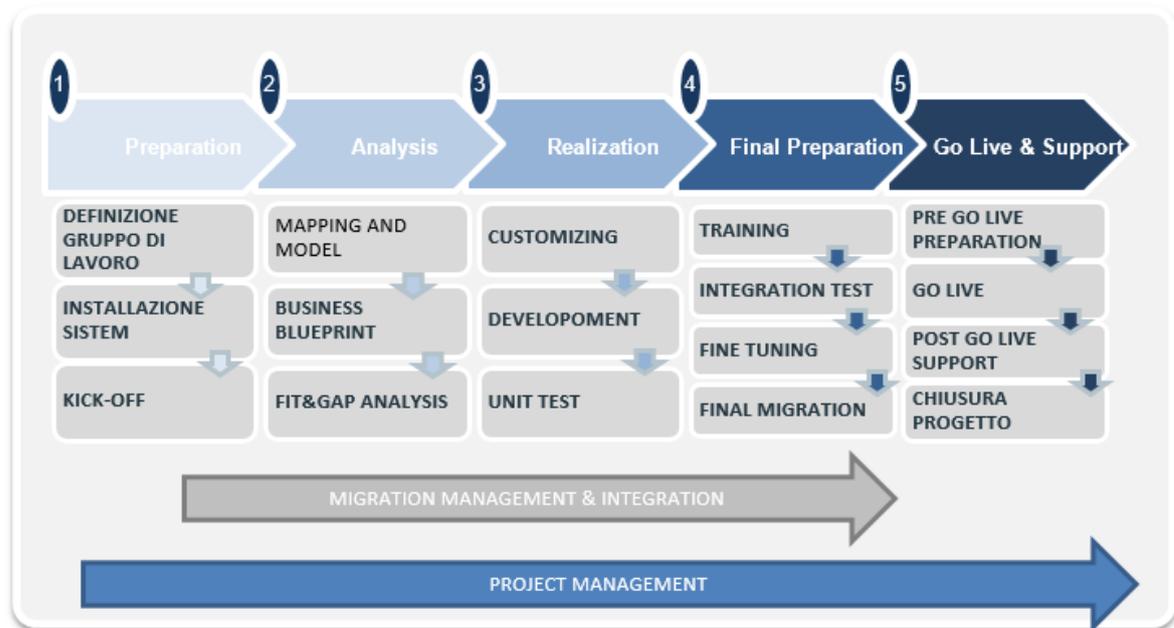


Figura 25. Metodo di lavoro Altea UP.

- **Preparation**

In questa fase si procede alla corretta e completa definizione degli obiettivi e dell'ambito di progetto. Si effettua una pianificazione preliminare di tutte le fasi progettuali previste, considerando la sequenza delle attività e la disponibilità delle diverse risorse. Avviene inoltre la definizione del gruppo di lavoro.

Gli attori coinvolti in questa fase suddividono le attività nel seguente modo:

Altea UP:

- Definisce il gruppo di lavoro progettuale;
- Identifica le responsabilità sia del Team di Altea che del Team Cliente;
- Definisce le regole di lavoro di organizzazione e di gestione della Commessa;
- Identifica le modalità di gestione e di condivisione delle Deliverables di Progetto.

Il Project Manager:

- Richiede l'apertura di un apposito sito di Commessa (utilizzabile sia come repository che come vero e proprio strumento di lavoro) al quale potrà accedere anche il Cliente previa autorizzazione (Richiesta Account Cliente);

- Formalizza e condivide con il Team interno le informazioni relative al Progetto: obiettivi, attori, macro-fasi, modalità di rendicontazione, gestione delle spese e delle trasferte, site di Commessa;
- Presenta ufficialmente il Progetto al Cliente tramite il Kick Off, condividendo obiettivi, metodologia, principali step.

Il Team Altea UP dedicato:

- predisporre gli ambienti necessari allo svolgimento del Progetto (Ambiente di Sviluppo, Ambiente di Quality, Ambiente di Produzione);
- Installa e configura il prodotto SAP e il relativo database di riferimento.

Le attività di installazione e configurazione sono formalizzate attraverso un'apposita relazione (Final Report of System Installation) che deve essere controfirmata dal Cliente. Al termine delle principali Milestone, vengono organizzati degli incontri con il Cliente per la condivisione dello stato avanzamento lavori (SAL). In accordo con il Cliente possono essere previsti eventuali SAL aggiuntivi (settimanali/mensili).

- **Analysis**

Si effettua un'analisi della pianificazione e dei requisiti, che si conclude attraverso l'approvazione del modello descritto nel documento di BBP.

Il Project Manager programma specifici incontri tra i Team Leader Altea UP (referenti di Stream) e i Key User del Cliente con lo scopo di:

- Verificare la copertura funzionale di SAP rispetto alle esigenze aziendali;
- Formulare eventuali ipotesi di customizzazione e/o di revisione di flussi o processi organizzativi;
- Definire i parametri che guidano il comportamento del sistema e le sue funzionalità.

Il Team Altea UP:

- Redige un apposito documento (Business Blueprint) per mappare i processi funzionali e mettere in evidenza eventuali Gap rilevati;
- Censisce accuratamente i Gap individuati sul sito di Commessa (Gap List).

In condivisione con il Cliente vengono definite:

- Le strategie di migrazione del Progetto, le tipologie di dati interessati, le modalità di trasferimento, le responsabilità e le modalità di validazione (Migration Strategy, impostazione Migration Plan);
- Le specifiche relative ai tracciati da ricevere dal sistema attualmente in uso da importare al nuovo sistema (Tracciati di Migrazione).

- **Realization**

Sulla base del documento approvato di BBP, si procede con la fase realizzativa del progetto, attivando quindi lo step operativo di implementazione della soluzione. Sono inoltre sviluppati i seguenti deliverables relativi al Project Management: Customizing, Development e Unit Test.

- Nell'attività di parametrizzazione denominata Customizing, sono impostati sia i parametri che guidano il comportamento del sistema, sia le funzionalità e le modalità censite durante il disegno della BBP.
- A fronte dell'attività di Customizing, Altea UP testerà le singole funzioni (Unit Test) con l'obiettivo di verificare che quanto implementato sia coerente con il modello definito.
- Per quanto riguarda gli sviluppi (development), Altea fornisce uno sviluppatore a supporto, che, sulla base delle indicazioni dell'IT del cliente, procederà negli sviluppi e supporterà il personale nei test interni qualora fosse necessario.

- **Final Preparation**

In questa fase si svolgono le seguenti attività:

- Training

Altea UP effettua delle sessioni di training agli utenti, su richiesta del cliente, restando a disposizione anche per gestire eventuali necessità.

- Integration Test

L'attività di Integration Test viene considerata un'attività di rilevanza strategica per il buon esito del progetto di implementazione del nuovo sistema informativo. Questa attività responsabilizza il Cliente nell'identificazione e nella condivisione della soluzione implementata, con focus sulle procedure e sulla correttezza delle informazioni prodotte. Inoltre, rappresenta il completamento del training degli utenti sulla soluzione implementata. I consulenti di Altea UP si occupano delle attività di preparazione dei casi test (scenari) e della definizione delle regole di controllo dei test effettuati. Lo scopo dell'Integration Test è quello di simulare le condizioni di normale operatività del sistema con dati reali. In questa fase il sistema è testato, quindi, dagli utenti in ottica integrata di processo, con una verifica dei flussi nella loro interezza, disegnati nel modello pianificato. Tale modalità di erogazione consentirà al Cliente di prendere la piena confidenza delle nuove operatività di sistema. L'Integration Test può iniziare solo dopo aver completato il popolamento dei dati reali nell'ambiente SAP dedicato.

- Fine Tuning

È la fase di perfezionamento finale del sistema. Dall'interazione tra i gruppi di lavoro Altea UP e del Cliente durante le fasi di Prototype ed Integration Test, si mettono in opera tutti quei micro perfezionamenti volti alla migliore stabilizzazione del sistema.

- Final Migration

Si effettua la migrazione finale dei dati dal sistema precedente a quello attuale, in funzione delle logiche di migrazione automatica e/o manuale definite (Migration Plan).

In caso di migrazione manuale dei dati, il Cliente è responsabile della preparazione dei dati e del loro caricamento a sistema; in particolare deve:

- Estrarre e bonificare i dati dai sistemi di origine;
- Trasformare i dati per la mappatura secondo i tracciati Altea UP;
- Validare e correggere i dati migrati;

- **Go Live & Support**

Si tratta delle attività che seguono immediatamente il Go Live, con le quali Altea UP e il Cliente gestiscono il corretto funzionamento e la normale operatività delle funzionalità del sistema SAP, per approfondire particolari funzionalità del sistema stesso o delle personalizzazioni, così come per affiancare gli utenti durante il primo utilizzo di SAP per le procedure quotidiane o periodiche.

Altea UP è in grado di fornire al Cliente un servizio di Application Maintenance modulare e flessibile, per garantire un supporto altamente professionale e continuativo nel tempo, capace di accompagnare il Cliente nella gestione di problematiche collegate all'operatività del *day-by-day* e di piccole evoluzioni della soluzione software consegnata, e inoltre di affiancare il Cliente nell'evoluzione dei propri processi di business e dei propri sistemi.

- **Project Management**

In questo ambito rientra l'attività di coordinamento e gestione di tutte le attività del progetto. Il servizio specialistico ha l'obiettivo di garantirne lo svolgimento ottimale, coordinando tutte le risorse coinvolte (interne ed esterne), organizzare incontri per certificare lo stato di avanzamento lavori, aggiornare periodicamente il planning, monitorando eventuali scostamenti tra preventivo e consuntivo agli enti preposti.

### 5.3 Hybrid Agile Framework Altea UP

L'Hybrid Agile Framework di Altea UP contempla le pratiche Agile basate sulla metodologia Scrum per migliorare i metodi tradizionali all'interno dei parametri tipici Waterfall (Figura 26).

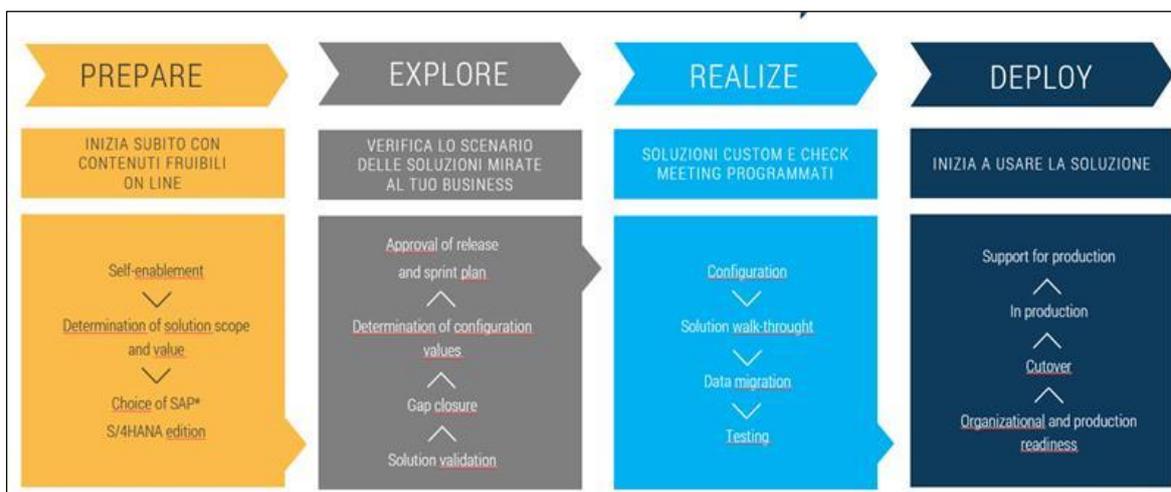


Figura 26. Hybrid Agile Framework di Altea UP.

L'Hybrid Agile Framework è caratterizzato da:

- **Fixed Scope** con tempo e risorse stimati per alcune tipologie di attività;
- **Estimated Scope** con tempi e risorse definite per le componenti Agile;
- **Requisiti** di alto livello definiti all'inizio del progetto, con priorità e modifiche possibili durante l'intero processo;
- **Cross-Functional Team** durante i diversi sprint;
- **Sprint** organizzati per funzionalità e/o processi e categorizzati per priorità.

L'approccio dell'Hybrid Agile Framework si serve pertanto concetti puramente Agile per garantire la velocità, la flessibilità e la trasparenza, utilizzando, laddove necessario, anche i metodi più tradizionali del metodo Waterfall.

L'Agile puro è un approccio empirico e iterativo allo sviluppo del software, che sottolinea un focus continuo sul valore, la flessibilità, il coinvolgimento del cliente e la rapida consegna delle funzionalità (Figura 27).

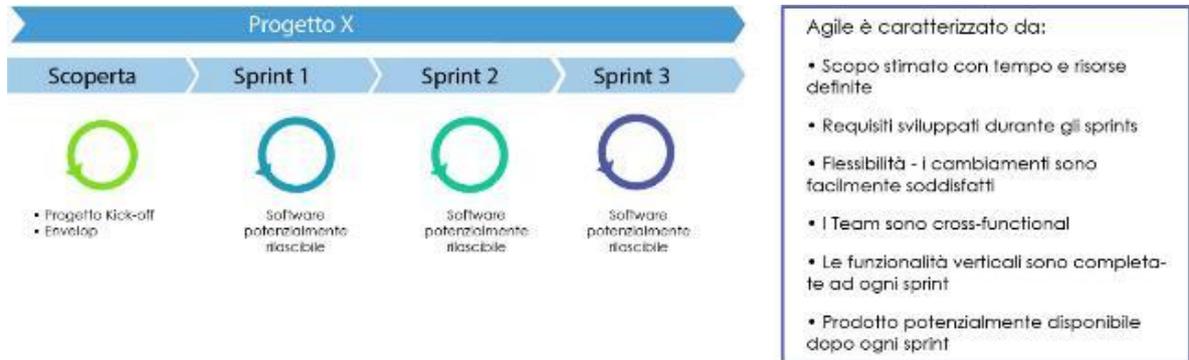


Figura 27. Agile Altea UP.

L'approccio Waterfall è invece lineare e sequenziale, che si basa su requisiti precisi e accurati e usa gli outputs di ciascuna fase per formare la base del lavoro per la fase successiva (Figura 28).

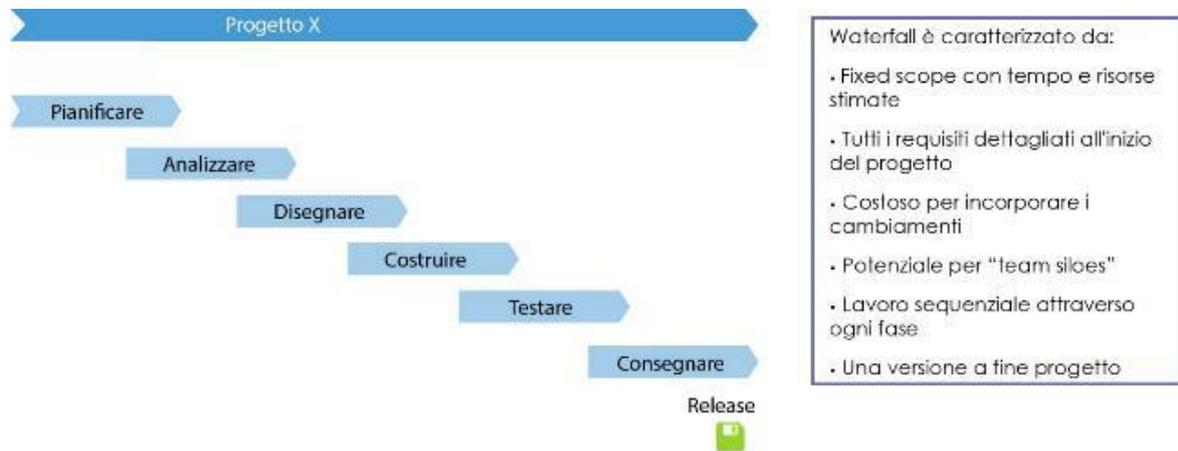


Figura 28. Waterfall Altea UP.

L'Hybrid Agile Framework di Altea UP combina i due metodi per trarre il meglio da entrambi, ricavandone i seguenti benefici (Figura 31):



Figura 29. Albero dei benefici dell'Hybrid Agile Framework.

- **I problemi sono identificati con largo anticipo**  
Le review effettuate alla fine di ogni sprint fanno emergere in anticipo ogni problema, che, di conseguenza, può essere affrontato immediatamente.
- **Capacità di accelerare la consegna di funzionalità a priorità elevata**  
L'intero progetto viene eseguito complessivamente su base prioritaria, considerando tutti i vincoli e le dipendenze.
- **Collaborazione rafforzata tra team**  
Il processo di focus continuo coinvolge tutti i team e rimuove eventuali silos.

- **Maggiore trasparenza e feedback**

Gli stakeholder sono coinvolti in ogni sprint e il progresso reale tangibile viene riesaminato regolarmente, attraverso un feedback continuo tra lo sviluppo ed il business.

- **Capacità di prototipare rapidamente le soluzioni**

Un focus su sprint più brevi con funzionalità definite e mirate consente ai diversi team di sviluppare prototipi per confermare i requisiti ogni qualvolta è necessario.

### 5.3.1 Caratteristiche dell'Hybrid Agile Framework

Le principali caratteristiche dell'Hybrid Agile Framework di Altea UP sono:

- Approccio modulare e flessibile, che applica tecniche agili allo sviluppo iterativo e ai cicli di test.
- Possibilità di organizzare il lavoro in sprint di funzionalità o di processo, invece che di fase.
- Definizione di "istanze" di consegna (deliverable) o di task nei piani di sprint.
- Priorità dei requisiti classificata in maniera dettagliata per evitare che ogni attività risulti urgente e strettamente necessaria.
- Approccio di pianificazione *wave waveing*, a ondate, che definisce e perfeziona i piani di sprint durante l'intero progetto, in quanto, con il procedere delle attività, tutti i dettagli e gli step successivi diventano più comprensibili.
- Frequenti dimostrazioni della soluzione software per il cliente e raccolta continua di feedback.
- Tracking costante dell'earned value durante gli sprint, utilizzando i trend di burndown.

Tipicamente, i lavori di progettazione e sviluppo sono condotti in sprint, con test effettuati contemporaneamente, nella misura possibile e praticabile. Una fase finale di test *end-to-end*

integrata è condotta dopo tutti gli sprint, al termine della quale è rilasciata una *release*. I progetti possono avere chiaramente più *release*.

Il piano principale è il Product Backlog che fornisce la tabella di marcia, ossia una visione ad alto livello di sprint, rilasci e funzionalità disponibili per release (*Figura 30*). Inoltre, raggruppa e conferma la priorità delle attività e le User Stories definite.

Product Backlog											
ID	Topic	Item : Description	Type	Priority	Status	Sprint	Owner Altea	Owner Client	Note Altea	Note Cliente	
PP003	Distinta Base	Definizione tipi di distinta base e impiego distinte	Attività di Progetto	Must	Open						
PP004	Distinte Base	Definizione dati di testata e dati di posizione	Attività di Progetto	Must	Open						
PP007	Distinta Base	Presentazione della distinta base materiale (radiatori, sedili e rubinetti); es. di struttura prodotto	Attività di Progetto	Must	Open						
PP009	Distinte Base	Definizione impostazioni a livello testata: impiego , stato , valori di default	Attività di Progetto	Should	Open						
PP_CO_00 1E	Calcolo Costi di produzione	Definizione per - la gestione del costo industriale (componenti, attività e Overhead) - gestione materiali consumo - politica di prezzo materiale (es. costo ottone per radiatore che subisce ampie variazioni di prezzo)	Attività Cliente	Must	Open						
PP_MM_0 01	Batterie con colori particolari	In qualità di Process Owner Logistica, ha richiesto l'impostazione dei blocchi alla creazione di ordini di produzione nel caso mancassero la distinta base e ciclo di produzione. vedere note	User Story	Must	Open				Per quanto riguarda i materiali che non hanno una distinta base e ciclo di produzione, non dovrebbe essere possibile poter creare ordini di produzione affinché non vengono create le anagrafiche mancanti.		

*Figura 30.* Esempio di Product Backlog.

Il piano di lavoro, definito Work Plan, elenca tutti i deliverable per istanze di consegna e l'effort totale. L'effort del Work Plan è dato dalla somma di tutte le istanze di consegna dei deliverable, la schedulazione include tutti gli sprint, mentre il monitoraggio a livello di progetto è a percentuale di completamento.

Le istanze di consegna dei deliverable sono pianificate in sprint e possono essere organizzate per processo, funzionalità o altri raggruppamenti. È creato un piano di sprint nel quale si elencano tutte le istanze in ambito per quel determinato *sprint*, raggruppate per oggetto di sviluppo, area funzionale o processo.

Il Design/Build/Test e l'Integration Test iniziale lavorano in stretta connessione durante lo sprint per aumentare da subito la qualità del deliverable. Le attività specifiche in ambito di esecuzione dello *sprint* dipendono ovviamente dal metodo utilizzato e dall'ambito specifico del progetto (*Figura 31*).

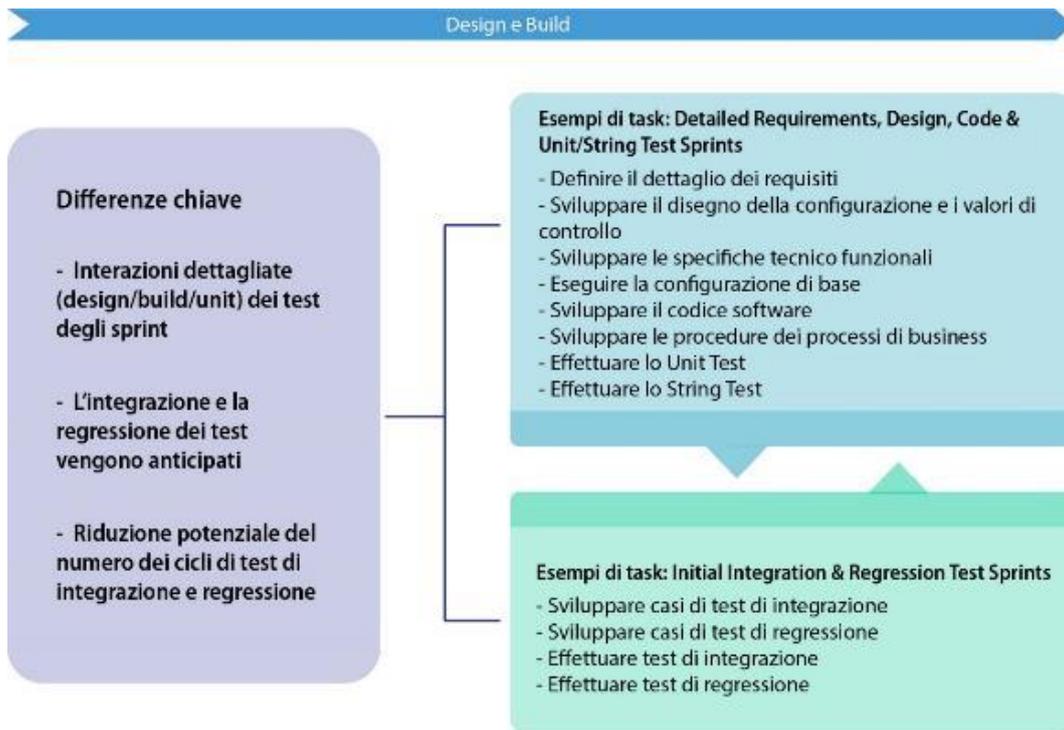


Figura 31. Analisi delle fasi di Design e Build.

Lo sprint viene gestito monitorando costantemente il Burndown Chart ed il Work Plan, che viene aggiornato con la percentuale di completamento alla fine di ogni sprint.

La velocità effettiva dei Team viene monitorata e utilizzata per aggiornare i successivi piani di sprint. Qualsiasi nuovo requisito che emerge dalla Sprint Review o da qualsiasi lavoro non completato viene restituito per ridefinirne la priorità (Figura 32).

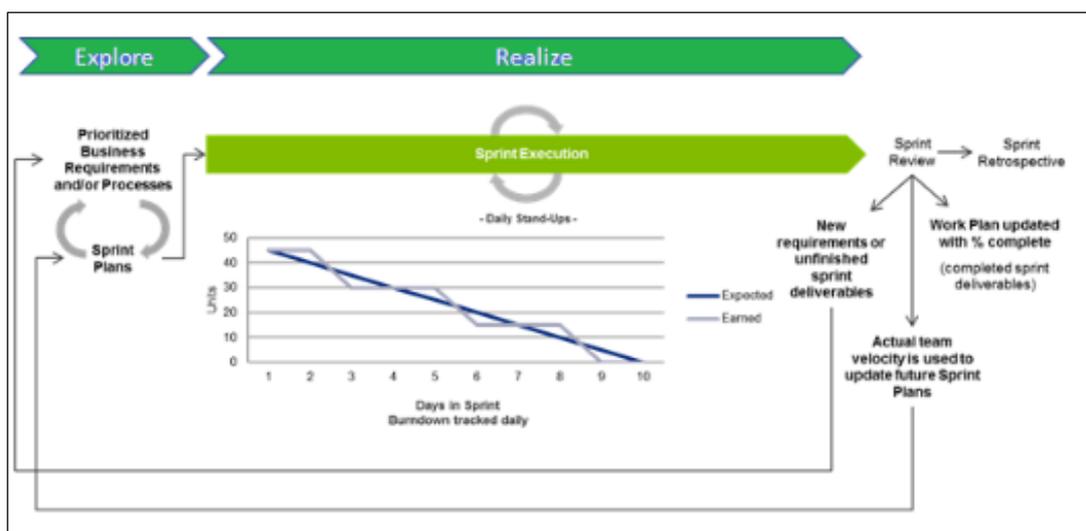


Figura 32. Gestione di uno sprint.

I progetti basati su questo framework, sono governati in buona parte utilizzando metodi, pratiche, strumenti e comportamenti appartenenti alla disciplina dell'Agile Project Management, in accordo con l'organizzazione cliente; per le fasi di Prepare, Explore e Realize, in particolare, è utilizzato il framework Scrum.

**APM (Agile Project Management):** si tratta di una metodologia fortemente orientata ai risultati, che si pone l'obiettivo di avviare, pianificare, eseguire, controllare e concludere progetti caratterizzati da un ciclo di vita change driven, cioè "adattivo", in quanto connotato da scenari di forte turbolenza della tecnologia e/o dei requisiti, con alto livello di modifiche e necessità di forte coinvolgimento degli stakeholder. Le metodologie adattive hanno carattere iterativo ed incrementale, con una durata breve (dalle due alle quattro settimane), e hanno il pregio di consentire piccoli miglioramenti incrementali che forniscono valore facilmente apprezzabile dagli stakeholder.

**Scrum:** il termine non è un acronimo, ma è derivato da una parola del linguaggio del rugby che indica il "pacchetto di mischia" e che evoca il contesto nel quale il Team "spinge" in modo coeso e vigoroso verso un'unica direzione; si tratta di un framework che mette a disposizione del progetto strumenti e tecniche dedicate alla suddivisione del lavoro in blocchi rapidi, cioè le iterazioni dette sprint, e alla loro esecuzione e controllo. Al termine di tali iterazioni, l'obiettivo è quello di fornire una "porzione" di prodotto funzionante e rispondente a requisiti di dettaglio, la cui priorità è variabile e dettata dalle mutabili caratteristiche dei fabbisogni degli stakeholder e dalla citata turbolenza dell'ambiente di progetto. L'organizzazione del lavoro effettuato in queste iterazioni è stabilita in incontri periodici che, con diverse gradazioni, coinvolgono tutti gli stakeholder. Nel framework Scrum, il progetto viene visto come una serie di task relativamente snelli, disegnati ed eseguiti in modo fortemente orientato alla domanda, secondo dei principi di adattamento che non sono necessariamente legati a una predeterminata pianificazione e a una sequenza iniziale.

La metodologia Scrum si focalizza maggiormente sull'obiettivo, ne riduce l'ambito agli aspetti essenziali e si dimostra particolarmente versatile di fronte al cambiamento delle condizioni del contesto in cui il progetto si trova. Tale metodologia propone un approccio alla gestione e alla governance del progetto che si basa fundamentalmente su tre punti:

- 1) Collaboriamo con il Cliente per dare priorità al valore di business.

- 2) Creiamo e consegniamo frequentemente e incrementalmente versioni funzionanti della soluzione per anticipare il feedback del Cliente.
- 3) Abbracciamo il cambiamento dei requisiti per ricercare una maggiore efficienza progettuale.

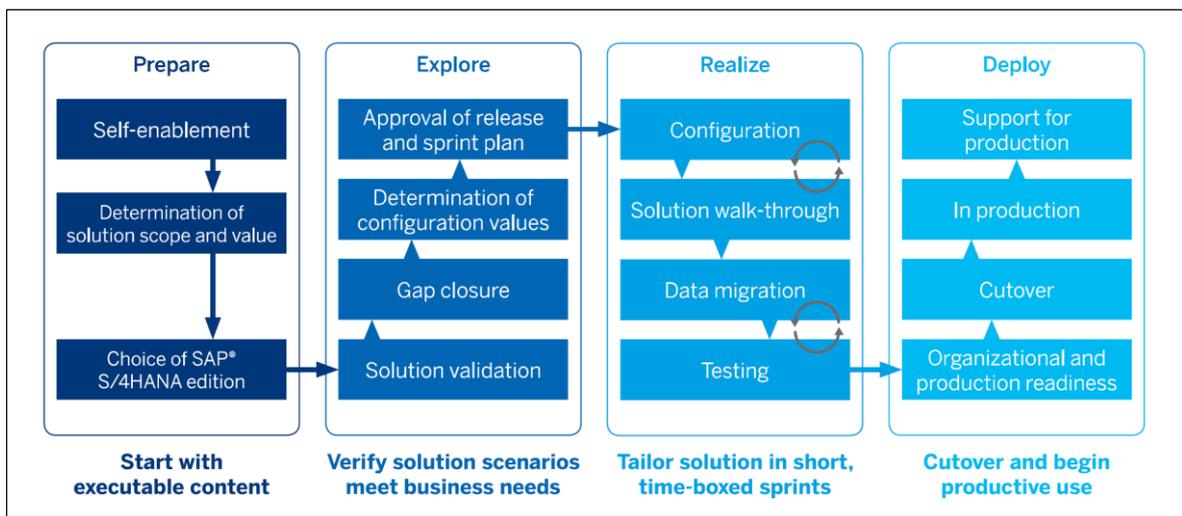
La pianificazione di dettaglio delle attività sarà adattiva e interattiva, nel rispetto di una timeline e di un piano di rilascio complessivo.

Attraverso tale approccio è possibile ridurre i rischi di progetto, grazie a una gestione dello stato di avanzamento basata su incrementi completi della soluzione complessiva e una gestione flessibile dei requisiti che massimizza, nel tempo, il valore per il cliente.

Le attività di test di sistema, che nella metodologia tradizionale vengono normalmente eseguite a ridosso della messa in esercizio, cioè del go-live, nella metodologia Scrum vengono distribuite per tutta la durata del progetto, in modo da ridurre i rischi di ritardi dovuti all'emergere di problemi in prossimità della conclusione del progetto.

### 5.3.2 Ciclo di vita del progetto secondo l'Hybrid di Altea UP

Riguardo al ciclo di vita del progetto, sarà caratterizzato dal seguente flusso (*Figura 33*):



*Figura 33.* Flusso attività Hybrid Agile Framework.

La descrizione di dettaglio delle fasi e il coinvolgimento degli stakeholder viene brevemente declinata come segue:

- **Prepare**

È la fase iniziale, durante la quale si pianifica il progetto, identificandone i confini, gli intenti e le caratteristiche del prodotto/servizio nell’ottica del business; viene inoltre definito il piano dei requisiti, cioè il Product Backlog, costituito da tutte le User Story e relative complessità. La fase consente di realizzare un piano operativo, definendo in dettaglio il coinvolgimento delle risorse del cliente e permettendo di dimensionare l’effort in funzione delle varie fasi di progetto.

- **Explore**

È la fase in cui, per ogni iterazione, la visione generale del progetto si declina in un piano di maggior dettaglio, approfondendo l’analisi del Product Backlog. Si mette in atto una fase di analisi fit/gap per consolidare e validare gli ambiti di dettaglio della soluzione. I gap identificati sono aggiunti al Backlog per l’utilizzo nella fase successiva. La fase di Explore prevede la redazione delle Functional Specification (FS) in sostituzione della Business Blue Print.

Le FS riportano i processi aziendali, le transazioni tecniche, il dettaglio operativo, i prerequisiti e le funzioni aziendali coinvolte negli step di processo. Quindi, ciascuna FS descrive un processo, riportando già tutte le fasi necessarie e la relativa configurazione del sistema (*Figura 34*).

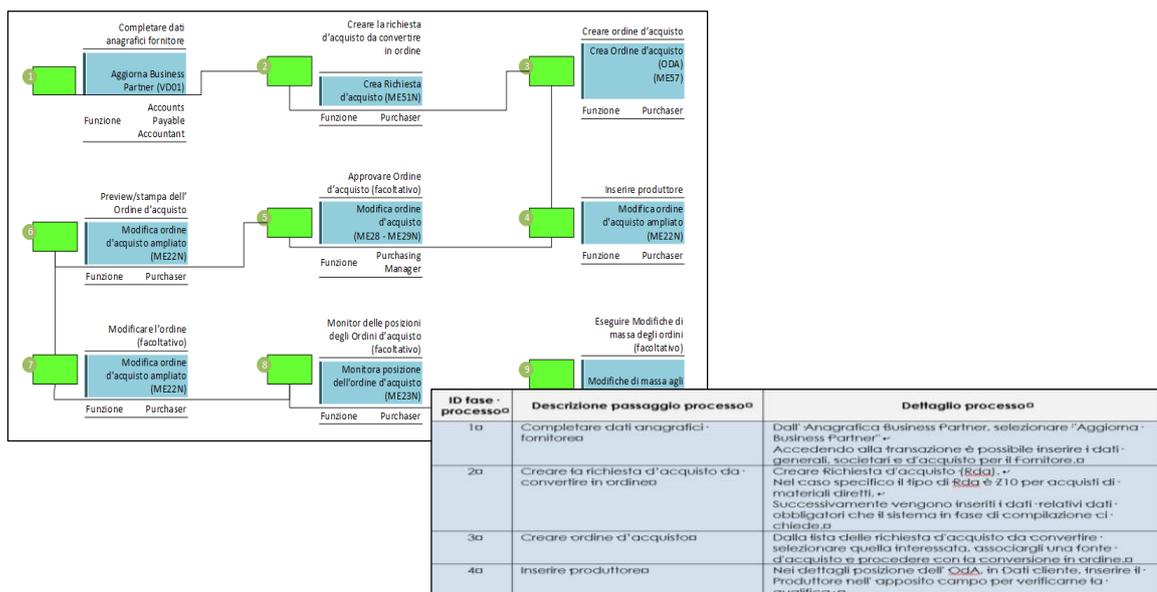


Figura 34. Esempio di FS.

- **Realize**

È la fase di esecuzione del progetto, durante la quale, per ogni iterazione, vengono realizzati i requisiti contenuti nel Product Backlog; vengono inoltre valutate le possibili alternative percorribili per soddisfare i requisiti del progetto. Per ogni iterazione, vengono valutati i risultati dell'esecuzione e le performance realizzate, al fine di adeguare i contenuti delle iterazioni successive. La Sprint Retrospective consente di avere un feedback immediato del modello disegnato e di poter gestire in modo tempestivo eventuali delta tra le aspettative degli utenti e il modello realizzato.

- **Deploy**

È la fase in cui il prodotto dei vari sprint viene portato in produzione e vengono effettuati i training e la data migration. Questa fase è governata con modalità classiche rispetto alle fasi precedenti. Inoltre, sono previste due fasi di test, per le quali saranno realizzate delle apposite schede utilizzate per documentare ogni prova e risultato (*Figura 35*):

- UAT per la verifica puntuali delle funzionalità;
- Integration Test per la verifica dell'intero processo.

The image shows two examples of test sheets. The top sheet is a form titled 'SCHEDA DI TEST' with a header 'ALTEA' and 'rev. 00'. It contains two main sections: 'Dati per esecuzione Test' and 'Partecipanti'. The 'Dati per esecuzione Test' section has columns for 'Dati', 'Data Value', and 'Notes', with rows for 'Società', 'Piani', and 'Organizzazione Acquisti'. The 'Partecipanti' section has columns for 'NOME COGNOME', 'RUOLO', and 'CUSTOMER'. The bottom sheet is a detailed test table titled 'SCHEDA DI TEST' with a header 'ALTEA' and 'rev. 00'. It has a sub-header 'DESCRIZIONE SCENARIO : Contabilità passiva'. The table has columns: 'DESCRIZIONE', 'STEP', 'FUNZIONE', 'AZIONE RICHIESTA', 'RISULTATO ATTESO', 'RISULTATO OTTENUTO', 'TESTER', 'RISULTATO OK / NON OK?', and 'DATA TEST'. It contains 7 rows of test cases with various statuses like 'OK' and 'KO -> da verificare'. At the bottom, there are tabs for 'Descrizione scenario', 'Fatturazione passiva', 'Interfacce', 'Fatturazione attiva', 'Contabilità generale', 'Contabilità cespiti', and 'Incassi e pagamenti'.

Figura 35. Esempio schede di test.

Riguardo all'attività di Data Migration, in questa fase viene preparato sia l'ambiente per le attività di Test, sia quello finale da utilizzarsi dopo il Go Live (ambiente SAP Production). Le attività di Data Migration permettono di simulare in maniera completa l'ambiente effettivo che il Cliente si troverà ad utilizzare e di eseguire test completi di tutte le funzionalità implementate per certificare la coerenza dell'implementazione di SAP nel nuovo sistema del Cliente. In questo ambito rientrano le attività che portano alla migrazione nel nuovo ambiente operativo delle anagrafiche e dei dati transazionali rilevanti sul nuovo database gestionale, mentre per i dati transazionali storici o statistici, chiusi o storicizzati relativi all'anno in corso e/o ad anni precedenti, non è previsto il caricamento.

Le attività, svolte da personale dedicato, consistono nella definizione delle modalità di migrazione dei dati e nella definizione delle necessarie procedure manuali o informatiche per il recupero dei dati dal precedente sistema e il passaggio al database di SAP. Tali procedure devono essere ripetibili e consentire verifiche degli errori presenti nei dati di partenza. La migrazione dei dati tramite programmi di migrazione avviene attraverso tools standard SAP, come LTMC e LSMW, con l'adattamento dei tracciati quando necessario per trasferire anche i campi Custom eventualmente presenti nel sistema di origine.

Altea Up fornisce al Cliente i tracciati di destinazione dei dati (file Excel) (*Figura 36 e Figura 37*) su cui poter “parcheggiare” i dati che saranno estratti dai sistemi di origine e per i quali sarà stata definita una gestione non manuale.

Lista campi per l'oggetto di migrazione: Fornitore										
Nome foglio	Nome del gruppo	Descrizione campo	Importanza	Tipo	Lunghezza	Decimale	Struttura SAP	Campo SAP	Valori Lo	Regole di compilazione
Key	Vendor		obbligatorio per il foglio	Testo	60		S_SUPPL_GEN	LIFNR		
	BP Group	BP Grouping	obbligatorio per il foglio	Testo	60		S_SUPPL_GEN	BU_GROUP		
	Account Group	Supplier Account Group		Testo	60		S_SUPPL_GEN	KTOXX		
	Supplier	Name		Testo	40		S_SUPPL_GEN	NAME_FIRST		
	Organization	Name 2		Testo	40		S_SUPPL_GEN	NAME_LAST		
		Name 3		Testo	40		S_SUPPL_GEN	NAME3		
		Name 4		Testo	40		S_SUPPL_GEN	NAME4		
	Supplier Person	Title		Testo	60		S_SUPPL_GEN	TITEL		
		Academic Title		Testo	60		S_SUPPL_GEN	TITLE_ACA1		
		First name		Testo	40		S_SUPPL_GEN	NAME_FIRST_P		
		Last name		Testo	40		S_SUPPL_GEN	NAME_LAST_P		

Figura 36. Esempio di template di migrazione.



- Partite aperte Co.Ge
- Saldi conti Co.Ge
- Giacenze di magazzino
- Oggetti di Controlling (Centri di Costo e Gerarchie, Centri di Profitto e Gerarchie, Ordini Interni, Attività).

### 5.3.3 Ruoli nel progetto Hybrid

Agile Project Management e Scrum sono caratterizzati da un forte pragmatismo, che si riversa non solo nelle pratiche agili, ma anche nella organizzazione delle risorse coinvolte. Di seguito si esaminano i ruoli previsti nel progetto (*Figura 38*):

#### **Scrum Team**

- **Develop Team**

È costituito dagli Specialisti Altea Up che avranno il compito di sviluppare le funzionalità previste nel Product Backlog in stretta collaborazione con il Develop Team del Cliente.

- **Scrum Master**

È la figura chiave di Altea Up: è il garante della metodologia utilizzata, e ha la responsabilità di assicurarsi che principi, pratiche e valori siano compresi, approvati e seguiti; ha inoltre il compito di indirizzare il Product Owner rispetto alla definizione del contenuto del Product Backlog, in particolare riguardo alle User Stories e priorità, e di facilitare le attività di tutti gli stakeholder che non fanno parte dello Scrum Team.

- **Product Owner**

È la figura chiave per il Cliente poiché ha la responsabilità del contenuto del Product Backlog e delle sue priorità, ed è il punto di contatto unico tra Scrum Master/Develop Team e il business del Cliente.

## Stakeholder

- **Steering Committee**

È costituito dallo Sponsor di Progetto e dal Management del Cliente, dal Sales Manager di Altea Up, dallo Scrum Master e dal Product Owner; ha il compito di verificare la disponibilità delle Risorse del Progetto, e di valutarne Rischi e Performance.

- **Process Owner/Key User**

Si tratta delle risorse del Cliente dedicate a fornire referenze funzionali e di processo per ogni macro-area coinvolta nel Progetto.

La metodologia, quindi, prevede una struttura delle risorse di progetto ben definita e chiara, con ruoli strategici, di controllo ed operativi a vari livelli.

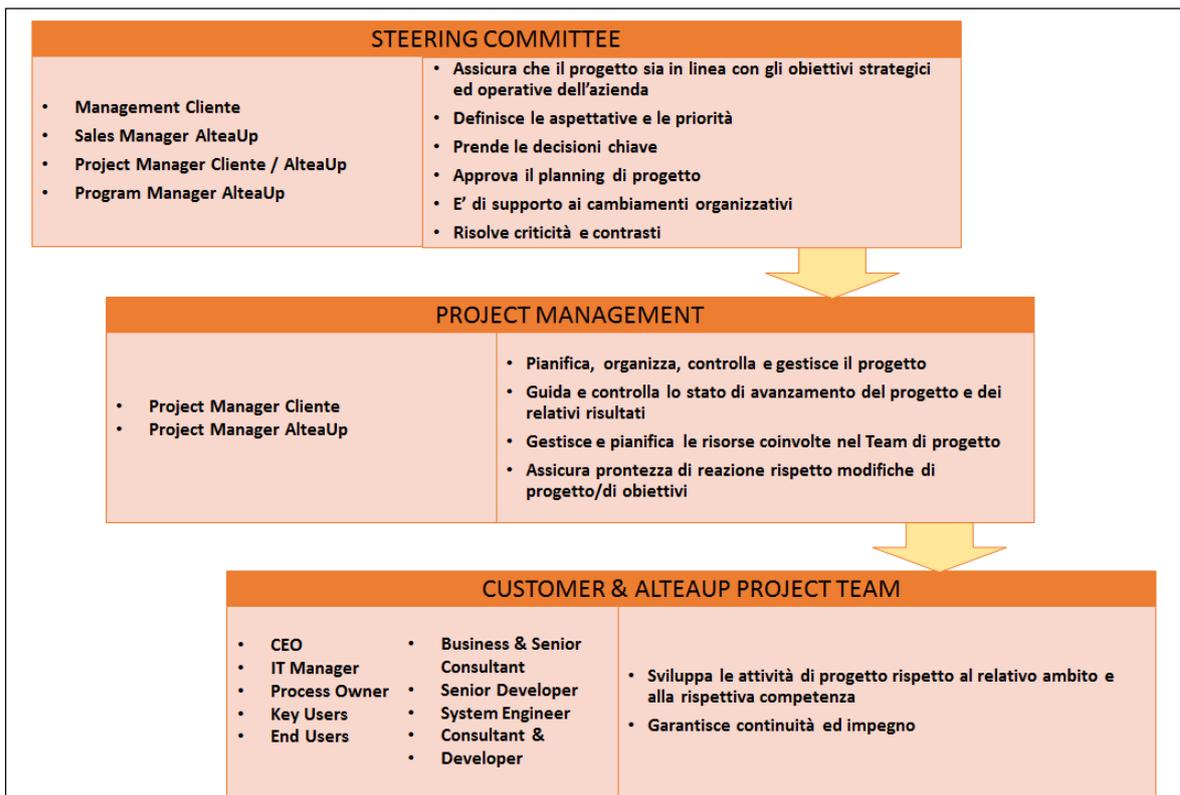


Figura 38. Organizzazione e ruoli di un progetto.

#### 5.3.4 *Principi di governo della dimensione temporale ed economica*

L'intero impianto metodologico, oltre a garantire le migliori condizioni di collaborazione tra le parti e una miglior comprensione tra il linguaggio del “business” e quello della “tecnologia”, consente, a differenza delle metodologie classiche, di dominare le due dimensioni in oggetto con una modalità fortemente anticipatoria di eventuali crisi. Queste ultime solitamente si riscontrano solo nelle fasi finali di progetti IT che si trovano a operare in ambienti ad alta complessità e ad alta incertezza.

Infatti, la possibilità di definire e aggiornare la Target Velocity e di confrontarla sistematicamente con quella Actual al termine di ogni sessione di sprint, permette, dopo un numero minimo di tre iterazioni di sprint, di acquisire consapevolezza circa la reale capacità di generare output da parte di tutti i team coinvolti e conseguentemente di intraprendere nell'immediato tutte le azioni di mitigazioni volte alla convergenza tra Target e Actual Velocity, imponendo quindi ai team nuovi target di efficienza che consentano di minimizzare gli impatti economici e temporali.

La consapevolezza continua circa le reali velocità di esecuzione e di svuotamento del Product Backlog consente di proiettare in avanti le date di close delle singole waves e di poter quindi valutare con largo anticipo la necessità di:

- A) Accelerare
- B) Parallelizzare ulteriormente
- C) Riprogrammare il piano temporale.

Quindi, permette di determinare con largo anticipo l'entità delle variabili economiche in gioco, consentendo allo Steering Committee, analizzando la lista delle User Story e corrispondenti Story Point, di poter rimodulare entrambe le dimensioni, quella “tempo” e quella “economica”.

Tutta la letteratura scientifica sostiene che questa nuova modalità per governare progetti ad alta complessità e alta incertezza sia in assoluto la migliore per raggiungere gli obiettivi prefissati.

### 5.3.5 Componenti della metodologia tradizionale

Restano escluse dalle iterazioni alcune attività, che di fatto non sono tracciate nei requisiti e dunque non sono presenti nel Product Backlog, ma che saranno invece gestite con approccio metodologico tradizionale (Waterfall):

- **Training**  
Attività di formazione orientate al trasferimento delle conoscenze relative all'utilizzo del servizio alle risorse del Cliente.
- **Data Migration/Final Migration**  
Attività di predisposizione degli ambienti, di creazione dei programmi di caricamento, di supporto al caricamento di un primo set di dati significativi validati dal cliente.
- **Close**
- **Supporto Post Go-Live**  
Ritocchi e supporto per il corretto funzionamento del servizio "a regime".

### 5.3.6 Tipologie di test

Di seguito si elencano le tipologie di Test previste nella metodologia:

- **Unit Test:** test delle singole funzionalità, in carico ad Altea Up;
- **User Acceptant Test:** test su singoli processi in modalità agile in carico al Cliente;
- **Integration Test:** test su processi *end-to-end* in modalità agile e integrazioni dei sottosistemi in carico al Cliente.

### 5.3.7 Reporting real-time

Un importante punto di forza della soluzione S/4HANA, implementata da Altea UP nel suo Hybrid Agile Framework, è il fatto che racchiude, in un unico ambiente, sia il sistema transazionale che quello analitico, grazie alla potenza del database in-memory HANA. Tale architettura abilita il reporting real-time a supporto dei processi decisionali, a tutti i livelli dell'organizzazione.

L'applicativo utilizza la user interface (layout schermate) di SAP chiamata FIORI:

"SAP FIORI" è una nuova interfaccia web lato utente sviluppata da SAP che si contraddistingue per la versatilità e la facilità d'uso; inoltre, è progettata per funzionare su tutti i tipi di dispositivi (desktop, tablet, smartphone) (Figura 39).

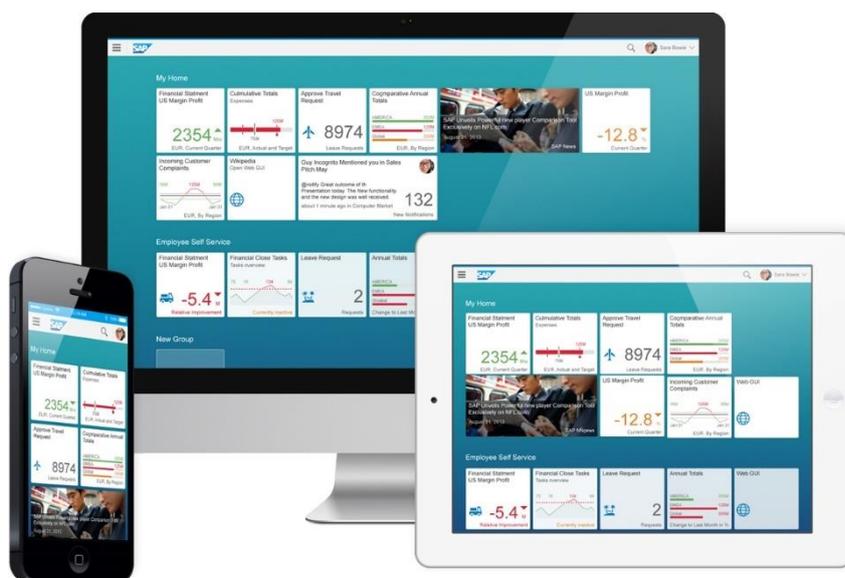


Figura 39. SAP FIORI.

Inoltre, la libreria APP FIORI rende disponibile una serie di APP pre-configurate a disposizione di ogni utente. (Figura 40).

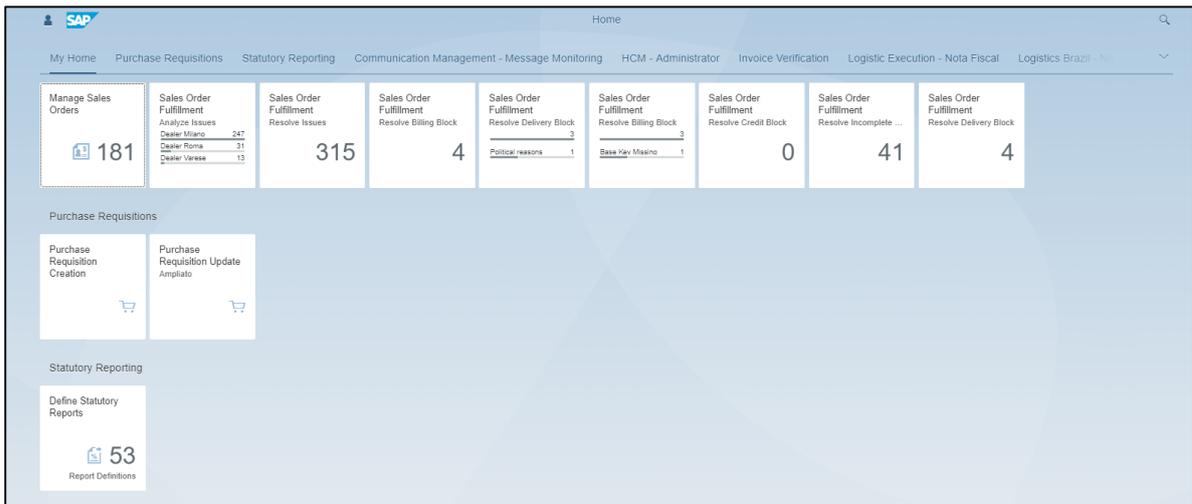


Figura 40. SAP FIORI.

L'utente è guidato nelle attività quotidiane attraverso un sistema "role-based" (basato su profili autorizzativi) che permette l'utilizzo delle sole APP necessarie al suo lavoro.

In quest'ottica è prevista in ambito di progetto l'attivazione delle sole APP individuate come necessarie al corretto funzionamento del sistema e al completamento dei diversi processi end-to-end individuati e verificati con il supporto delle best-practice SAP.

Un altro strumento utilizzato per facilitare la comunicazione tra tutti gli stakeholders del progetto è lo Sharepoint, un repository di documenti ufficiali, liste e calendari riunioni. Il sito è organizzato in modo tale da rispecchiare la suddivisione delle fasi del progetto: in questo modo anche il Cliente è sempre a conoscenza dello stato di avanzamento delle attività (Figura 41).

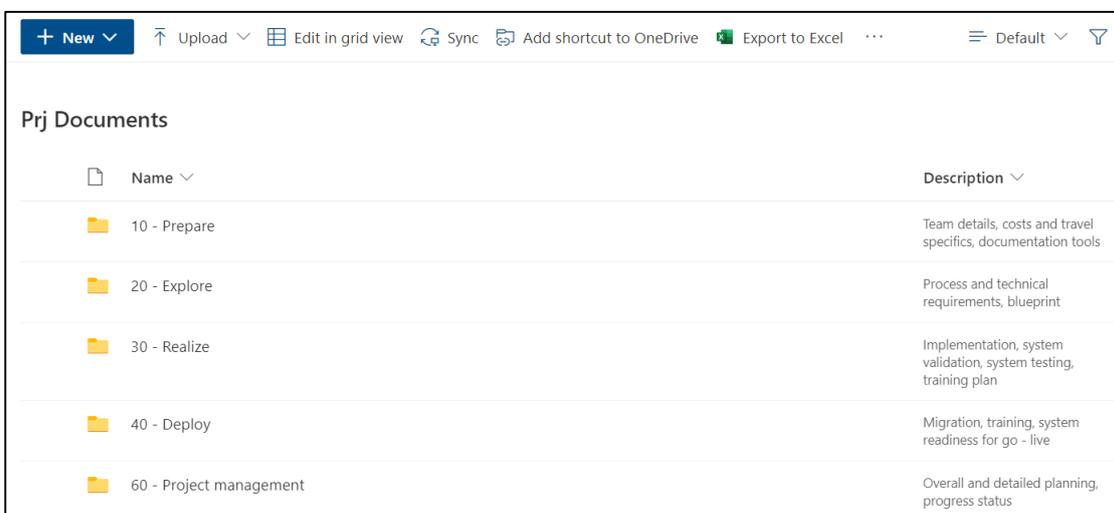


Figura 41. Esempio di Sharepoint.

La comunicazione tramite il sito di progetto è di tipo “pull” e presuppone che gli stakeholders si colleghino autonomamente. Nelle sezioni pubbliche, aperte a tutti gli stakeholders, devono essere presenti soltanto i documenti ufficiali, approvati e rilasciati, mentre non devono essere presenti bozze o documenti non previsti come deliverables di progetto.

Lo Sharepoint nella sezione Issue List dovrà riportare tutti i punti aperti, discussi nelle diverse sessioni o comunicati dal cliente o dai key user. La Issue list raccoglie le seguenti casistiche:

- Richieste del cliente/key user in merito a funzionalità non rilevate nelle deliverables consegnate;
- Richieste di funzionalità non previste dalla soluzione standard proposta;
- Errori di sistema (bug);
- Errori di configurazione o documentazione.

La sezione Gap List dovrà riportare i GAP emersi e approvati.

Il Project Calendar sarà utilizzato come calendario degli incontri con il Cliente. Per ciascun incontro pianificato dovranno essere indicati l’argomento, le risorse coinvolte, l’orario di inizio e fine.

## Bibliografia e sitografia

- Materiali forniti da Altea Up.
- Appunti di corsi universitari.
- *Diagramma di PERT e di GANTT. Strumenti per la gestione/controllo dei progetti*, <https://www.unieda.it/modulo-e4-improving>.
- Boin Cristian, *Project management – 7 consigli utili a tutti*, in “Cristian Boin – Manager in affitto”, <https://cristianboin.com/consigli-di-project-management/>.
- Fabris Virginia, *Come impostare obiettivi SMART? Il metodo per raggiungere la vetta*, in “Appvizer”, <https://www.appvizer.it/rivista/organizzazione/project-management/obiettivi-smart>.
- M. Stefano, *Un esempio di Work Breakdown Structure di progetto*, in “TeraNet PPM”, <https://www.t-ppm.it/esempio-wbs-di-progetto-software>.
- Lavecchia Vito, *Che cos'è l'OBS (Organization Breakdown Structure) e la matrice delle responsabilità*, in “Informatica e ingegneria online”, <https://vitolavecchia.altervista.org/che-cose-lobb-organization-breakdown-structure-e-la-matrice-delle-responsabilita/>.
- *Resource Breakdown Structure*, <http://resourcebreakdownstructure.blogspot.com/2013/06/resource-breakdown-structure.html>.
- *Cost Breakdown Structure Template*, in “Visual Paradigm Online”, <https://online.visual-paradigm.com/de/diagrams/templates/work-breakdown-structure/cost-breakdown-structure-template/#>.
- Royce Winston W., *Managing the development of large software systems* (1970).
- Ragazzi Massimo, *Tutte le differenze tra Agile Project Management e il PM tradizionale*, in “Agile School”, <https://www.agile-school.com/blog/agile-project-management-vs-pm-waterfall>.
- McCormick Mike, *Waterfall vs Agile Methodology* (2012).
- *Metodologia Agile e PMBOK: Waterfall e metodo iterativo*, in “Humanware – Project Media Center”, <https://www.humanwareonline.com/project-management/center/pmbok-agile/>.

- *Extreme Chaos*, The Standish Group,  
[https://www.cin.ufpe.br/~gmp/docs/papers/extreme\\_chaos2001.pdf](https://www.cin.ufpe.br/~gmp/docs/papers/extreme_chaos2001.pdf).
- *Manifesto per lo sviluppo Agile Software*,  
<https://agilemanifesto.org/iso/it/manifesto.html>.
- *Agile Alliance*, <https://www.agilealliance.org/>.
- Mazzucco Marco, *Agile transformation: come cambia il modo di lavorare per affrontare la trasformazione digitale*, <https://www.zerounoweb.it/cio-innovation/metodologie/agile-transformation-come-cambia-il-modo-di-lavorare-per-affrontare-la-trasformazione-digitale/>.
- Doll Jason, *Effective Agile sprint cycles for data consulting projects*, in “Neal Analytics”, <https://nealanalytics.com/blog/effective-agile-sprint-cycles-for-data-consulting-projects/>.
- *Scrum: come funziona la metodologia Agile più diffusa*, in “Mònade”,  
<https://monade.io/it/scrum-come-funziona-la-metodologia-agile-piu-diffusa/>.
- Bruins Hans, *Naked Scrums*, <https://medium.com/@hans.bruins/naked-scrum-81218f6f4833>.
- Hirotaka Takeuchi, Nonaka Ikujiro, *The New Product Development Game*, in “Harvard Business Review”, gennaio-febbraio 1986.
- Federicis Daniele, *Change Management & Project Management*, in “Il blog di Management”, <https://www.blog-management.it/2018/04/10/change-management-project-management/>
- Salameh H., *What, When, Why, and How? A Comparison between Agile Project Management and Traditional Project Management Methods*, in “International Journal of Business and Management Review” (2014), v. 2, n. 5, pp. 52-74.
- Menecali Mirko, *Approccio Agile al Project Management: meglio del Waterfall?*, in “Sinfo online”, <https://blog.sinfo-one.it/waterfall-e-agile-due-approcci-contrapposti-al-project-management>.
- *What is Hybrid Agile, Anyway?*, <https://www.agilealliance.org/what-is-hybrid-agile-anyway/>.

## Ringraziamenti

A conclusione di questo lavoro di tesi, che giunge come coronamento di un percorso di studi, non mi resta che ringraziare tutti coloro che hanno contribuito al raggiungimento di questo mio traguardo, che mi ha permesso di maturare dal punto di vista professionale e umano. Vorrei ringraziare in primo luogo tutta la mia famiglia, in particolare i miei genitori, mia zia e le mie nonne che hanno sempre creduto in me durante questi anni, con il loro dolce e instancabile sostegno mi hanno appoggiato in ogni mia scelta, condividendo con me gioie e sacrifici, successi e difficoltà, sempre pronti a spronarmi a mostrare il meglio di me e superare i miei limiti. Ringrazio la mia fidanzata Giulia per avermi trasmesso la sua immensa forza e per esserci stata sempre, capace di capirmi e sostenermi in tutti i momenti difficili. Ringrazio tutti coloro che sono entrati a far parte della mia vita con l'appellativo di colleghi di corso o di università, ma che adesso ne fanno parte in veste di amici, e che attraverso un sostegno reciproco e sincero hanno reso questo traguardo ancora più prezioso. Un altro ringraziamento sentito va agli amici di lunga data, che nonostante la distanza e le diverse strade intraprese fanno ancora parte della mia vita, supportandomi e sopportandomi in questi anni. Un ringraziamento particolare va alla Prof.ssa Giulia Bruno, per aver accettato l'incarico di relatore per la mia tesi di laurea, per il supporto fornitomi durante la stesura e per la disponibilità dimostrata. Ringrazio l'azienda Altea Up, presso cui ho svolto un tirocinio formativo di tre mesi durante il quale ho avuto modo di conoscerne ed apprezzarne la realtà; ho proseguito la collaborazione per la stesura della tesi e da poco sono stato assunto in qualità di Junior Consultant SAP. Sono grato per le competenze acquisite e per il tempo che mi è stato dedicato. Infine, rivolgo un augurio a me stesso, affinché possa continuare ad avere la determinazione e la costanza che mi contraddistinguono per poter realizzare i miei sogni e inseguire i miei obiettivi.