

Politecnico di Torino

Corso di Laurea

A.a. 2022/2023

Sessione di Laurea Marzo 2023



Politecnico di Torino

Analisi della metodologia di Project Management ibrida utilizzata nell'azienda Coolshop S.R.L.

Relatori Accademico:

Relatore Aziendale:

Candidato:

Giovanni Zenezini

Roberto Gavazov

Marcello Zampella

1. INTRODUZIONE	4
1.1. OBIETTIVO DELLA TESI	4
1.2. LA GESTIONE DEL PROGETTO	4
2. METODOLOGIE TRADIZIONALI	8
2.1. METODOLOGIA WATERFALL	8
2.1.1. CARATTERISTICHE	8
2.1.2. STRUMENTI	8
2.1.2.1. PIANIFICAZIONE	9
2.1.2.2. PROGRAMMAZIONE	12
2.1.3. FASI	14
2.1.3.1. RACCOLTA DEI REQUISITI	15
2.1.3.2. DESIGN	15
2.1.3.3. IMPLEMENTAZIONE	16
2.1.3.4. TEST	16
2.1.3.5. RILASCIO	16
2.1.3.6. MANUTENZIONE	16
2.1.4. APPLICAZIONI	17
2.1.5. VANTAGGI E SVANTAGGI	17
2.2. METODOLOGIA AGILE	18
2.2.1. IL MANIFESTO	19
2.2.2. APPLICAZIONI	22
2.2.3. FRAMEWORK	22
2.2.4. SCRUM	23
2.2.4.1. SCRUM TEAM	24
2.2.4.2. ARTEFATTI	26
2.2.4.3. EVENTI	29
2.2.5. KANBAN	30
2.2.5.1. PRINCIPI	31
2.2.5.2. PRATICHE	31
2.2.6. VANTAGGI E SVANTAGGI	33
2.3. METODOLOGIA HYBRID	34
2.3.1. METODOLOGIE TRADIZIONALI - CONFRONTO	34
2.3.2. GESTIONE IBRIDA	36
3. METODOLOGIA COOLSHOP	37
3.1. DESCRIZIONE AZIENDA COOLSHOP	37
3.2. NASCITA METODOLOGIA	38
3.3. DESCRIZIONE DEL PROCESSO LAVORATIVO	39
3.3.1. DISCOVERY	40
3.3.2. DEFINIZIONE DEL PROGETTO	40
3.3.3. PIANIFICAZIONE DI ALTO LIVELLO	41
3.3.4. WAVE	41
3.3.5. RELEASE	42
3.3.6. SPRINT	43
3.4. STRUMENTI AZIENDALI	46

4. CASO DI STUDIO - PROGETTO PARTNER PORTAL	47
4.1. DESCRIZIONE AZIENDA CLIENTE CMCO	47
4.2. DESCRIZIONE DEL PROGETTO	48
4.2.1. DEFINIZIONE DEL PROGETTO	49
4.2.2. ESECUZIONE DEL PROGETTO	51
4.2.3. RISULTATI	55
4.2.4. UTILIZZO DELLA METODOLOGIA DI PROJECT MANAGEMENT	55
5. APPROCCIO E METODO DI RICERCA	59
5.1. INTERVISTA CON PALADIN/PM - da discovery a Release 4	59
5.1.1. Tematica 1: Passaggio di Consegna	59
5.1.2. Tematica 2: Budget di Progetto	60
5.1.3. Tematica 3: Turnover dei Membri del Team	61
5.1.4. Tematica 4: Gestione del Team	61
5.1.5. Tematica 5: Organizzazione delle Release	62
5.2. INTERVISTA CON PALADIN/PM - da Release 5 a chiusura progetto	62
5.2.1. Tematica 1: Gestione del Team	62
5.2.2. Tematica 2: Integrazione CMS	63
5.2.3. Tematica 3: Passaggio di Consegna	63
5.2.4. Tematica 4: Consegna del Progetto	64
5.3. INTERVISTA CON HERO/TECHNICAL LEADER	64
5.3.1. Tematica 1: Gestione del Team	65
5.3.2. Tematica 2: Ingolfamento della Risorsa	65
5.3.3. Tematica 3: Budget di Progetto	66
5.4. INTERVISTA CON DEVELOPER	66
5.4.1. Tematica 1: Comunicazione Interna al Team	66
5.4.2. Tematica 2: Rapporto con Technical Leader	67
5.5. INTERVISTA CON JEDI/UX EXPERT	67
5.5.1. Tematica 1: Gestione del Team	68
5.5.2. Tematica 2: Allocazione sui Progetti	68
5.5.3. Tematica 3: Fase di Gara	68
5.5.4. Tematica 4: Fase di Workshop	69
6. RISULTATI	70
6.1. APPLICAZIONI	70
6.2. VANTAGGI	72
6.3. SVANTAGGI	73
7. CONCLUSIONI, PROSPETTIVE FUTURE E RIFLESSIONI	74

1. INTRODUZIONE

1.1. OBIETTIVO DELLA TESI

Negli ultimi anni si è registrata una crescente tendenza all'utilizzo di metodologie di gestione dei progetti ibride, attraverso le quali le aziende combinano le caratteristiche delle metodologie tradizionali Waterfall e Agile. Ciò accade perchè i Project Manager devono rispondere ad esigenze della propria azienda e del cliente, che variano da progetto a progetto, quindi risulta difficile definire uno standard che possa apportare benefici a tutti e si preferisce adottare soluzioni personalizzate allo specifico progetto.

Proprio per questa ragione, i Project Manager dell'azienda Coolshop s.r.l., basandosi sulle caratteristiche dell'azienda e sui progetti tipici commissionati, hanno ideato una metodologia ibrida, divenuta successivamente lo standard a cui fare riferimento all'interno dell'organizzazione.

Il presente lavoro di tesi è nato dal mio rapporto con l'azienda Coolshop, iniziato con un tirocinio curriculare come analista funzionale.

Dopo aver sperimentato la metodologia e constatato i benefici apportati ai progetti, ho ipotizzato che potesse essere altrettanto efficace per progetti con caratteristiche simili. Lo scopo di questa tesi è quindi quello di validare questa metodologia, definendo in quali contesti può apportare beneficio e quali sono i vincoli del suo utilizzo.

Con il fine di spiegare nel dettaglio il funzionamento della metodologia e per quali ragioni è caratterizzata da determinati elementi, nei capitoli successivi saranno illustrate nel dettaglio le metodologie tradizionali Waterfall e Agile.

Successivamente, sarà spiegato la metodologia di Project Management usato in azienda e sarà usato il progetto "Partner Portal" come caso di studio, analizzandone le caratteristiche e indagando le correlazioni tra le specifiche del progetto e i benefici che la metodologia ha apportato.

1.2. LA GESTIONE DEL PROGETTO

I progetti sono uno strumento utile al fine di apportare valore all'organizzazione. Rispetto agli altri strumenti esistenti, i progetti presentano i seguenti vantaggi:

- **Focalizzazione**
Un progetto è un'organizzazione temporanea creata per un unico obiettivo: apportare valore.
- **Controllo**
Ogni progetto è soggetto a vincoli predefiniti, come scadenze, budget e standard di qualità.
- **Specializzazione**
Il project management è diventato una professione, che prevede buone prassi, strumenti, metodi e schemi di certificazione.

(ICB4 - IPMA Italy. *Individual Competence Baseline for Project Management*, 2017)

I progetti aziendali nascono da opportunità identificate, valutate e documentate da parte dell'organizzazione; parte di queste opportunità sono ulteriormente sviluppate in uno specifico case business e possono risolversi in uno o più progetti che forniscono output rilasciabili, definiti "deliverables" (ISO).

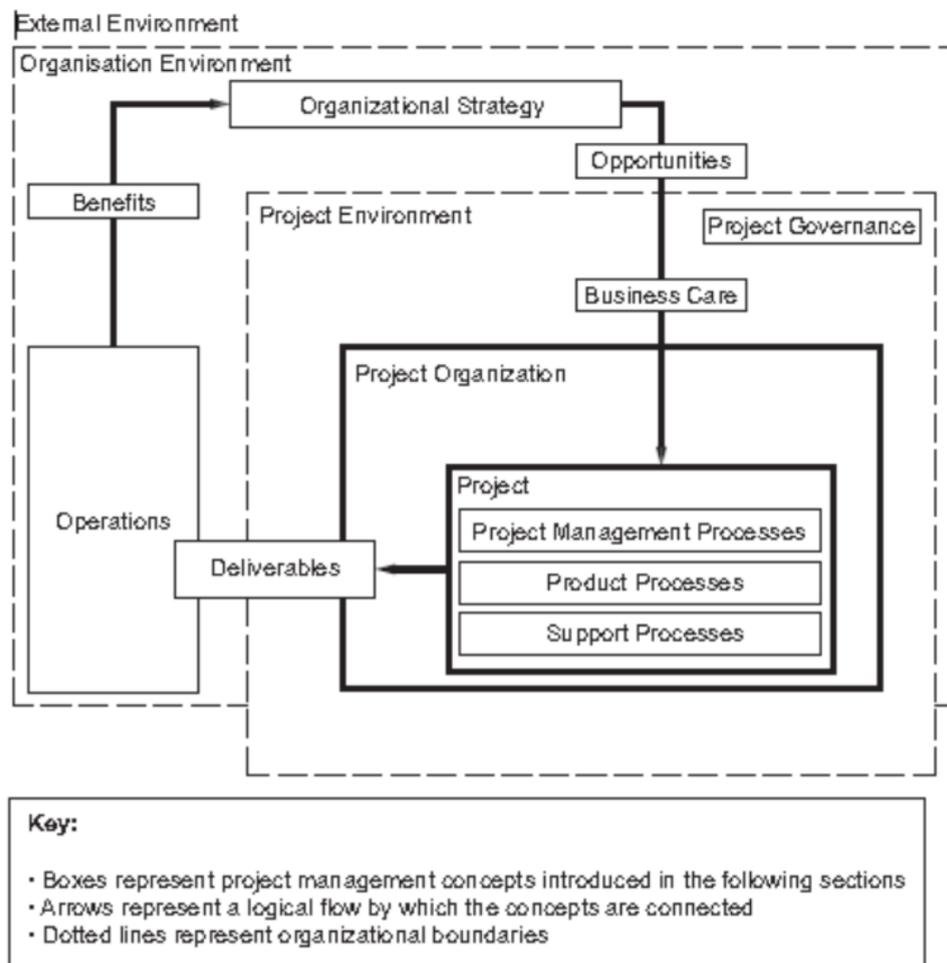


Figure 1: Organigramma

Un progetto, dunque, è un'impresa atta a creare un prodotto, un servizio o un risultato unico. Caratteristica essenziale del progetto è la durata temporanea, con un inizio e una fine definiti; la fine del progetto è determinata da particolari condizioni che possono assumere i suoi obiettivi, i quali possono diventare:

- raggiunti
- irraggiungibili
- non più determinanti, quindi il cliente desidera terminare il progetto

A differenza del progetto, il prodotto risultante tende ad essere duraturo nel tempo. Anche se in alcuni prodotti e attività del progetto possono essere presenti elementi ripetitivi, questa ripetizione non cambia le caratteristiche fondamentali e uniche del lavoro del progetto (*A Guide to the Project Management Body of Knowledge*).

Ad esempio, due siti di e-commerce possono sfruttare tecnologie simili; tuttavia, ogni sito rimane unico, con un diverso design, diversi data model, diversi interessi e ambiti...

Il project management è l'applicazione ad un progetto di metodi, strumenti, tecniche e competenze.

Un project management efficace porta molti benefici all'organizzazione e agli stakeholder, aumenta le probabilità di raggiungere gli obiettivi e assicura un utilizzo efficiente delle risorse, soddisfacendo le diverse esigenze degli stakeholder del progetto

Il ruolo del Project Manager è quello di prendere decisioni, pianificare e controllare l'incertezza per portare il progetto il più possibile in linea con i costi, i tempi e la qualità previsti, bilanciando quello che viene definito "triplice vincolo"; difatti, variazioni ad un lato del triangolo hanno ripercussioni sugli altri due, determinando diverse configurazioni qualitative del prodotto risultante.



Figure 2: Triplice Vincolo

Non solo, un buon PM deve essere in grado di indirizzare le persone coinvolte nel progetto, le quali possono essere comunità di utenti, parti interessate esterne e interne, committenti, team di progetto interno e manodopera, nonché progettisti esterni, subappaltatori, fornitori e consulenti.

Proprio per questo un PM non deve saper solo pianificare, programmare e controllare, ma deve possedere qualità di leadership, conoscenza dei comportamenti umani, psicologia, capacità di comunicazione e conoscenze multidisciplinari.

Quando si tratta di parti esterne e parti interessate, che spesso sono fondamentali per il successo di un progetto, il PM dovrebbe trarre vantaggio dall'aggiunta di buone pratiche di comunicazione umana alla corretta amministrazione del contratto.

Una buona pratica di Project Management contribuisce a creare un ambiente collaborativo tra le persone coinvolte nel progetto con ruoli e responsabilità differenti (De Marco).

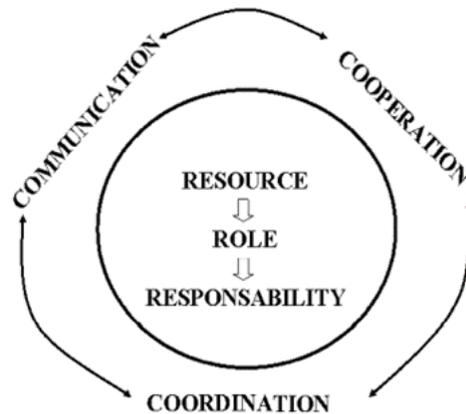


Figure 3: PM skills

Il project management si occupa di organizzare ed integrare le fasi del progetto, determinate da necessità di governance e di controllo.

Queste fasi rispondono ad una sequenza logica, con un inizio e una fine, e usano risorse per produrre output rilasciabili. Per ogni progetto possiamo riconoscere cinque macro-fasi:

- **Avvio**
Il progetto nasce per necessità interne o esterne all'organizzazione; sono definiti i requisiti, l'offerta e il contratto
- **Pianificazione**
Si prevede lo sviluppo futuro del progetto per ottimizzare i tempi della realizzazione, l'impiego delle risorse e i costi da sostenere.
- **Esecuzione**
Avviene l'effettivo sviluppo del progetto, costruendo il prodotto o servizio
- **Monitoraggio e Controllo**
Si monitora l'andamento del progetto e, eventualmente, si interviene al fine di rimanere negli obiettivi iniziali
- **Chiusura**
Si consegnano i deliverables al cliente, si chiude il contratto e si impostano le attività di AMS.

Le fasi di progetto sono globalmente conosciute come ciclo di vita del progetto (ISO).



Figure 4: Ciclo di vita del progetto

All'inizio del progetto, a seconda delle sue caratteristiche e delle competenze dell'organizzazione, viene scelta una metodologia di gestione del progetto, ossia un sistema di principi, tecniche e procedure usate dalle persone che lavorano in un dato ambito ("Metodologie di gestione dei progetti: i dodici framework più usati • Asana").

È dimostrato che utilizzando uno stesso standard metodologico di riferimento per tutti i progetti di un'organizzazione è possibile ridurre i rischi e le incertezze e di conseguenza anche i costi ed i tempi complessivi ("Metodologia Project Management | Project Management Center").

2. METODOLOGIE TRADIZIONALI

Nel seguente capitolo saranno analizzate nel dettaglio le metodologie tradizionali Waterfall e Agile, analizzandone caratteristiche, strumenti e fasi di sviluppo, illustrando i contesti applicativi e valutandone pro e contro.

Successivamente le due metodologie saranno confrontate per far emergere una terza metodologia, quella Ibrida, la quale sfrutta le caratteristiche di entrambe le metodologie precedenti.

2.1. METODOLOGIA WATERFALL

La metodologia Waterfall è stato il primo approccio alla gestione dei progetti, nata all'interno dell'industria manifatturiera e edile (Laoyan). L'obiettivo di tale approccio consiste nell'ottimizzare e rendere efficienti le attività necessarie per portare a termine il progetto iniziale, quindi il completamento del progetto entro i tempi, il budget e scope previsti (Špundak, 2014).

2.1.1. CARATTERISTICHE

La metodologia waterfall è stata costruita per portare al successo i progetti manifatturieri ed edilizi; ne consegue che le sue caratteristiche fondamentali, elencate di seguito, siano state plasmate per adattarsi a quegli ambiti:

- **Linearità**
Le fasi sono sviluppate una ed una sola volta; non sono previste reiterazioni.
- **Sequenzialità**
Ogni attività cade a "cascata" (da qui il nome "waterfall") nell'attività successiva, quindi l'output della fase precedente è l'input della fase successiva. Per questa ragione non è possibile passare alla fase successiva prima di aver completamente terminato quella precedente
- **Precisa documentazione**
Più la documentazione è precisa, minore sono le probabilità di incorrere in cambi di requisiti e rilavorazioni, le quali causerebbero un'eccessiva onerosità in termini di tempi e costi a causa della sequenzialità
- **Scarsa comunicazione con il cliente**
Data la difficoltà del modello a discostarsi dal piano iniziale, la comunicazione con il cliente, superata le prime fasi di definizione del progetto, è ridotta al minimo: il cliente vede direttamente il prodotto finito alla conclusione del progetto.

2.1.2. STRUMENTI

Per aiutare il Project Manager durante l'attività di pianificazione, la metodologia Waterfall divide una prima attività di pianificazione e una successiva di programmazione, prevedendo l'utilizzo di alcuni strumenti.

Si discuterà prima degli strumenti di pianificazione con le quali sono scomposte le attività di progetto (WBS), l'organigramma del team (OBS) e i costi (CBS) del progetto.

Successivamente, si descriveranno gli strumenti di programmazione, che permettono al Project Manager di definire il cronoprogramma delle attività precedentemente individuate; si

distingueranno gli strumenti che permettono di allineare team di progetto e cliente, quindi overall master schedule e project schedule, e le tecniche di scheduling, come Matrix Scheduling, Gantt Diagram e Network Diagram.

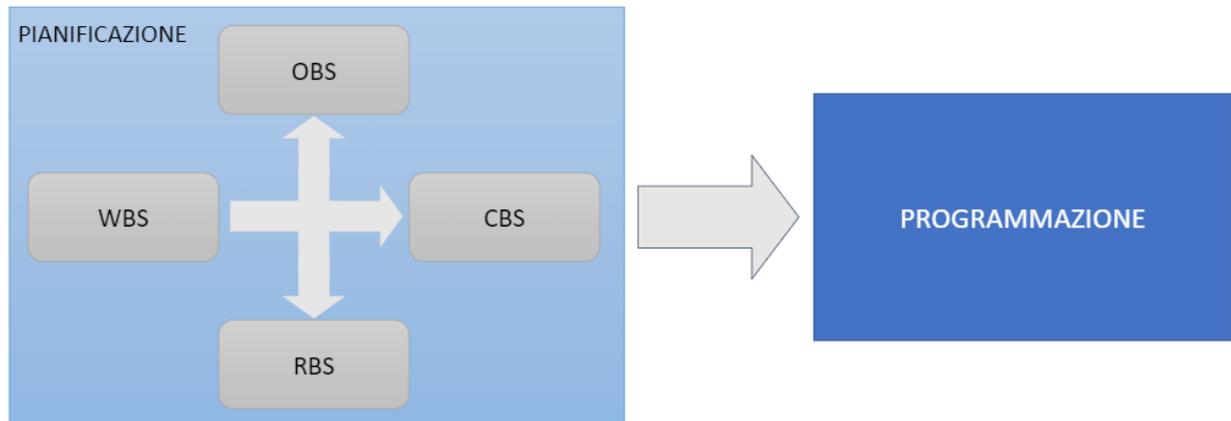


Figure 5: Processo di Pianificazione

2.1.2.1. PIANIFICAZIONE

Durante questa prima fase di pianificazione, la metodologia propone l'utilizzo di alcuni strumenti che hanno lo scopo di determinare "cosa" verrà fatto, "chi" svolgerà le attività e "quanto" le attività costeranno (De Marco).

WORK BREAKDOWN STRUCTURE

La WBS è lo strumento utilizzato per capire "Cosa" verrà fatto. È una rappresentazione analitica del progetto che scompone le attività in task sempre più piccoli, organizzati in gerarchie in base ai deliverable. L'obiettivo della WBS è quello di spaccettare le attività il più possibile arrivando al work package, unità di lavoro indivisibile, che permette di identificare un monte ore per ogni risorsa e un centro funzionale affidatario, in modo da attribuire un costo e una responsabilità alla commessa (De Marco, 2018).

Inoltre, la WBS definisce i possibili vincoli che possono esistere tra i work packages, ad esempio: il team non può passare alla lavorazione di un determinato work package prima di averne completato prima un altro.

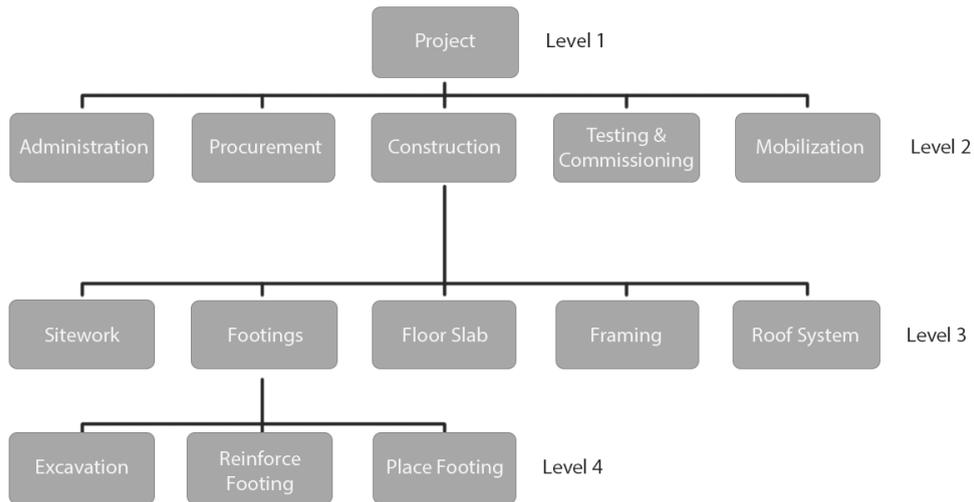


Figure 6: WBS

L'utilizzo della WBS presenta i seguenti vantaggi:

- Il Project Manager è sicuro di non trascurare nulla
- È una base per calcolare costi del progetto
- Rende facile la comunicazione interna all'organizzazione

ORGANIZATIONAL BREAKDOWN STRUCTURE

Attraverso la OBS il Project Manager decide "Chi" svolgerà le attività di progetto. Lo strumento permette di identificare la struttura organizzativa dell'azienda e i membri del team di progetto con le loro relazioni gerarchiche (De Marco, 2018).

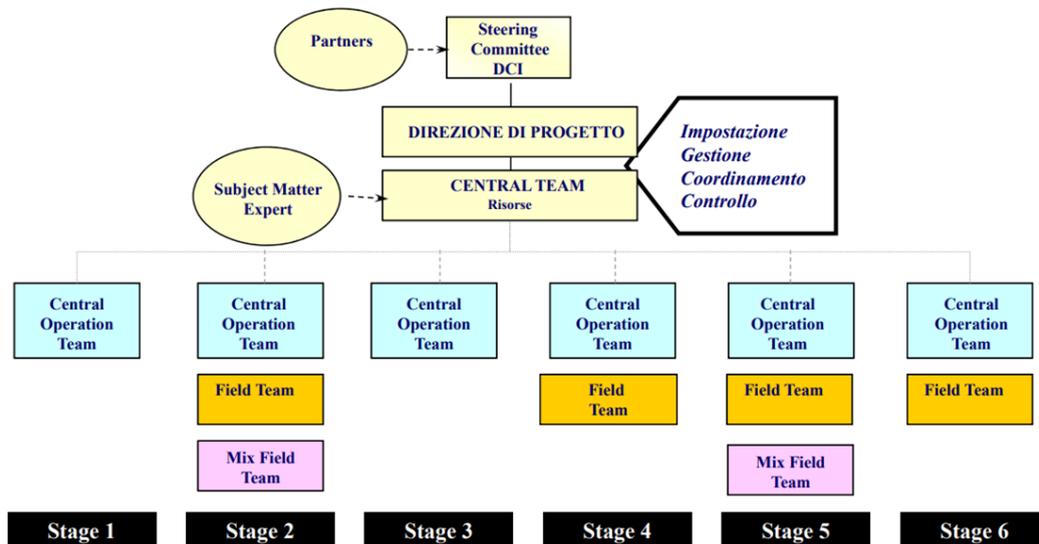


Figure 7: OBS

L'OBS permette di:

- Individuare i ruoli
- Assegnare responsabilità
- Definire i flussi informativi

Incrociando l'OBS con la WBS si costruisce la matrice compiti/responsabilità; infatti, l'intersezione tra un work package e un centro di responsabilità determina l'individuazione di un compito, che può assumere uno dei quattro ruoli RACI, descritti di seguito

- **Responsible:** gli è assegnata l'esecuzione dell'attività
- **Accountable:** gli è assegnata la responsabilità sul risultato dell'attività
- **Consultable:** aiuta e collabora con il Responsible
- **Informed:** deve essere informato all'inizio e alla fine dell'attività

Step	Project Initiation	Project Executive	Project Manager	Business Analyst	Technical Architect	Application Developers
1	Task 1	C	A/R	C	I	I
2	Task 2	A	I	R	C	I
3	Task 3	A	I	R	C	I
4	Task 4	C	A	I	R	I

Figure 8: RACI

RESOURCE BREAKDOWN STRUCTURE

L'RBS identifica tutte le risorse utilizzate nel progetto per svolgere le attività definite dalla WBS, ripartendole per categorie.

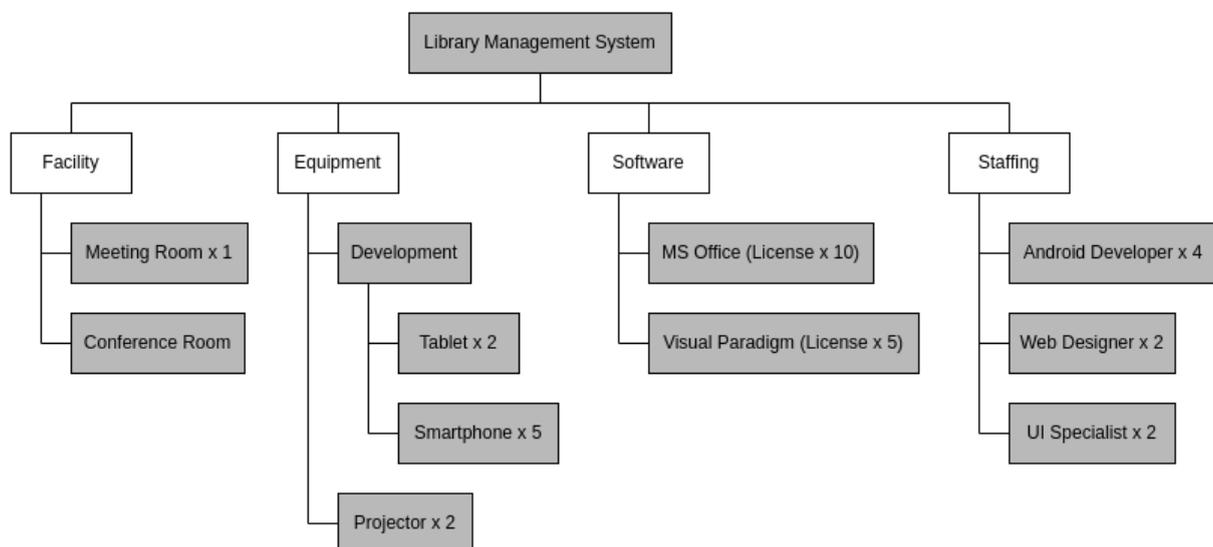


Figure 9: RBS

Esistono varie metodologie per stimare le risorse necessarie alle attività di progetto:

- **Per analogia**
Il progetto viene confrontato con altri simili.
- **Giudizio degli esperti**
Viene richiesto il giudizio di un esperto che può valutare quali siano le risorse

necessarie per portare a completamento un progetto sulla base della sua esperienza.

- **Calcolo analitico**

Il dimensionamento delle risorse è effettuato sulla base della produttività delle risorse per le diverse tipologie di lavoro.

- **Metodo parametrico**

Viene effettuata una stima delle risorse necessarie sulla base di modelli parametrici standard.

COST BREAKDOWN STRUCTURE

Con la CBS il Project Manager può capire quali saranno i costi diretti imputabili al progetto, valorizzando economicamente ogni attività, noti i costi unitari e l'entità delle risorse assegnate (De Marco, 2018).

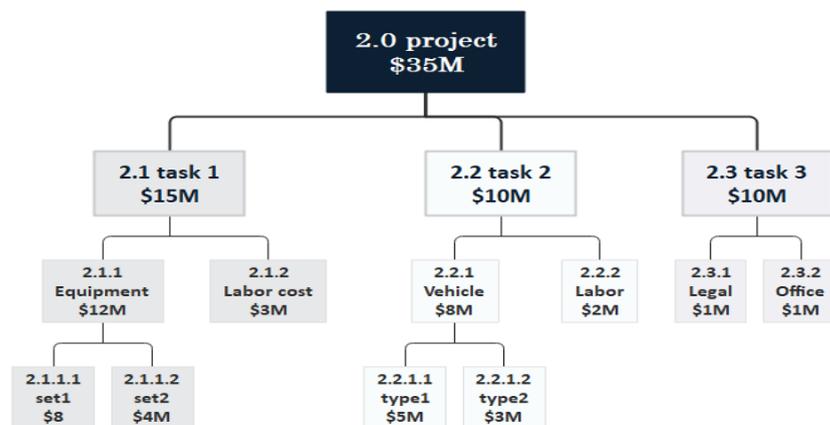


Figure 10: CBS

È possibile effettuare la stima dei costi attraverso due approcci diversi:

1. **Top Down**

Si valuta il costo complessivo del progetto e lo si scompone attraverso iterazioni nelle varie componenti.

2. **Bottom Up**

Si valutano i costi delle singole attività per ogni Work Packages e si calcolano i costi totali per aggregazione.

2.1.2.2. PROGRAMMAZIONE

Dopo aver individuato le componenti del progetto durante la prima fase di pianificazione, nella fase di programmazione il Project Manager decide “quando” le attività saranno svolte. Contestualmente, il PM dovrà redigere i seguenti documenti al fine di allineare l'organizzazione e il cliente:

- **Overall Master Schedule**

Il documento viene redatto in fase di offerta e presentato ai diversi enti aziendali durante il kick-off meeting. In sostanza, è una schedulazione del progetto a livello di sintesi che identifica i principali deliverable e i componenti della struttura di ripartizione del lavoro e le tappe fondamentali della schedulazione, dette milestone (A Guide to the Project Management Body of Knowledge., 2013). È utilizzato come

È un cronoprogramma delle attività visualizzato attraverso diagramma a barre. Si rappresentano le attività nell'ultimo livello della WBS attraverso segmenti di lunghezza proporzionale alla durata. Guardando il diagramma si riesce ad intuire quali attività sono correlate, ma non vi è un'esplicita dichiarazione di dipendenze. Anche se è uno strumento molto efficace per comunicare al committente o al manager come procederà il progetto, vi è la necessità di mostrare le relazioni tra le attività in modo più esplicito (De Marco, 2018).

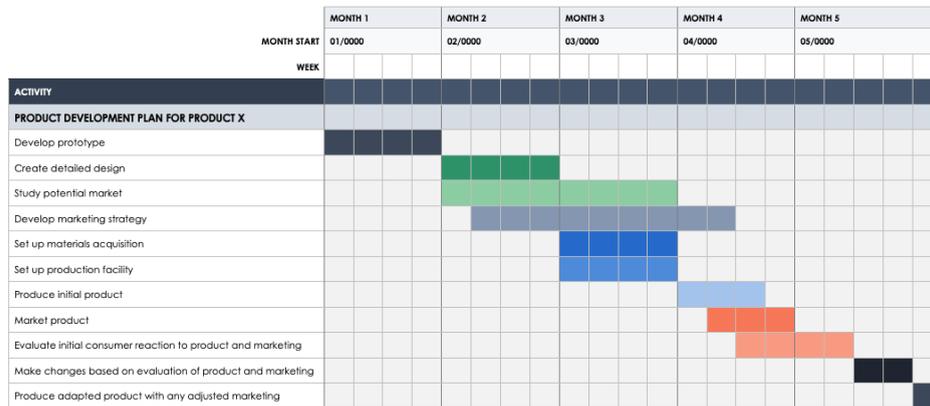


Figure 13: Gantt Chart

- **Network Diagram**

È una rappresentazione grafica delle relazioni logiche tra le attività della schedulazione del progetto. Può includere tutti i dettagli del progetto o una o più attività di sintesi e può essere accompagnato da una riassunto narrativo che descrive l'approccio di base utilizzato per la sequenza delle attività e l'eventuale presenza all'interno della rete di attività insolite (*A Guide to the Project Management Body of Knowledge.*, 2013).

Il pregio di questo metodo è quello di definire in modo chiaro le relazioni esistenti tra le attività, permettendo l'utilizzo di algoritmi (come il CPM) per calcolare la durata del progetto, identificando il percorso critico, ossia "la sequenza di attività che rappresenta il percorso più lungo di un progetto, che determina la durata più breve possibile del progetto" (*A Guide to the Project Management Body of Knowledge.*, 2013).

2.1.3. FASI

Come detto in precedenza, ogni progetto è unico, quindi è ovvio che siano presenti differenze tra progetti che utilizzano la metodologia waterfall, ma i concetti di base saranno presenti ed applicabili in tutti i progetti; di seguito sono descritte le fasi della metodologia.

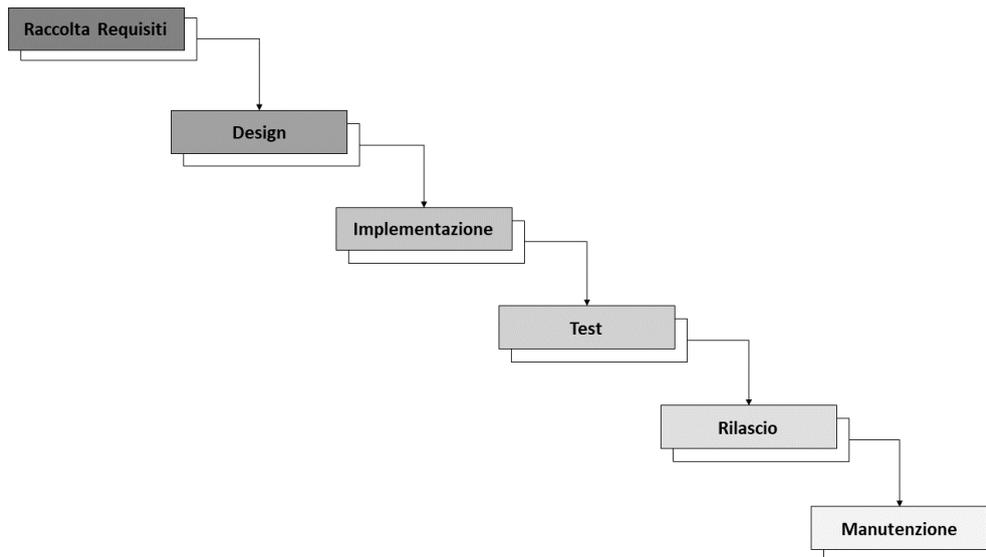


Figure 14: Fasi metodologia Waterfall

2.1.3.1. RACCOLTA DEI REQUISITI

La raccolta dei requisiti è il primo processo di pianificazione. Questa fase del progetto solitamente risulta essere la più lunga e delicata tra tutte, a causa della sequenzialità delle fasi nella metodologia waterfall, che non permettono la modifica dei requisiti iniziali a progetto avviato. Quindi, durante questa fase bisogna raccogliere la maggior quantità di informazioni possibili circa le specifiche del prodotto necessarie per soddisfare le richieste del cliente.

Il deliverable di questa fase è il documento di specifica dei requisiti, che sarà il documento cardine di tutti gli sviluppi delle successive fasi, il quale raccoglie:

- chi lavora in ogni fase
- le dipendenze fondamentali
- le risorse necessarie
- Analisi dei costi
- una cronologia che illustri quanto tempo ci vorrà per ogni fase
- Analisi dei benefici attesi

Quindi, per tutta la durata della fase di raccolta dei requisiti cooperano cliente, organizzazione aziendale e tecnici del team, in quanto nel documento saranno contenute le caratteristiche tecniche del sistema, motivo per il quale sarà necessario anche definire quali saranno i vincoli da apporre alle richieste del cliente. L'obiettivo della fase è quello di capire quali siano le funzionalità richieste per il prodotto finale, ma senza soffermarsi sul come queste saranno sviluppate.

2.1.3.2. DESIGN

Con la fase di design si conclude il processo di pianificazione.

L'obiettivo di questa fase è quello di elaborare la soluzione più adatta al fine di soddisfare le esigenze del cliente; partendo dal documento di specifica dei requisiti, redatto nella fase precedente, viene definita l'architettura del progetto.

A questo punto del progetto, i tecnici del team specificano quali saranno le tecnologie e le tecniche che verranno utilizzate per sviluppare la soluzione finale, definendo sia l'architettura di alto livello che la componentistica più specifica, in modo da dividere il progetto in vari moduli.

È molto importante che in questa fase venga documentata ogni decisione effettuata; in tal modo, durante tutta la durata del progetto, si può garantire consistenza con le decisioni prese nella fase di Design.

2.1.3.3. IMPLEMENTAZIONE

Nella fase di implementazione, i tecnici del team sviluppano il prodotto ideato in precedenza attraverso i documenti redatti (Laoyan).

Il team di progetto è diviso in sottogruppi, a cui sono affidati moduli definiti nella fase precedente. Il sottogruppo passa al modulo successivo solo dopo aver completato e testato con successo quello precedente. Il test viene effettuato sul modulo "isolato"; in questa fase non si tiene conto dell'integrazione del modulo con il resto del sistema.

L'obiettivo di questa fase è quello di implementare tutti i moduli definiti nella fase precedente.

2.1.3.4. TEST

Durante la fase di test, i singoli moduli sviluppati durante la fase precedente sono integrati all'interno del sistema. L'obiettivo è quello di garantire la consistenza tra i singoli moduli, quindi che il prodotto finale rispetti i requisiti raccolti.

I tester del progetto si occupano di testare che la soluzione fornita soddisfi i requisiti del cliente, redigendo un documento che identifica problemi sorti durante il controllo qualità (Laoyan), i quali saranno poi risolti da parte dei tecnici di sistema. È buona prassi pianificare un controllo di accettazione da parte del cliente (UAT), il quale può notificare se qualcosa non è stato implementato come deciso durante la fase di analisi dei requisiti. Al fine di semplificare la fase di UAT, insieme al cliente, può essere redatto un documento che raccoglie i criteri di accettazione, un insieme di regole che il sistema deve avere per soddisfare i requisiti del cliente.

2.1.3.5. RILASCIO

In questa fase il prodotto viene definitivamente consegnato al cliente (Laoyan); contestualmente, sono organizzati dei workshop con l'utenza al fine di insegnare il corretto utilizzo del prodotto.

2.1.3.6. MANUTENZIONE

Soprattutto nel settore informatico, dopo che il prodotto è stato consegnato al cliente, può capitare che siano scoperti nuovi malfunzionamenti non scovati durante la fase di testing, che debbano essere cambiati alcuni requisiti del sistema, o che cambi l'ambiente entro il quale il sistema viene utilizzato (Laoyan).

Per questa ragione, l'obiettivo di questa fase è quello di costruire un servizio di supporto al cliente, capace di rilasciare fix e aggiornamenti, così che il prodotto possa rimanere commerciabile.

2.1.4. APPLICAZIONI

La metodologia waterfall risulta essere efficace in contesti stabili e già esplorati dall'organizzazione, nei quali i requisiti del prodotto finale sono perfettamente descrivibili (Kisielnicki & Misiak, 2017) e il cliente non risulta avere necessità di effettuare dei cambiamenti dopo aver concluso le fasi iniziali, per cui rimangono gli stessi per tutta la durata del progetto. Difatti, questo approccio risulta essere molto efficace nei progetti che necessitano di replicare prodotti e servizi già noti, per i quali budget e tempi sono già definiti. Inoltre, grazie all'attenta e continua elaborazione di documenti durante tutte le fasi del processo, per cui ogni passaggio è spiegato nei minimi dettagli, questa metodologia è particolarmente consigliata alle organizzazioni i cui team sono frequentemente soggetti a turnover, poichè un nuovo membro del team potrà facilmente documentarsi, e quando i tecnici hanno un bagaglio di skills di livello medio-basso, dato che ogni fase ha risultati specifici e un processo di revisione.

2.1.5. VANTAGGI E SVANTAGGI

Come detto in precedenza, all'inizio di ogni progetto, il project manager dovrà scegliere quale approccio alla gestione usare per tutta la durata del progetto.

A tal proposito, sono di seguito descritti i vantaggi e gli svantaggi dell'adozione della metodologia waterfall.

VANTAGGI

- **Informazione condivisa**
Gli obiettivi del progetto sono discussi sin dal primo momento, pertanto tutti gli stakeholders, interni ed esterni, acquisiscono consapevolezza dei risultati desiderati. Ciò rende la progettazione e il delivery più semplici, minimizzando spreco di tempo e sforzo.
- **Essenzialità**
Viene sviluppato solo ciò che è effettivamente necessario e gli sviluppatori possono correggere eventuali errori già nelle prime fasi, permettendo alla fase di attuazione di svolgersi senza intoppi.
- **Monitoraggio**
Poiché il progetto è chiaramente tracciato dall'inizio alla fine, si è in grado di misurare rapidamente lo stato del lavoro raggiunto dal progetto e confrontare la metrica con quanto programmato, calcolando gli scostamenti e le stime a finire.
- **Controllo**
Siccome la metodologia conserva una visione integrata di tutto ciò che deve essere fatto e delle interrelazioni, consente un maggior livello di controllo sulle fasi di realizzazione e sui rischi associati

SVANTAGGI

- **Sequenzialità**
A causa della sequenzialità, il ritardo su una fase del progetto provoca il ritardo dell'intero progetto, in quanto tutti i processi slittano fino a che il processo in questione non è terminato.

- **Obsolescenza**
Chiusa la fase di raccolta dei requisiti, non più possibile modificare le specifiche; questo potrebbe portare ad un invecchiamento prematuro della soluzione ideata se all'interno del mercato comparisse una nuova tecnologia
- **Rischi legati alla fase di raccolta dei requisiti**
Nella pratica, per il cliente e i membri del team è estremamente difficile definire in un unico momento, durante le fasi iniziali del progetto, tutti i requisiti che il prodotto finale dovrebbe avere.
- **Modello conservativo**
La rigidità del modello non permette al team di generare ed esprimere nuove idee, il che potrebbe rendere il progetto poco attrattivo e i membri del team non efficienti (Špundak, 2014).
- **Test non iterativi**
Effettuare i test sull'integrazione dei moduli solo dopo la fase di implementazione, crea la condizione di rischio per la quale è possibile verificare le assunzioni stabilite dal team durante la fase di design solo quando il prodotto è stato ultimato.
- **Mancanza di feedback dal cliente**
Conclusa la raccolta dei requisiti, i feedback da parte del cliente sono previsti soltanto durante la fase di test; ma, molto spesso, il cliente si rende conto di come dovrebbe essere gestita una funzionalità solo quando questa è accessibile. La fase con cui il cliente entra per la prima volta a contatto con le funzionalità del prodotto è così avanzata da rendergli impossibile fornire feedback a causa dell'altissimo costo che potrebbe avere effettuare delle modifiche
- **Una taglia non va bene per tutti**
L'approccio waterfall alla gestione dei progetti enfatizza la robustezza come uno dei suoi vantaggi, creando metodi e tecniche da applicare a tutti i progetti in modo uniforme. Ma oggi i progetti sono sempre più complessi, con un numero maggiore di compiti e interrelazioni, mentre l'approccio si basa su relazioni gerarchiche e lineari tra i compiti e non può riflettere adeguatamente tutta la complessità e le dinamiche dei progetti odierni (Špundak, 2014).

2.2. METODOLOGIA AGILE

Al giorno d'oggi i progetti nascono in un ambiente perturbato dai progressi della tecnologia e da richieste di innovazioni continue da parte della clientela; quindi, il team di progetto si ritrova ad affrontare problematiche mai affrontate in altri progetti, senza potersi affidare a procedure consolidate per le quali vi è conoscenza sui livelli di incertezza e rischio di esecuzione. Progetti di questo tipo presentano alti tassi di cambiamento, complessità e rischio.

Queste caratteristiche possono creare problemi per l'approccio waterfall, che mira a determinare in anticipo la maggior parte dei requisiti e a limitare i cambiamenti; il cambiamento dei requisiti iniziali non può essere predetto attraverso una attenta raccolta dei requisiti, in quanto nasce da un cambiamento dell'ambiente esterno al progetto (Project Management Institute, Inc., 2017).

Nasce l'esigenza di nuove metodologie la cui caratteristica principale sia l'adattabilità: il cambiamento è inevitabile, quindi i nuovi approcci hanno bisogno di accettare i cambiamenti e riconoscere che è quasi impossibile creare un piano di progetto sin dall'inizio.

La metodologia di Project Management Agile nasce proprio per far fronte a questa esigenza; l'agilità può essere definita come la capacità di rispondere al cambiamento per creare valore in un ambiente aziendale turbolento. Un ambiente agile è caratterizzato dall'incertezza e necessita di una profonda conoscenza del settore di appartenenza del progetto, così che l'organizzazione possa consegnare il progetto nel più breve tempo possibile, anche in presenza di importanti cambiamenti durante il ciclo di vita del progetto (Špundak, 2014).

Al fine di accogliere tali cambiamenti, la metodologia Agile itera molteplici volte tutte le fasi del ciclo di vita del progetto in brevi periodi di tempo, solitamente 4 settimane, e lo scope finale viene costruito ad ogni iterazione (Project Management Institute, Inc., 2017).

In tal modo, è possibile lavorare su pezzi di requisiti, invece che sull'intera soluzione, privilegiando lo sviluppo delle funzionalità, piuttosto che la loro pianificazione, rendendo possibile eseguire simultaneamente le attività di sviluppo e test e consegnando al cliente parti della soluzione finale, mano a mano che la si costruisce.

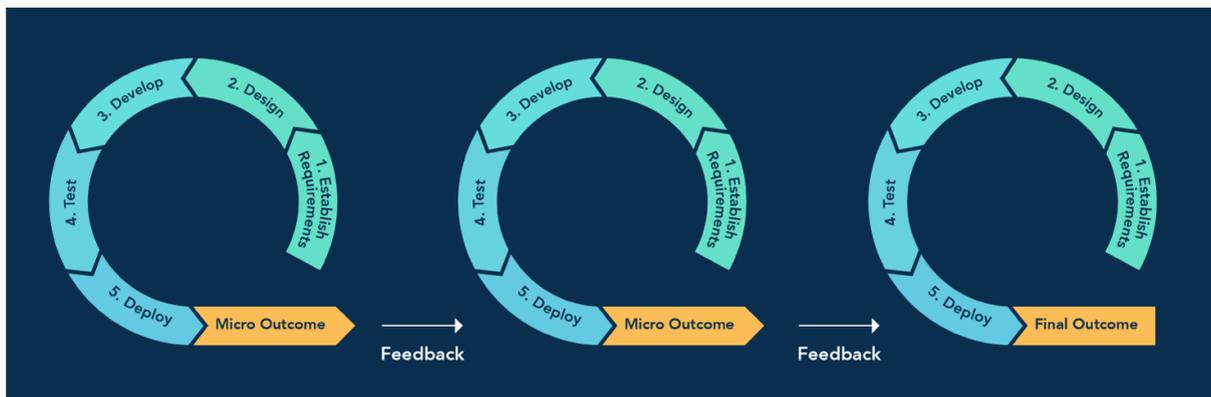


Figure 15: Ciclo di Sviluppo Agile

2.2.1. IL MANIFESTO

La nascita della metodologia Agile può essere ricondotta al Febbraio 2001, quando un gruppo di 17 persone del gruppo "Agile Alliance", consapevoli della necessità di una nuova metodologia di Project Management soprattutto per il settore Software, il quale è più incline ad incertezze e cambiamenti durante l'intera vita del progetto, si ritrovò per scrivere il Manifesto Agile. Il documento ha l'obiettivo di delineare valori e principi per la conformazione di una nuova metodologia più snella e flessibile rispetto quella precedente (*History: The Agile Manifesto*).

4 VALORI

L'obiettivo del manifesto non è quello di porre la metodologia Agile in netta contrapposizione con quella waterfall, ma di cambiare la prospettiva del Project Management, portando maggiore attenzione su alcuni aspetti piuttosto che altri.

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

Di fronte all'incertezza e all'innovazione i processi preconfezionati spesso risultano fallaci; poiché lo sviluppo del software deve essere guidato dalla creatività della squadra e dalle esigenze del cliente, il team di progetto deve essere aperto all'apprendimento e libero di sperimentare processi nuovi e flessibili (*Business Agility*).

Le probabilità di raggiungere l'obiettivo di progetto sono fortemente influenzate dalla coesione e affiatamento del gruppo; l'elemento fondamentale di questo primo valore è la comunicazione, sia per l'aspetto sociale che per il trasferimento efficiente del know-how.

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

La metodologia waterfall, basandosi sull'idea che una documentazione molto dettagliata permettesse di ideare una soluzione perfettamente in linea con le necessità del cliente, concentrava una lunga fase di raccolta dei requisiti prima dell'effettivo sviluppo del prodotto. In contrapposizione, per la metodologia Agile il prodotto funzionante è ogni giorno un indicatore migliore sull'allineamento alle esigenze del cliente. Si richiedono quindi unicamente le informazioni necessarie per iniziare lo sviluppo della successiva iterazione (*Business Agility*).

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

La negoziazione di un contratto porta alla definizione di requisiti che molto difficilmente possono cambiare. Proprio perchè per la metodologia Agile è spesso impossibile raccogliere in anticipo i requisiti e sono necessari feedback costanti da parte del cliente, si preferisce un approccio collaborativo che permette al team, a progetto in corso, di correggere alcuni aspetti della soluzione e al cliente di perfezionare la propria visione e ridefinire le proprie esigenze (*Business Agility*).

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

Durante il ciclo di sviluppo di un progetto, spesso i cambiamenti risultano inevitabili; una progettazione effettuata anni prima del rilascio può incontrare un mercato molto diverso rispetto a quello pronosticato, inoltre il cliente potrebbe richiedere cambiamenti dopo aver perfezionato la propria visione e le proprie esigenze. Agile risponde rilasciando piccole componenti del prodotto finale, piuttosto che seguire rigidamente un piano a lungo termine (*Business Agility*).

12 PRINCIPI

Le idee della metodologia Agile espresse nei 4 Valori sono tradotte nei 12 Principi, utili per aiutare e guidare la transizione dei team in Agile e verificare se le pratiche che stanno seguendo sono in linea con la cultura agile (*I Principi Sottostanti Al Manifesto Agile*).

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

2.2.2. APPLICAZIONI

La metodologia Agile viene utilizzata in contesti poco stabili, dove il team di progetto necessita di esplorare nuove tecnologie per far fronte ai requisiti del cliente. Grazie alla flessibilità concessa, il team può quindi sperimentare approcci diversi, costruendo soluzioni innovative.

Molto spesso accade che sia il cliente a richiedere l'utilizzo della metodologia agile, perchè potrebbe dubitare di conoscere tutti i requisiti che dovrà possedere la soluzione finale, o perchè desidera avere l'opportunità di cambiare tali requisiti. Inoltre, la metodologia risulta efficace quando il cliente desidera un time to market ristretto, perchè permette di rilasciare il Minimum Value Product a breve distanza dall'inizio del progetto. Altre volte, invece, il cliente ne vieta l'utilizzo, in quanto il contesto lavorativo richiede particolari vincoli che sono contrari ai principi Agile; ad esempio, il settore sanitario impone la produzione di una documentazione molto dettagliata (*When to Use Waterfall Vs. Agile, 2019*).

L'applicazione della metodologia Agile impone alcuni requisiti nei confronti dell'organizzazione fornitrice e cliente. Per gestire l'instabilità e i continui cambiamenti del progetto, il team deve essere capace di autogestirsi e possedere figure cross-funzionali capaci di completare ogni deliverable, motivo per il quale l'azienda deve essere provvista di personale di grande esperienza e possedere una struttura che permetta una comunicazione costruttiva e funzionale. Inoltre, a causa della scarsa documentazione prodotta e per salvaguardare la coesione del team, è importante che i componenti del team non varino durante il ciclo di vita del progetto.

D'altro canto, il cliente deve essere disponibile a partecipare al progetto, comunicando con il team per la definizione di nuovi requisiti e testando continuamente i deliverables.

Infine, siccome all'inizio del progetto non è chiaro in cosa consisterà precisamente la soluzione finale, il cliente deve poter disporre di ampio budget in termini di costo e tempo.

2.2.3. FRAMEWORK

La metodologia Agile non impone semplicemente uno schema da seguire per condurre le attività di Project Management, bensì richiede uno stile di gestione e una struttura del team diversa. I valori e principi esposti precedentemente definiscono una struttura flessibile

composta da valori chiave, principi e strategie suggerite, e non un insieme di regole e processi rigidi (Project Management Institute, Inc., 2017).

La metodologia richiede team generalmente piccoli, co-locati, responsabilizzati, inter-funzionali e auto-organizzati, suggerendo l'idea che se sono fornite risorse e supporto appropriati al team, i membri saranno pienamente coinvolti e motivati, evolvendo le loro relazioni e ruoli per risolvere i problemi e raggiungere gli obiettivi aziendali.

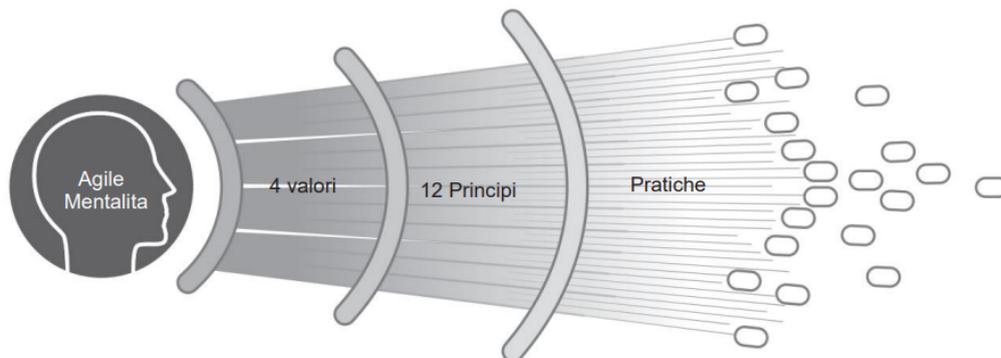


Figure 16: Framework Agile

Questo punto di vista spiega l'emergere di diverse Pratiche Agile; le più conosciute sono Scrum e Kanban, le quali saranno spiegate dettagliatamente nei successivi capitoli.

2.2.3.1. SCRUM

Scrum è indubbiamente il framework agile più diffuso; lo strumento aiuta i team a generare valore attraverso un approccio iterativo e adattivo per risolvere problemi complessi.

La filosofia del framework pone le sue basi sull'empirismo, per il quale la capacità di prendere decisioni può basarsi solo sull'esperienza e non su previsioni future, e sul lean thinking, che aiuta a creare valore mantenendo il focus solo su ciò che è davvero necessario (Sutherland & Schwaber).

Il metodo può essere modificato e applicato dalle aziende per adattarsi meglio al progetto, ma le seguenti tre caratteristiche sono immutabili:

- **Trasparenza**
Se la capacità di prendere decisioni si basa solo sull'esperienza, gli stakeholder e il team di progetto devono avere piena visibilità su obiettivi e andamento del progetto,
- **Ispezione**
Eventuali difformità rispetto al prodotto ideato possano essere subito scovate, quindi i test devono essere frequenti ma non tali da determinare ritardi nei lavori.
- **Adattamento**
Se la soluzione risulta inaccettabile, il team di progetto deve adeguare il prodotto o intervenire sul processo di produzione. La correzione deve essere portata a termine il più velocemente possibile per evitare nuovi scostamenti; il team deve autogestirsi, così da evitare di dover aspettare l'approvazione dall'alto.

(Sutherland & Schwaber)

Il framework prende il nome dal gioco del rugby, dove un gruppo di giocatori spinge unitamente nella stessa direzione per recuperare la palla. Analogamente, Scrum necessita che il team si autogestisca in modo coordinato, quindi prevede che i membri condividano visione e valori che dirigano in modo armonico e coerente gli sforzi.

È importante instillare all'interno del team i seguenti cinque valori:

- **Impegno**
Ogni singolo membro del team si impegna al fine di raggiungere gli obiettivi di progetto e supportare gli altri membri
- **Focus**
Il focus del team è rivolto esclusivamente al lavoro dello sprint, eliminando ogni fonte di distrazione
- **Apertura**
Il team agisce in modo aperto nei confronti del lavoro, affrontando problemi, successi, fallimenti ed errori.
- **Rispetto**
Ogni membro del team deve rispettare gli altri e pretendere che sia fatto altrettanto con la sua persona
- **Coraggio**
Il team deve avere il coraggio di fare la cosa giusta, per quanto difficile o impopolare (Sutherland & Schwaber)

Il framework si basa su iterazioni time-boxed, ossia dalla durata standardizzata di 3-4 settimane, chiamati sprint, i quali scompongono l'effort di progetto e hanno come obiettivo il rilascio di nuove funzionalità del prodotto finale.

Il processo di avanzamento del progetto messo in piedi dalla metodologia Scrum, fa uso di strumenti quali lo Scrum Team, gli Eventi e gli Artefatti; in seguito si illustreranno questi strumenti per delineare il processo di gestione del progetto Scrum.

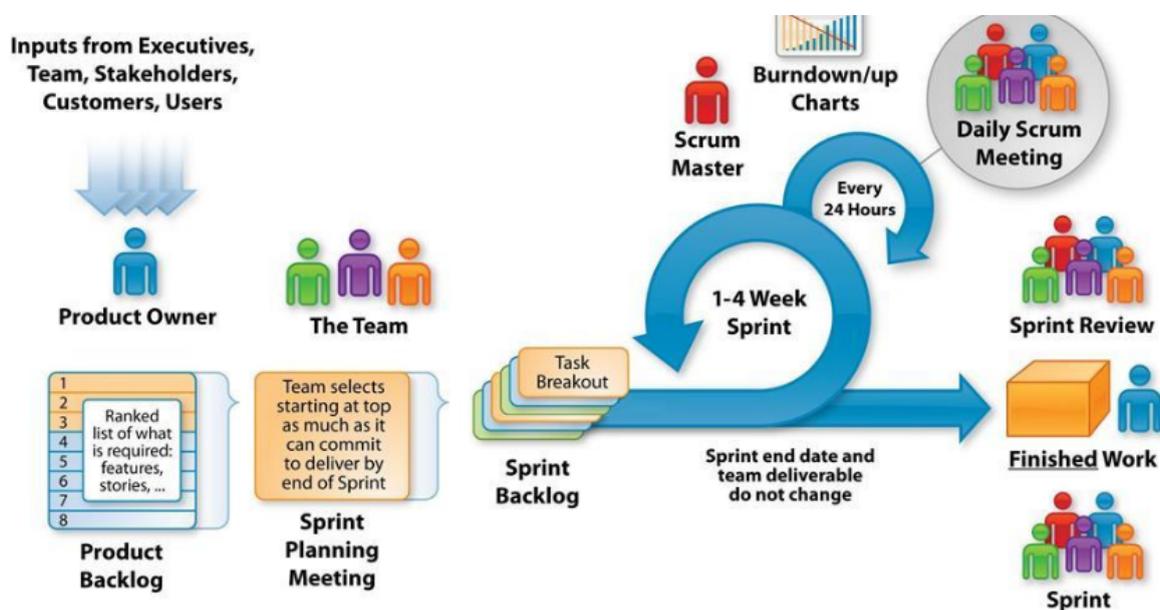


Figure 17: Scrum Framework

2.2.3.1.1. SCRUM TEAM

Lo Scrum Team consiste in un piccolo gruppo di persone composto da uno Scrum Master, un Product Owner e gli sviluppatori.

All'interno di un team Scrum non esistono sotto-team o gerarchie. Si tratta di un'unità coesa di professionisti concentrati su un unico progetto alla volta (Sutherland and Schwaber).

I team Scrum sono cross-funzionali, il team è composto da persone con diverso background tecnico, grazie al quale è possibile creare valore ad ogni Sprint, e autogestiti, quindi si

decide all'interno del team quando e come fare qualcosa, motivo per il quale è il team ad essere responsabile dell'incremento di valore dello sprint (Filippetti).
 In genere i team scrum non superano le 10 persone, in modo tale da rimanere agili e snelli.

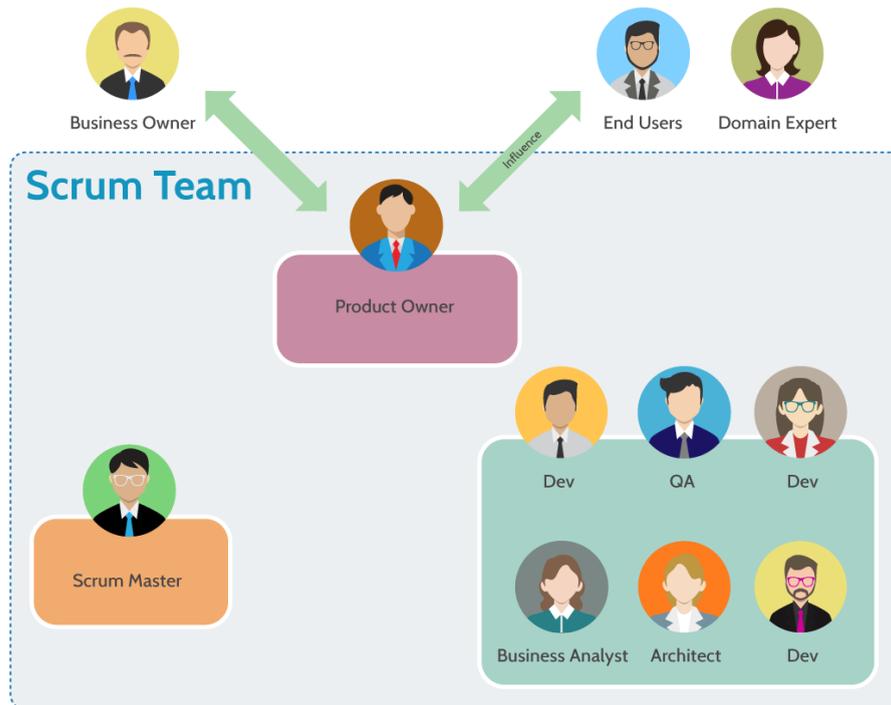


Figure 18: Scrum Team

Product Owner

Il Product Owner rappresenta gli stakeholders del progetto; il suo obiettivo è quello di massimizzare ad ogni Sprint il valore portato dal Team alla soluzione finale (Sutherland & Schwaber).

I compiti del Product Owner sono quindi:

- Creare i componenti della Product Backlog, definendo i requisiti del prodotto e assegnando loro una priorità.
- Comunicare efficacemente il contenuto della Product Backlog a tutti gli stakeholders di progetto.
- Aggiornare la Product Backlog con i risultati di ogni sprint, rendendoli chiari e trasparenti.
- Occuparsi della stesura della User Story, ossia informazioni legate all'utente, alle funzionalità e al motivo per cui dovranno essere sviluppate, che dovranno essere presentate al team di sviluppo.
- Definire il cronoprogramma delle release, definendone i criteri di accettazione

(Filippetti)

Scrum Master

Lo Scrum Master è il responsabile della gestione del progetto e della corretta applicazione di regole e processi Scrum. Il suo obiettivo è quello di moderare e agevolare le relazioni all'interno della squadra (Sutherland & Schwaber).

I compiti dello Scrum Master sono:

- Insegnare la teoria Scrum e metterne in pratica le tecniche nel team e dell'organizzazione.

- Garantire il funzionamento e la produttività di tutti gli eventi Scrum
- Rimuovere gli ostacoli che limitano la capacità del team di raggiungere l'obiettivo dello Sprint e i deliverable previsti.
- Occuparsi del miglioramento del team, migliorando le pratiche all'interno del framework Scrum, proteggendolo da influenze esterne, allenando i componenti all'autogestione.
- Assistere il Product Owner nel creare e presentare la Product Backlog al resto del team e agli stakeholders.

(Filippetti)

Team di Sviluppo

Il team di sviluppo è composto da un gruppo di persone che si occupa di sviluppare e testare la soluzione, apportando nuovo valore ad ogni sprint. Le competenze del team devono essere cross-funzionali, così da poter apportare valore ad ogni sprint (Sutherland & Schwaber).

I compiti dei membri del team di sviluppo sono i seguenti:

- Creare un piano per lo Sprint
- Creare lo Sprint Backlog
- Instillare la qualità aderendo a una definizione di "Fatto";
- Adattare ogni giorno il piano all'obiettivo dello Sprint
- Comportarsi in modo responsabile verso gli altri membri del team
- Comunicare allo Scrum Master lo stato di avanzamento delle attività ed eventuali anomalie riscontrate
- Comunicare al Product Owner lo stato di avanzamento dei lavori

(Filippetti)

2.2.3.1.2. ARTEFATTI

Per garantire la trasparenza, l'ispezione e l'adattamento durante l'intero ciclo di vita del progetto, Scrum crea e aggiorna degli strumenti, chiamati Artefatti. Inoltre, gli artefatti sono utili al fine di tracciare lo stato di avanzamento dei lavori ad ogni iterazione (Sutherland & Schwaber).

Product Backlog

Il Product Backlog è il documento contenente tutto il lavoro da effettuare durante il corso del progetto; ogni item del Product Backlog rappresenta una attività da svolgere per creare, mantenere o evolvere la soluzione finale (Sutherland & Schwaber).

Il Product Owner ha la responsabilità di creare, aggiornare e rendere visibile a tutti il Product Backlog per tutta la durata del progetto, inserendo i Product Backlog Items, ossia i singoli requisiti del prodotto; un requisito di prodotto è contenuto all'interno della Product Backlog se il Product Owner lo ritiene importante attraverso il metodo MoSCoW:

- **Must have:** il requisito è necessario
- **Should Have:** il requisito dovrebbe esserci.
- **Could Have:** il requisito potrebbe esserci.
- **Won't have:** il requisito dovrà essere scartato

Successivamente i membri del team assegnano ad ogni PBI una stima di effort necessaria per svilupparla, utilizzando l'unità di misura degli story points, così che il Product Owner possa decidere quali PBI saranno svolte prima.

	User story	Story point(s)	Priority
High priority	As a user, I am able to search for documents so I can find them more easily	2	1
	As a site visitor, I can compare different types of accounts to see which account type suites me best	1	2
	As a user, I can submit questions through the website so I know how to better use the product	1	3
	As a site visitor, I am shown what I can do in the product so I know whether or not this product will fill my needs	2	4
	As a user, I want to be able to retrieve documents that were deleted so I can reclaim documents that were deleted on accident	3	5
	As a site visitor and user, I can sign up for newsletters to remain up to date on the product	2	6
	As a user, I am notified when a new feature is released so I know what is possible	1	7
	As a user, I can change my user name if desired	3	8
	As an admin, I need the ability to update which team a user belongs to so I can make sure all teams are up to date	3	9
Low priority	As a user, I can enable spell check so I can be confident my final document has no spelling errors	4	10

Figure 19: Product Backlog

Infine, il Product Backlog contiene il Product Goal, che descrive il beneficio, collegato alla strategia aziendale, che si vuole conseguire attraverso ciò che il progetto realizza. Ciò è utile perchè orienta gli sforzi del team e degli stakeholders, oltre a rappresentare la base per definire gli obiettivi di ogni singolo sprint (Filippetti, 2022).

Incremento

L'incremento è una componente della soluzione finale creata durante uno Sprint. Ogni Incremento è "additivo", si aggiunge agli Incrementi realizzati in precedenza in un insieme verificato, coeso e perfettamente integrato e, poiché in Scrum l'Incremento ha lo scopo di fornire valore, ogni Incremento deve essere utilizzabile, funzionante e potenzialmente rilasciabile (Filippetti, 2022).

Epiche

Con il fine di riorganizzare tutti i task all'interno della Product Backlog, il Product Owner può raggruppare più task all'interno di un'epica, la quale può essere lavorata per più sprint prima di essere consegnata (Schwaber, 2022).

In genere, le epiche contengono tutti i task necessari per sviluppare una certa funzionalità.

Sprint Backlog

Lo Sprint Backlog è un piano costruito dal team di sviluppo, da seguire durante lo Sprint successivo. Esso è costituito da:

- Lo Sprint Goal, ossia l'obiettivo che il team si pone di raggiungere durante lo Sprint

- I Product Backlog Items selezionati per essere svolti durante lo Sprint, presi dalla cima della Product Backlog; il numero di items selezionati dipende dal numero di Story Point medi completati dal team durante gli Sprint precedenti.
- Il piano operativo con il quale si intende realizzare quegli Items, quindi trasformarli in un Incremento.

(Sutherland & Schwaber)

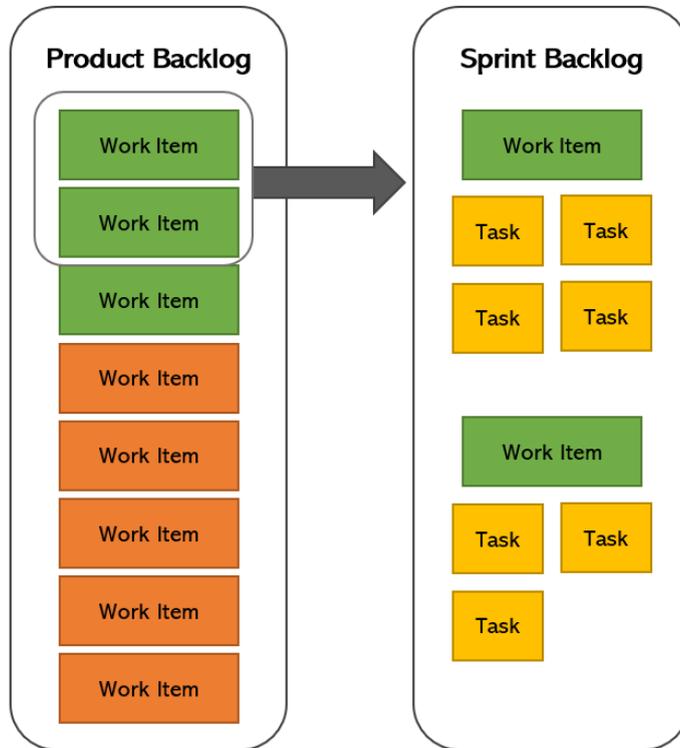


Figure 20: Sprint Backlog

Sprint Burndown Chart

Lo Sprint Burndown Chart è un grafico che mette in relazione, giorno per giorno, durante tutta la durata dello sprint, gli story points inizialmente stimati per finire lo sprint con successo con quelli effettivamente necessari per terminare i task dello sprint backlog.

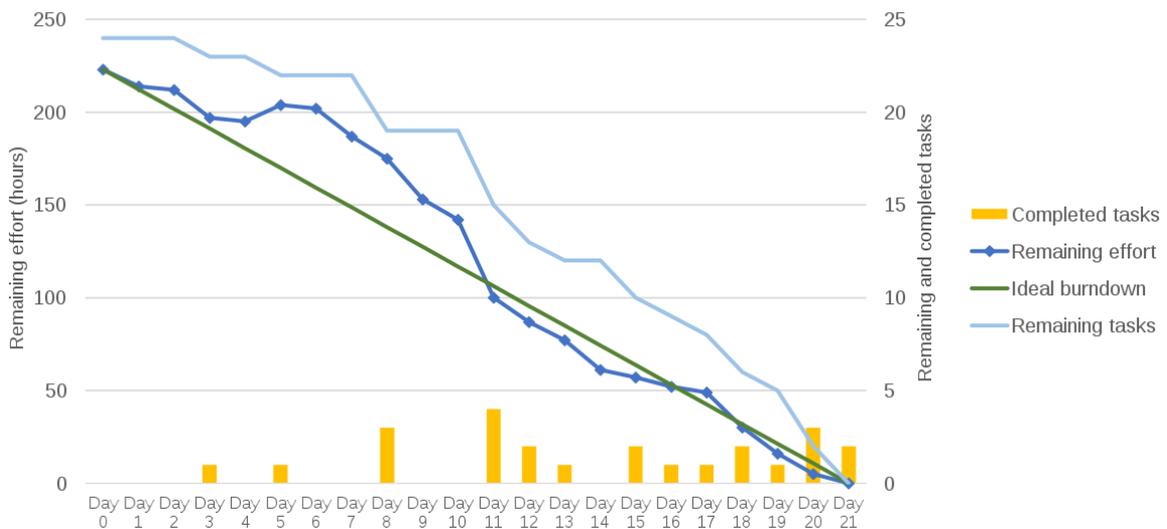


Figure 21: Sprint Burndown Chart

Velocity Chart

Lo scopo del Velocity chart è quello di mostrare graficamente il valore prodotto dal team di progetto alla fine di ogni sprint, calcolato come numero di story point consegnati.

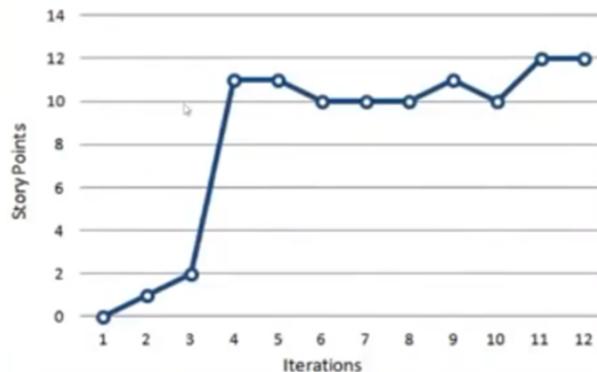


Figure 22: Velocity Chart

2.2.3.1.3. EVENTI

Gli Eventi consistono in opportunità per ispezionare e adattare gli artefatti Scrum, così da garantire la necessaria trasparenza; gli Eventi, se eseguiti come da framework, creano regolarità e aumentano la produttività, oltre a rendere superflue ulteriori riunioni. Scrum prevede che gli eventi siano time-box, dalla durata predefinita, così che possa essere dato il giusto peso alla fase di pianificazione, riducendo gli sprechi di tempo (Filippetti, 2022). Tra gli Eventi rientra anche lo Sprint, il quale è un Evento particolare in quanto contenitore di tutti gli altri Eventi.

Sprint

Lo Sprint è il vero cuore del framework Scrum: nulla in Scrum accade al di fuori di uno Sprint. Tutto il lavoro necessario per raggiungere l'obiettivo di prodotto, tra cui Sprint Planning, Daily Scrums, Sprint Review, e Sprint Retrospective, viene svolto al suo interno. Lo Sprint è un evento time-boxed dalla durata massima di un mese (Sutherland & Schwaber).

Sprint Planning Meeting

Lo Sprint Planning Meeting è l'Evento di apertura di ogni nuovo Sprint. All'Evento partecipa l'intero Scrum team e se interessati anche i manager aziendali e clienti, con l'obiettivo di decidere le attività da intraprendere durante lo Sprint successivo e quale sarà il risultato del lavoro del team di progetto, sviluppando lo Sprint Backlog. L'avvio dell'Evento richiede i risultati dello Sprint precedente e la Product Backlog aggiornata, al fine di capire quali siano le attività con la priorità più alta da assegnare al team nello Sprint successivo (Filippetti, 2022).

In genere, la durata dello Sprint Planning Meeting non supera le 8 ore.

Daily Scrum

Il daily scrum è un meeting della durata di circa 15 minuti che si svolge ogni giorno, in cui gli sviluppatori del team valutano l'avanzamento rispetto allo Sprint Goal e, qualora fosse necessario, ripianificano lo Sprint Backlog (Plaza). Durante questo evento tutti i membri del

team di sviluppo comunicano cosa hanno fatto durante il giorno precedente, cosa faranno durante la giornata odierna e se si aspettano che l'attuale task potrà avere impatti sul lavoro altrui.

Sprint Review

Lo Sprint Review è una riunione dalla durata massima di quattro ore, penultimo evento dello Sprint. Lo scopo dell'evento è quello di esaminare i risultati dello Sprint e determinare gli adattamenti futuri (Sutherland & Schwaber).

Durante lo Sprint Review, lo Scrum Team raccoglie i feedback degli stakeholders su cosa è cambiato all'interno e all'esterno dell'organizzazione, lavorando con loro per decidere come proseguire, adattando il Product Backlog (Filippetti, 2022).

Sprint Retrospective

Lo Sprint Retrospective è l'evento di chiusura dello Sprint. L'evento consiste in una riunione, che si protrae per un massimo di 3 ore, durante la quale il team Scrum esamina come è andato l'ultimo Sprint in termini di interazioni tra individui, processi e strumenti, pianificando modi per migliorare la qualità e l'efficacia del team (Sutherland & Schwaber).

2.2.3.2. KANBAN

Kanban è un altro framework Agile sviluppato nel 1940 in Toyota e consiste in un sistema di pianificazione del lavoro per ottimizzare l'efficienza produttiva e raggiungere il Just In Time (JIT).

Il JIT è una filosofia di produzione che consiste nell'avere ciò che è necessario, nella giusta qualità, nel posto giusto e al momento giusto; per arrivare a questo obiettivo, bisogna produrre in modo Lean, ossia eliminare tutte le fonti di spreco, con processi costanti e fatti di piccoli miglioramenti osservabili (Wakode et al., 2015).

Kanban permette al team di progetto di lavorare in modo Lean, massimizzando la produttività e riducendo i tempi morti, i quali possono verificarsi in qualsiasi processo, flusso di lavoro o procedura e di solito sono causati da inefficienze del processo stesso (Wakode et al., 2015).

L'implementazione del framework necessita che siano seguiti i seguenti 4 principi e 6 pratiche Kanban.

2.2.3.2.1. PRINCIPI

Per indurre il team a lavorare in modo Lean è necessario apportare alcune modifiche al processo attualmente in uso, ma spesso il team di progetto oppone resistenza alla richiesta di cambiamento perchè può creare paura e abbassa l'autostima, comunicando che le competenze esistenti non sono più apprezzate (Anderson & Reinertsen, 2010). Kanban, attraverso i suoi 4 principi, è capace di apportare i cambiamenti in modo gentile; i processi preesistenti all'interno del team non sono sconvolti, ma piuttosto si avvia un processo di continui piccoli miglioramenti in relazione tra loro, basandosi su ciò che si sta già facendo. Grazie a questo approccio non distruttivo è più facilmente accettato dai team e dal management, poiché non minaccia il flusso di lavoro effettivo, l'organizzazione del team o le persone coinvolte (Sistema Kanban: Una Panoramica Della Metodologia Kanban, 2021). Inoltre, ciò lo rende estremamente flessibile: Kanban può essere applicato a qualsiasi tipo di progetto o flusso di lavoro in qualsiasi campo (Sistema Kanban: Una Panoramica Della

Metodologia Kanban, 2021).

Di seguito sono descritti i 4 Principi Kanban:

- **Iniziare con ciò che si fa ora**
Come già detto, Kanban migliora il processo esistente, riconoscendo il valore dove esiste e affrontando i problemi che ne ostacolano i risultati (*4 Core Kanban Principles and 6 Practices (2022)*).
- **Concorda sull'ottenimento di cambiamenti evolutivi e progressivi**
La realizzazione del cambiamento in piccoli incrementi continui si sposa con il concetto di miglioramento continuo e contribuisce a ridurre al minimo le interruzioni e a rendere più gestibile l'intero processo di cambiamento (*4 Core Kanban Principles and 6 Practices (2022)*).
- **Rispetta inizialmente i ruoli, le responsabilità e le gerarchie**
Rispettare i ruoli dell'organizzazione rafforza l'idea che Kanban non sarà eccessivamente dirompente e non cambierà completamente i modelli operativi preesistenti, ma li migliorerà delicatamente senza cambiare la struttura di base. L'obiettivo è quello di ottenere il supporto di Kanban in un ambiente in cui le persone sono interessate a mantenere le pratiche consolidate (*4 Core Kanban Principles and 6 Practices (2022)*).
- **Incoraggia iniziative di leadership a tutti i livelli**
Kanban riconosce il potere della collaborazione, ma permette anche a chiunque di appropriarsi di un problema e di affrontarlo. Grazie alla trasparenza offerta, chiunque armato di una solida giustificazione e di dati a sostegno della propria argomentazione è in grado di agire (*4 Core Kanban Principles and 6 Practices (2022)*).

2.2.3.2.2. PRATICHE

Per promuovere uno sviluppo Lean, il Metodo Kanban introduce un sistema adattivo complesso, che parte da pratiche semplici necessarie per promuovere comportamenti complessi, adattivi ed emergenti (Anderson & Reinertsen, 2010).

Le pratiche sono:

- **Visualizzare il flusso di lavoro**
Per implementare Kanban viene utilizzata una Kanban board, che consente di visualizzare il flusso di lavoro in modo tale che il team possa rendersi conto di quali siano i colli di bottiglia del sistema (*Metodo Kanban Nel Project Management | Project Management Center*).
La board è divisa in colonne dal numero variabile; ogni colonna corrisponde ad un diverso stato di lavorazione dei task.
All'interno della board i task sono rappresentati da schede Kanban, posizionati all'interno di una specifica colonna a seconda dello stato di lavorazione del task.
Ogni scheda Kanban, quindi, inizia il suo percorso nella prima colonna "To Do", dove il task non è ancora stato iniziato. Una volta che il team di progetto inizia la lavorazione del task, la scheda Kanban si muove lungo le successive colonne, fino ad arrivare all'ultima colonna "Done", quando il team ha terminato la lavorazione del task (Ilmi et al., 2020).

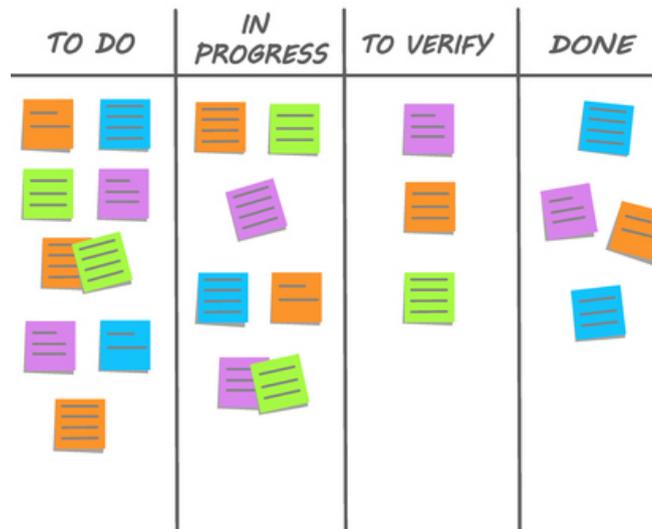


Figure 23: Kanban Board

- Limitare il lavoro in corso**
 All'interno della Kanban Board il numero delle schede Kanban è limitato, motivo per cui quando non ci sono più schede libere, non è possibile avviare alcun task. Solo quando un task verrà portato a termine, e quindi una scheda Kanban sarà libera, il team potrà iniziare un nuovo task (Anderson & Reinertsen, 2010). Questo meccanismo è noto come sistema pull, perché il nuovo lavoro viene inserito nel sistema solo quando c'è la capacità di gestirlo. Il sistema garantisce che il team non venga sovraccaricato di lavoro, così da garantire un ritmo di sviluppo sostenibile, il che assicura una maggiore qualità e prestazioni più elevate (Anderson & Reinertsen, 2010).
- Misurare e gestire il flusso**
 Il flusso di lavoro, ossia il passaggio di un task da uno stato ad un altro, deve essere costantemente monitorato e riportato con precisione attraverso le schede Kanban nella Kanban Board. La situazione ideale è che il flusso sia veloce e regolare, perché significa che il team di progetto sta generando valore nel più breve tempo possibile, riducendo al minimo i rischi e i costi causati dai ritardi (*5 Principles for an Effective Kanban System* -, 2018).
- Rendere esplicite le politiche di processo**
 Sino a quando i meccanismi e processi all'interno del team di progetto non sono definiti in modo esplicito, non sarà possibile al gruppo eseguire una vera autovalutazione che possa portare ad un miglioramento dei propri processi. Offrendo una comprensione esplicita, Kanban rende possibile una discussione oggettiva, razionale e basata su prove empiriche (*5 Principles for an Effective Kanban System* -, 2018).
- Utilizzare modelli per riconoscere le opportunità di miglioramento.**
 I cambiamenti da apportare all'interno del processo di lavoro devono sempre essere verificati attraverso l'uso di misurazioni oggettive. Individuare ed applicare le giuste misurazioni può essere difficile, ma poche metriche possono fornire tutte le informazioni utili per ottimizzare un processo (*Metodo Kanban Nel Project Management* | Project Management Center).

2.2.4. VANTAGGI E SVANTAGGI

Come fatto in precedenza per la metodologia Waterfall, anche per la metodologia Agile saranno di seguito proposti vantaggi e svantaggi nel suo utilizzo.

VANTAGGI

- **Flessibilità**
Durante il ciclo di vita di un progetto, le priorità e i requisiti possono facilmente cambiare.
Agile permette di rispondere al cambiamento mantenendo i costi contenuti (*When to Use Waterfall Vs. Agile, 2019*).
- **Time to market ridotto**
La maggiore attenzione sullo sviluppo del prodotto, più che sulla stesura della documentazione, e l'impronta iterativa della metodologia permettono il rilascio del Minimum Value Product a distanza non eccessiva rispetto l'inizio del progetto (*When to Use Waterfall Vs. Agile, 2019*).
- **Apprendimento incoraggiato**
Agile dona maggiori responsabilità al team, il quale si auto-organizza ed è spinto a riflettere sul proprio lavoro ad ogni iterazione, il che permette di lavorare in modo sempre più efficiente ad ogni sprint (*When to Use Waterfall Vs. Agile, 2019*).
- **Maggiore Fiducia sul progetto**
La possibilità di testare tutti i deliverable ad ogni iterazione e la partecipazione a tutti gli step del progetto creano maggiore fiducia sul progetto per tutti i membri del team. Non solo, grazie alla continua collaborazione tra team e cliente, anche quest'ultimo aumenta la propria fiducia nel proprio fornitore (*Agile: La Chiave per La Digital Transformation E La Crescita Aziendale, 2020*).

SVANTAGGI

- **Scope e tempistiche imprevedibili**
La flessibilità di Agile permette al cliente di cambiare i requisiti della soluzione finale anche a fasi avanzate del progetto, il che può portare alla necessità di sviluppare nuove funzionalità e rilavorare altre già rilasciate; per questa ragione, a priori è difficile definire quali saranno scope e tempistiche di rilascio della soluzione (*When to Use Waterfall Vs. Agile, 2019*).
- **Il cliente deve dedicare effort al progetto**
In Agile il cliente è visto come parte attiva del progetto, a cui sono affidate responsabilità come test della soluzione ad ogni iterazione, fornire feedback ed eventuali cambi di requisiti dovuti al cambiamento delle condizioni interne ed esterne al progetto. Per questa ragione, il cliente deve necessariamente fornire tempo e risorse da dedicare esclusivamente al progetto (*When to Use Waterfall Vs. Agile, 2019*).
- **Bassa possibilità di Turnover**
Affinché la metodologia Agile funzioni correttamente, necessita di alta fiducia tra i membri del team e una maggior attenzione allo sviluppo del prodotto, piuttosto che alla redazione della documentazione (*When to Use Waterfall Vs. Agile, 2019*).
Ciò si traduce in minori possibilità per l'organizzazione di cambiare i componenti del team durante il corso del progetto, in quanto la risorsa dovrà avere delle skills di un

livello abbastanza alto per lavorare con documentazione ridotta e il team dovrà dedicare effort nell'integrazione del nuovo componente.

2.3. METODOLOGIA HYBRID

Finora sono stati descritti i due approcci alla gestione del progetto tradizionali, elencando quali sono i rispettivi vantaggi e gli svantaggi della loro applicazione. Come si è già detto, a priori non è possibile determinare quale sia la metodologia migliore, ma la scelta dipenderà dalle caratteristiche dell'azienda cliente, fornitrice e del progetto.

In questo capitolo saranno messe a confronto le metodologie tradizionali, illustrando in quali contesti sia meglio utilizzare l'approccio Waterfall rispetto quello Agile e viceversa, per poi presentare una nuova metodologia, definita Ibrida, le cui caratteristiche sono una combinazione dei due approcci.

Infine, sarà presentata la metodologia dell'azienda Coolshop, oggetto di studio della tesi, spiegandone le caratteristiche e il funzionamento.

2.3.1. METODOLOGIE TRADIZIONALI - CONFRONTO

Nel seguente capitolo saranno messe a confronto le metodologie tradizionali, illustrando i casi in cui scegliere la metodologia Waterfall e Agile.

Caratteristica	Waterfall	Agile
Documentazione	Dettagliata su tutti gli aspetti del progetto	Richiesto solo il necessario
Cliente	Coinvolto solo durante la fase iniziale	Coinvolto durante tutta la durata del progetto
Membri del Team	Alta numerosità, Turnover frequenti	Bassa Numerosità, Esperti, collocati nello stesso luogo, team consolidato
Dimensioni del Progetto	grande	piccolo
Azienda	Grande	Piccola, Smart
Membri del Team	Molti	Pochi
Requisiti	Chiari, basso tasso di cambiamenti	Non chiari, frequenti cambiamenti
Criticità del sistema	Fallimento ha serie conseguenze	Non molto critico
Pianificazione	Lineare	Iterativa
Enfasi	Processo	Persone
Stile del Management	Autocratico	Distribuito

Misure di Performance	Conformità al piano	Valore apportato
-----------------------	---------------------	------------------

Table 1: Waterfall vs. Agile

L'approccio Waterfall è maggiormente appropriato per i progetti che possiedono chiari obiettivi e requisiti iniziali, per il quale il tasso di modifica sarà basso e il coinvolgimento dei clienti in fasi successive quella iniziale è superfluo; per questa ragione, è anche consigliato per i progetti in cui è richiesta una precisa documentazione formale in ogni momento del progetto a causa della necessità di continue e frequenti verifiche, anche di autorità esterne all'organizzazione, come i tipici progetti di costruzione o di ingegneria (Špundak, 2014). Inoltre, la metodologia è consigliata per grandi progetti in termini di numero di membri del team e di durata (Špundak, 2014)

Infine, come anticipato in precedenza, la scelta della metodologia Waterfall dipende anche dalle caratteristiche dell'azienda; in genere, l'organizzazione necessita di basso sforzo e poca preparazione per essere implementarla, in quanto solitamente combacia con i processi già in piedi, ed è solitamente preferita dalle aziende più grandi, perchè garantisce maggior controllo sul lavoro, e dai team i cui membri sono poco esperti, variano facilmente, oppure il cui project manager non è in contatto quotidiano (Špundak, 2014).

Per quanto riguarda l'approccio Agile, invece, è destinato soprattutto ai progetti creativi ed innovativi, caratterizzati da alti livelli di incertezza, obiettivi di progetto poco chiari e requisiti che cambiano frequentemente, come ad esempio i progetti di ricerca o sviluppo di nuovi prodotti innovativi (Špundak, 2014).

Per queste ragioni i progetti sono organizzati in modo iterativo, quindi sono maggiormente consigliati per progetti in cui il cliente è disponibile a partecipare durante tutta la durata del progetto e in cui è preferibile rilasciare prodotto funzionante da subito, come i progetti di sviluppo software (Špundak, 2014).

A causa della ridotta documentazione prodotta e dell'impatto del fattore umano e in particolare della comunicazione tra i membri del team, questo approccio è efficace se il personale del team è molto preparato, lavora in un luogo comune, i membri siano pochi e ci sia un basso ricambio (Špundak, 2014).

Infine, il metodo Waterfall è definibile plan-driven, poiché si concentra sullo scope del progetto mantenendo fissi i requisiti e modificando costi e tempi, mentre la metodologia Agile è di tipo value-driven, mantenendo fisso tempo e risorse, lasciando variare requisiti e quindi scope (Kisielnicki & Misiak, 2017).

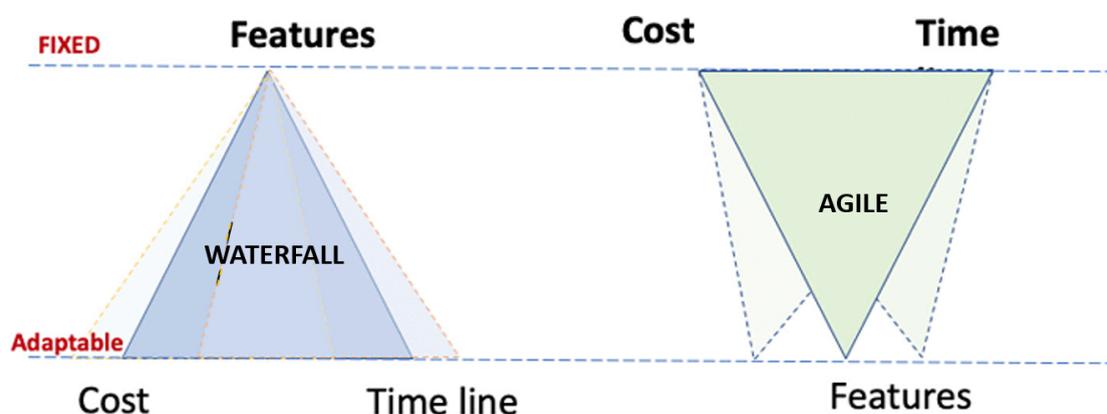


Figure 24: Waterfall vs. Agile

2.3.2. GESTIONE IBRIDA

Come visto, l'approccio Waterfall e Agile possiedono vantaggi e svantaggi, motivo per il quale non è immediato per il Project Manager decidere quale tecnica utilizzare, ma la sua valutazione sarà influenzata dal tipo di organizzazione, dalla disponibilità del cliente e dal tipo di progetto.

Al fine di sfruttare i vantaggi di entrambe le tipologie di progetto, la metodologia migliore potrebbe adottare una combinazione di elementi provenienti dagli approcci Waterfall e Agile, poiché l'adozione esclusiva di uno o dell'altro non si adatta perfettamente al progetto (Špundak, 2014). Le metodologie ibride partono da questo presupposto e, dopo aver individuato le modalità di utilizzo e le aree di integrazione tra i due modelli (Špundak, 2014), ne combinano le caratteristiche sulla base delle esigenze del progetto, al fine di creare una metodologia unica.

Ad esempio, un approccio ibrido potrebbe essere composto da un'attenta raccolta dei requisiti iniziale, con relativa stesura della documentazione, come per la metodologia waterfall, per poi essere seguita da una fase di sviluppo iterativa tipica della metodologia Agile.

3. METODOLOGIA COOLSHOP

Con il fine di validare la metodologia di Project Management utilizzata nell'azienda Coolshop, nel seguente capitolo sarà illustrato il suo funzionamento.

Al fine di valutare la metodologia, per le motivazioni spiegate in precedenza, è utile raccontarne il contesto di utilizzo, quindi l'organizzazione, i progetti seguiti e la tipologia di clientela.

3.1. DESCRIZIONE AZIENDA COOLSHOP

Coolshop è un'azienda di consulenza informatica nata nel 2010 a Torino. L'azienda ha dimensione media, con all'attivo circa 70 dipendenti; lavora solo su progetti su commissione per lo sviluppo di applicativi custom B2B o B2C. La strategia adottata dall'azienda la porta a concentrarsi su progetti che possono creare collaborazioni con i clienti per lunghi periodi di tempo, invece che su progetti one-off.

Nel processo di sviluppo software, Coolshop pone al centro dell'attenzione la User Experience, ossia la progettazione di un'esperienza utente che ne possa soddisfare le esigenze e lo aiuti a raggiungere i suoi obiettivi con facilità, con il fine di aumentare l'adozione del prodotto.

I contratti di progetto hanno sempre come forma di pagamento il prezzo fisso a corpo, stabilendo il prezzo in base al numero di ore totali preventivate che le risorse dell'azienda dedicano al progetto, dove le risorse hanno tutte lo stesso prezzo. Il prezzo della commessa è quindi calcolato sulla stima dell'effort necessario per completare la commessa.

La gerarchia aziendale è divisa su due livelli

- **Strategic Apex**

Composta dal CEO e dalla Guard, responsabili di risolvere i conflitti manageriali del livello inferiore e definire strategie e visione aziendale.

I membri di questo livello non sono allocati sui singoli progetti, ma si occupano di tutti, fornendo best practice, monitorando che date di consegna e qualità siano rispettate, occupandosi dello staffing e garantendo la visibilità cross-progetto, così da assicurare la condivisione di soluzioni che possono aiutare progetti diversi.

- **Operating Core**

I membri di questo livello sono divisi in dipartimenti diversi e allocati sui progetti.

- **Paladin**

I membri di questo dipartimento sono Project Manager e Analisti Funzionali; ai progetti è solitamente assegnato un solo Project Manager che supervisiona il lavoro di uno o più analisti funzionali, ma può capitare che entrambe le posizioni siano ricoperte da un singolo membro Paladin.

Il loro compito è quello di raccogliere i requisiti e i casi d'uso del cliente per indirizzare il disegno della soluzione, cogliendo opportunità di miglioramento e nuovi spazi commerciali da proporre al cliente.

Inoltre, interagisce con gli altri membri dell'azienda per il successo del progetto e stimola e coinvolge il team nelle scelte di Experience, organizzando meeting interni.

Infine, il Paladin è il maggiore responsabile del successo del progetto a cui è assegnato, quindi ne controlla i relativi tempi, costi e si accorda con il cliente per il rilascio delle funzionalità, testando la qualità dell'applicativo e formando il cliente.

- **Hero**

I membri di questo dipartimento sono Technical Leader e Developers; in ogni progetto è sempre presente un Technical Leader, che nei progetti più piccoli è anche l'unico Developer, ma in genere supervisiona il lavoro di altri Developers .

I Developers hanno la responsabilità di scrivere codice per l'applicativo, informare gli altri membri del team sullo sviluppo dei task a loro assegnati e, a volte, deployare il codice nei vari ambienti di sviluppo.

I Technical Leaders, invece, definiscono l'architettura tecnica della soluzione, dividono le funzionalità in task e li assegnano ai Developers, stimano l'effort necessario per le attività e bilanciano la qualità del codice con le esigenze del progetto.

- **Jedi**

I membri di questo dipartimento sono i Designers; il loro compito è quello di definire quale sarà la User Experience dell'applicativo, intervistando i clienti per delineare le Buyer Personas e fornendo visuals e mockup della soluzione.

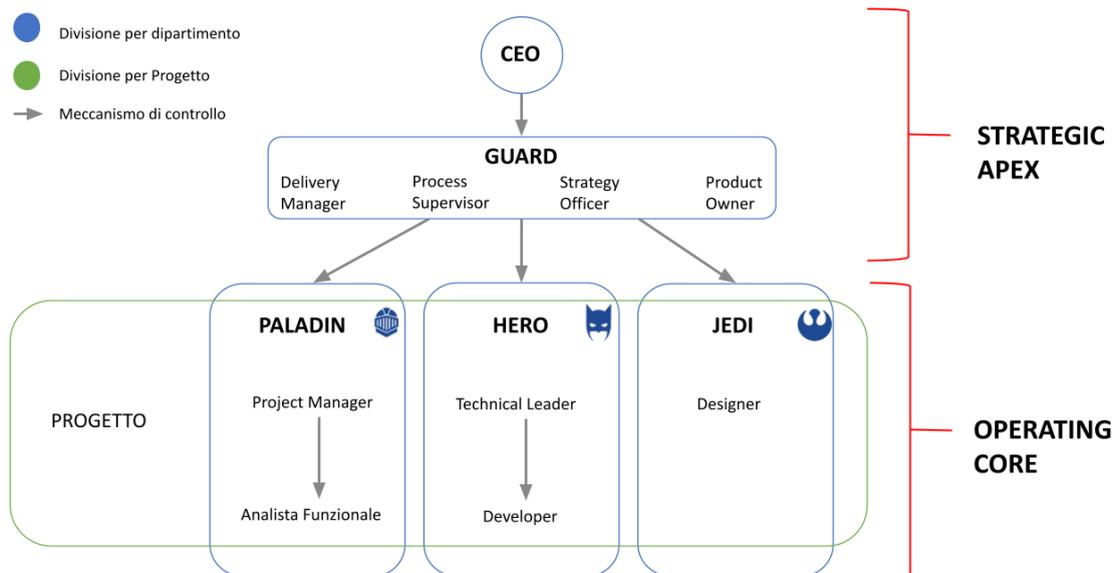


Figure 25: Organigramma Coolshop

3.2. NASCITA METODOLOGIA

Come detto in precedenza, le caratteristiche della metodologia ibrida devono adeguarsi sia all'azienda fornitrice che cliente, pescando dalla metodologia Waterfall o Agile a seconda delle necessità.

Partendo dall'idea che Agile possa essere la metodologia che porta maggior valore nel processo di sviluppo software, Coolshop ha tentato di adottare la metodologia Agile il più possibile, implementando quella Waterfall ove necessario.

Nei capitoli precedenti è stato esposto come l'implementazione di successo della metodologia Agile necessita di alto livello di competenze dei componenti del team di progetto e una formazione solida per tutta la sua durata. All'interno di Coolshop, la giovane età dell'azienda, le sue dimensioni ridotte e le scelte strategiche sulla gestione del personale dovute alla vicinanza di fucine di Developers, Project Managers e Analisti Funzionali come possono essere il Politecnico di Torino e l'Università degli studi di Torino, comportano disparità di esperienza e frequente turnover delle risorse allocate al progetto, fenomeno già tipico del settore informatico. Inoltre la lunga durata delle collaborazioni con i clienti comporta la necessità di trasferire efficacemente le informazioni di progetto ai futuri membri del team.

Affinché la metodologia di Project Management sia consistente con le scelte aziendali, è quindi necessario implementare caratteristiche Waterfall per la fase di Design e Implementazione, richiedendo una buona documentazione degli applicativi che possa supportare il lavoro dei developers e tramandare l'informazione nel caso di turnover del team, e rigidi processi di sviluppo delle funzionalità, a causa della necessaria supervisione del codice da parte del Technical Leader.

Non solo, la clientela di Coolshop difficilmente richiede software molto innovativi, perciò benché sia necessario mantenere una certa flessibilità dei requisiti perchè possono cambiare nel tempo, funzionalità simili sono già state implementate in precedenza. Per questa ragione non è difficile definire una stima dei tempi necessari per l'implementazione delle funzionalità e quindi dettagliare attraverso tecniche Waterfall un preciso cronoprogramma delle Release, che può essere modificato in corso d'opera.

Per questa ragione, a Coolshop è richiesto di adottare una metodologia value driven, adattando i requisiti del software sviluppato alle esigenze del cliente, ma aggiustando tempi e costi di consegna preventivati per rimanere plan driven.

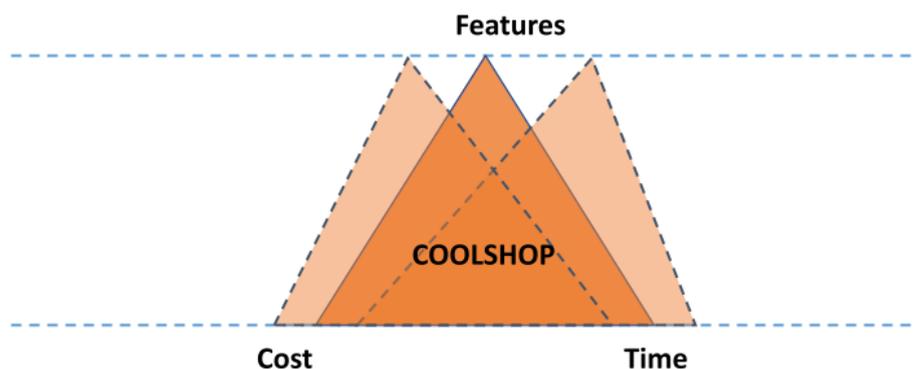


Figure 26: Coolshop value and plan driven

3.3. DESCRIZIONE DEL PROCESSO LAVORATIVO

Nel seguente capitolo sarà descritta la metodologia utilizzata dai team di Coolshop per la gestione dei loro progetti, operando un confronto con le metodologie tradizionali. cliente

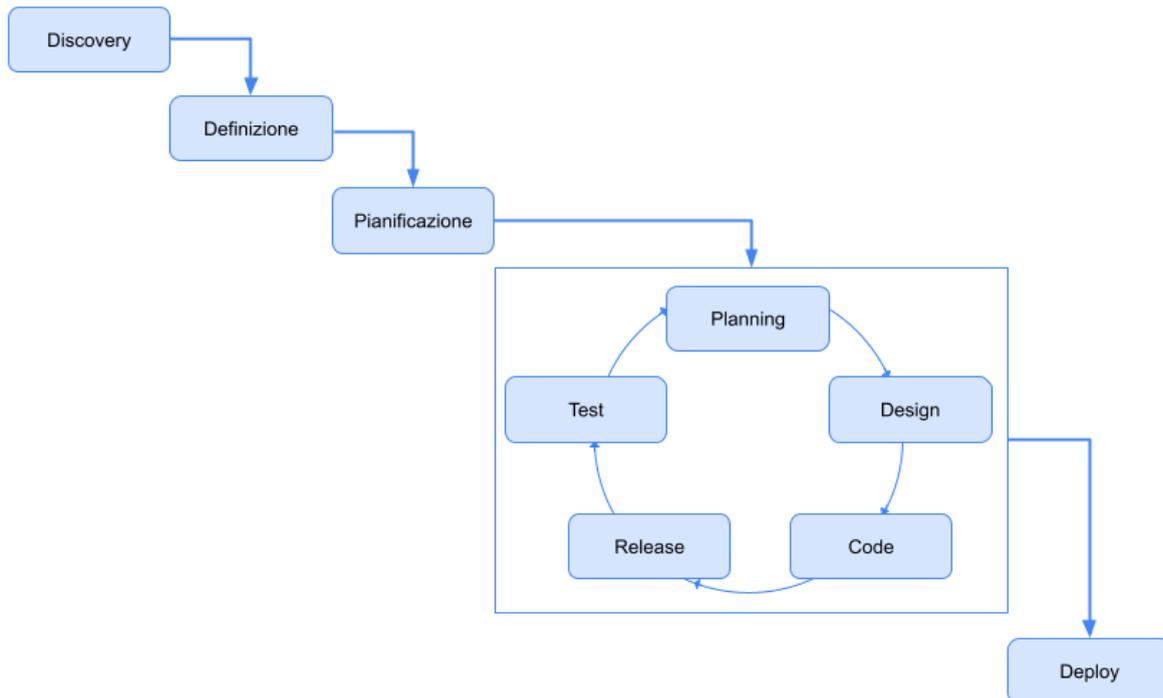


Figure 27: Fasi Metodologia Coolshop

3.3.1. DISCOVERY

Questa fase ha lo scopo di fornire in un paio di settimane una stima di massima di costi, tempi e fattibilità del progetto.

Inizialmente i membri del team di progetto si incontrano con il cliente al fine di definirne lo scope. Successivamente il Paladin del Team effettua una prima raccolta dei requisiti e ne definisce la relativa priorità di sviluppo; i requisiti sono successivamente esaminati dai membri dei vari dipartimenti, per fornire ad ogni funzionalità una stima di effort e i vincoli di implementazione.

Inoltre, in questa fase, il membro Jedi propone al cliente delle interviste per definire la Buyer Personas, grazie alla quale sarà possibile delineare una migliore User Experience nelle fasi successive.

3.3.2. DEFINIZIONE DEL PROGETTO

L'obiettivo di questa fase è quello di redigere lo Statement Of Work (SOW) del progetto, che definisce:

- Le finalità per le quali il progetto è avviato
- La descrizione di ciò che verrà incluso al suo interno
- Divisione delle funzionalità concordate in Wave di rilascio nell'ambiente di Produzione. Può capitare che il progetto sia composto da una sola Wave, perchè non si ritiene opportuno definire funzionalità che saranno sviluppate molto in là nel tempo, in quanto le esigenze di fondo potrebbero cambiare.
- Le risorse predisposte da Coolshop e dall'azienda cliente, con annessa matrice RACI
- Timeline preliminare, con data di inizio e fine delle Wave

- Primo intorno del Budget di progetto

In genere, all'interno del SOW viene concordato un Workshop all'inizio di ogni Wave, durante il quale sono decise nel dettaglio le funzionalità che saranno implementate, definendo così la Product Backlog.

La Product Backlog concordata risulta essere semirigida: le funzionalità in essa contenute non possono essere modificate, ma team di progetto e cliente potranno accordarsi per cambiare il dettaglio delle funzionalità, aggiungerne di nuove e modificare le caratteristiche di implementazione, rivedendo timeline e budget.

3.3.3. PIANIFICAZIONE DI ALTO LIVELLO

Basandosi sulla timeline preliminare e le funzionalità concordate all'interno della Product Backlog, in questa fase si definisce il preventivo del Budget di progetto e relativa timeline.

Le funzionalità presenti all'interno della Product Backlog necessarie al completamento di una wave sono scomposte dall'Hero in attività di alto livello, formando le Wave Backlog; per ogni macro-attività, Hero, Paladin e Jedi definiscono la stima di effort necessaria per la sua implementazione ed eventuali vincoli di implementazione, definendo la stima interna del progetto.

Successivamente, il Paladin in accordo con le esigenze del cliente e rispettando i vincoli di precedenza tra le funzionalità, definisce la Timeline di Progetto, dividendo le singole Wave in varie Release, a cui è assegnato lo sviluppo di alcune funzionalità.

	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE
WAVE 1										
RELEASE 1										
RELEASE 2										
RELEASE 3										
WAVE 2										
RELEASE 1										
RELEASE 2										

Figure 26: Waves Gantt Chart

Le stime interne di Budget e di tempo del progetto sono successivamente presentate al cliente evidenziando uno scarto percentuale, utile per assorbire l'effort necessario allo sviluppo di eventuali cambi di requisiti.

Si può facilmente notare che per la pianificazione di alto livello, la metodologia Coolshop è prettamente Waterfall, definendo date di consegna rigide, e fornendo una documentazione vincolante per entrambe le parti.

3.3.4. WAVE

Come detto in precedenza, l'intero progetto è diviso in Wave, il cui scopo è il rilascio nell'ambiente di Produzione di un nuovo applicativo, o una sua versione aggiornata contenente nuove funzionalità. L'ambiente di produzione è l'ambiente informatico nel quale il software viene utilizzato dagli utenti finali.

Solitamente le Wave hanno durata di circa nove mesi, motivo per cui la loro iterazione permette di rimanere flessibili sulla pianificazione del progetto: qualora dovessero cambiare le condizioni interne o esterne di Coolshop o del cliente, alla fine della wave si potrà riorganizzare quella successiva.

Le Wave si dividono in Release, il cui obiettivo è il rilascio di specifiche funzionalità nell'ambiente di test del cliente

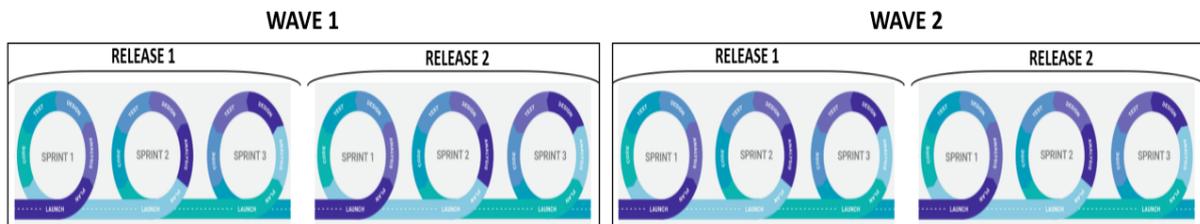


Figure 28: Project Waves

All'inizio di ogni Wave si definiscono quali sono le macro attività da sviluppare in ogni Release, definendo quali saranno i deliverable; viene definita così la Release Backlog. Alcune attività di supporto al team di progetto sono assegnate al cliente, necessarie per la corretta implementazione della funzionalità; ad esempio, ipotizzando che un'attività del team sia l'integrazione del catalogo dei prodotti del sito con il Product Information Management (PIM) del cliente, gli saranno richieste le API necessarie.

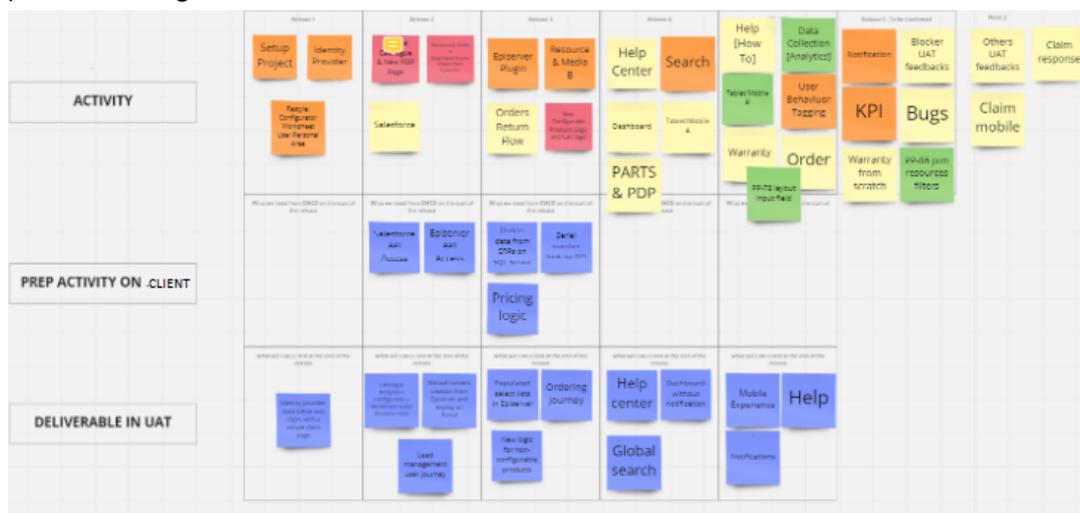


Figure 29: Project Releases

3.3.5. RELEASE

Come già anticipato, le Wave sono divise in Release che possono durare da 1 a 3 mesi. Ogni Release ha come obiettivo il rilascio di alcuni deliverable; una Release potrà dirsi terminata solo quando tutti i deliverable saranno stati consegnati al cliente.

Le Release sono a loro volta suddivise in Sprint, all'interno delle quali avvengono le attività di raccolta dei requisiti, design, sviluppo e test della funzionalità.

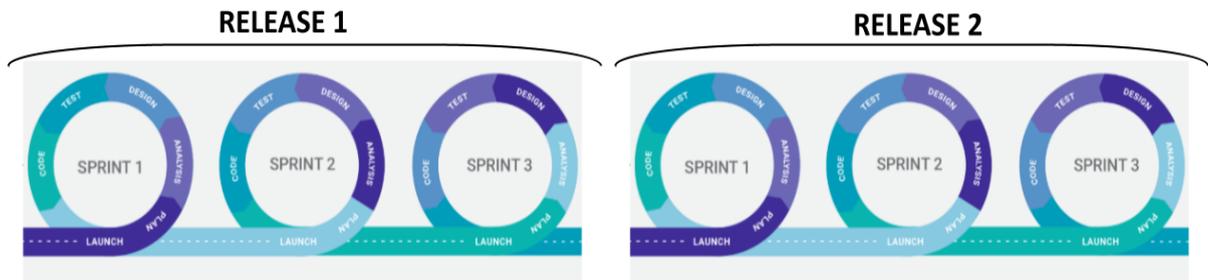


Figure 29: Project Waves

All'inizio di una Release, il Paladino di progetto, sulla base del tempo a disposizione per la Wave e all'effort necessario per completare la Release, si accorda con lo Strategic Apex dell'organizzazione per decidere se rimuovere o aggiungere risorse al team di progetto. Inoltre, durante questa fase il Paladino determina quali attività della Wave Backlog saranno svolte durante i vari Sprint.

Alla fine della Release, il Paladino e il cliente si incontrano per discutere del suo andamento, applicando eventuali azioni correttive sul piano delle Release future. Inoltre, durante questo periodo il team di progetto si riunisce per una cerimonia chiamata Release Retrospective, al fine di esaminare come è andata la Release in termini di interazioni tra individui, processi e strumenti, pianificando modi per migliorare la qualità e l'efficacia del team.

Si può notare che a questo livello di profondità l'iterazione delle Release sfrutti le caratteristiche della metodologia Agile, dando la possibilità al cliente di richiedere modifiche delle funzionalità, o della loro priorità, nel caso di cambiamenti all'interno o all'esterno dell'organizzazione e al team di ragionare sulla qualità del proprio lavoro.

3.3.6. SPRINT

All'interno degli Sprint avviene lo sviluppo vero e proprio delle funzionalità. Solitamente lo Sprint non dura più di quattro settimane.

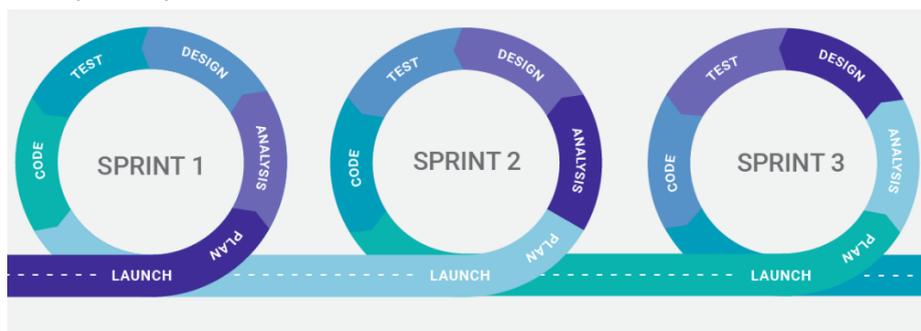


Figure 30: Project Sprints

All'inizio di ogni Sprint, il membro Paladino e l'Hero di progetto si riuniscono in uno Sprint Planning Meeting durante il quale riepilogano le attività da portare a termine durante lo Sprint e decidono a chi assegnarle.

Durante tutta la durata dello Sprint, il team di progetto si riunisce una volta al giorno per una cerimonia chiamata Stand Up Meeting, durante la quale ogni membro del team spiega cosa

ha fatto la giornata precedente e cosa farà durante la giornata corrente, chiedendo aiuto in caso di necessità e dichiarando se hanno novità che possono impattare il lavoro degli altri.

Per ogni funzionalità, il membro Paladin sviluppa un Functional Design, che descrive nel dettaglio la specifica funzionalità dell'applicativo e, se necessari, conterrà i visuals prodotti dallo Jedi. Dopo che il Functional Design è stato approvato dall'Hero e dal cliente, i task necessari alla sua implementazione sono creati all'interno di un'unica epica.

Con il fine di avere una rapida panoramica sullo stato del progetto, i task sono raccolti all'interno di una Kanban Board; ogni volta che un task cambia di stato, viene spostato da una colonna a quella successiva. La priorità del task è definita dall'altezza in cui è posizionata sulla Kanban Board; quindi, ogni volta che un membro del team deve pescare da una determinata colonna, prende il task più in alto a lui assegnato.

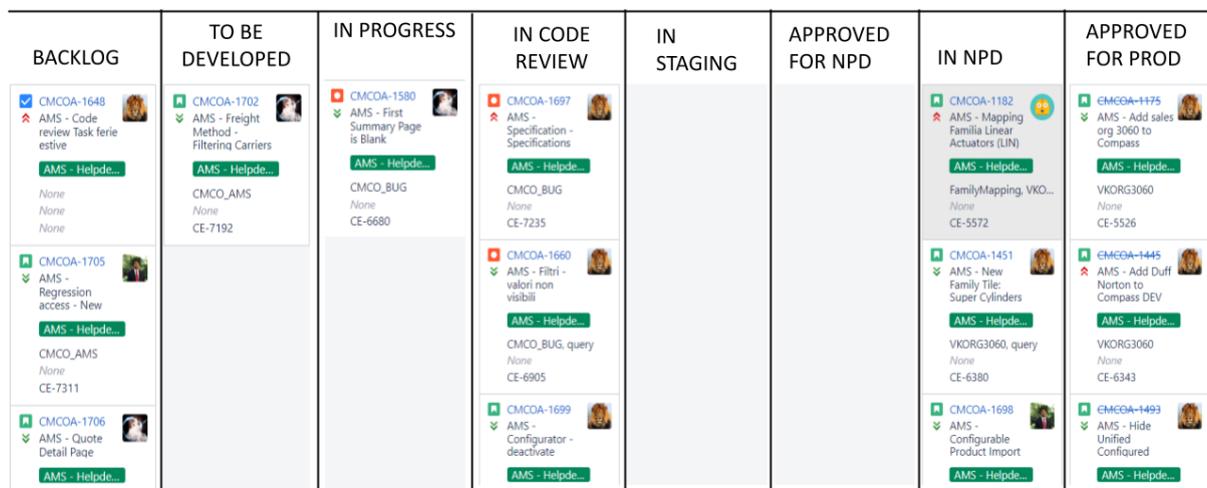


Figure 31: Coolshop Team Kanban Board

I task seguono tutti lo stesso processo, descritto di seguito.

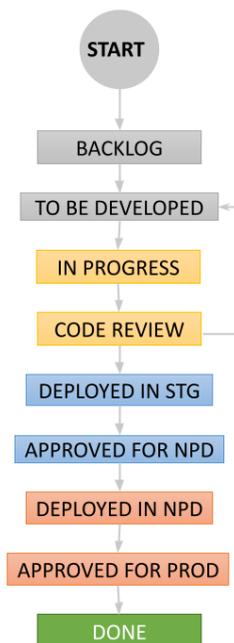


Figure 32: Processo di sviluppo dei task

Il Paladin si occupa della creazione dei task all'interno della Backlog; ogni task è composto da una descrizione della funzionalità da implementare e dai test case che lo sviluppato deve superare per testimoniare che i requisiti del cliente siano stati rispettati e l'integrazione con il resto del sistema sia andata a buon fine.

L'Hero legge la funzionalità da implementare descritta all'interno del task nella Backlog, ne esegue l'analisi tecnica e la assegna ad una risorsa developer, etichettando il task in To Be Developed.

Ogni volta che i devs iniziano lo sviluppo di un nuovo task, applicano la label "In Progress" a quest'ultimo.

Con il fine di rilasciare codice di qualità, il codice è testato su più ambienti prima che raggiunga l'ambiente di Produzione, dove è visibile agli utenti finali.

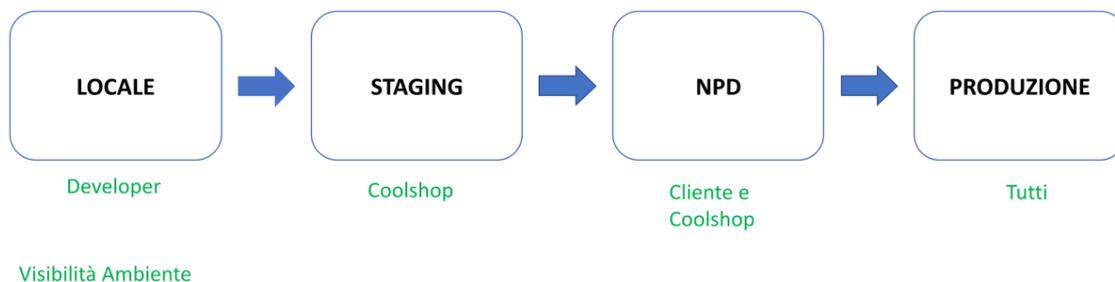


Figure 33: Ambienti di Sviluppo Software

Una volta che lo sviluppo del task è stato portato a termine, passa in "Code Review" e assegnato all'Hero, che ha la responsabilità di controllare il codice sviluppato, richiedendo eventuali correzioni, e quindi rimandando il task in To be Developed, altrimenti rilasciandolo nell'ambiente di Staging.

In questo ambiente è responsabilità del Paladin testare le funzionalità, seguendo i test case redatti all'interno del task. Se le condizioni poste in essere dai test case sono rispettate, il Paladin approva il task per l'ambiente NPD, dove sono rilasciati tutti i task alla fine dello sprint.

L'ambiente NPD replica l'ambiente di Produzione, all'interno del quale il cliente può testare l'applicativo prima che sia mostrato all'utente finale. L'ambiente rappresenta il primo contatto del cliente con l'applicativo funzionante ed è per lui fondamentale per capire se i requisiti richiesti corrispondono alle sue reali esigenze. In questo modo, il cliente può fornire feedback e possono essere apportate eventuali modifiche prima che l'applicazione sia effettivamente rilasciata all'utente finale.

Con il fine di rimanere allineati sulla correttezza delle funzionalità implementate, il Paladin redige un Testbook che definisce quali azioni compiere sull'applicativo per testare la funzionalità e i relativi risultati che ci si aspetta di ottenere.

Se le funzionalità soddisfano i requisiti del cliente, saranno approvate per l'ambiente di produzione, dove saranno rilasciate alla fine della Wave.

La rigidità del processo di sviluppo dei task è dovuta alla diversa esperienza e capacità dei membri del team di Coolshop; i task devono essere supervisionati dall'Hero di progetto in

quanto unico membro del team capace di valutare la consistenza del codice con il resto del sistema.

Alla fine di ogni Sprint, il Paladin si riunisce con il cliente per riepilogare ciò che è stato fatto all'interno dello Sprint, definendo eventuali scostamenti dal piano iniziale e definendo eventuali azioni correttive da svolgere durante lo Sprint successivo, modificando eventualmente la timeline di progetto.

Inoltre, il Paladin redige e modifica il file di budget, notificando variazioni tra budget preventivato ed erogato.

3.4. STRUMENTI AZIENDALI

Nel capitolo che segue si illustreranno gli strumenti forniti a supporto del Project Manager.

- **Jira**
Piattaforma di gestione delle attività a supporto dei Team.
Il software viene utilizzato per creare, assegnare, tracciare e monitorare le attività di progetto.
Su Jira ogni progetto funge da contenitore di task dedicati, che possono essere organizzati in vari modi, ad esempio assegnandoli a delle epiche.
L'applicativo risulta particolarmente utile perchè ogni utente può loggare sul task il numero di ore che ci ha dedicato, quindi è facile risalire al budget erogato mensilmente sul progetto e misurare l'efficienza del singolo e del team.
- **File di Staffing**
Documento condiviso tra tutti i componenti dell'azienda, utilizzato per pianificare e assegnare le risorse ai progetti, definendone l'effort che dovrà erogare sul progetto per il mese specifico.
Il file è un valido strumento per pianificare il carico di lavoro per ogni membro del team, identificare eventuali lacune di risorse e, quindi, rischi per il progetto.
- **File di Budget di Progetto**
Durante la fase di Definizione del Progetto, il file è utilizzato per definire l'effort necessario per ogni macro funzionalità da sviluppare, per poi risalire al budget complessivo del progetto.
Successivamente, il documento è utilizzato per monitorare l'andamento del progetto in termini di budget, misurando il discostamento tra effort erogato e preventivato.
- **Miro**
Miro è un software di collaborazione visiva utilizzato per condividere idee all'interno del Team.
Viene utilizzato per visualizzare quali sono le funzionalità da svolgere per ogni release e condividere opinioni su come migliorare l'efficacia all'interno del gruppo
- **Google Calendar**
Lo strumento permette ad ogni membro di Coolshop di verificare la disponibilità giornaliera dei colleghi, fornendo la possibilità di organizzare facilmente i meeting.

4. CASO DI STUDIO - PROGETTO PARTNER PORTAL

Nel presente capitolo sarà analizzato il progetto “Partner Portal”, usato come caso di studio reale della metodologia Coolshop, per esaminare in dettaglio come è stata utilizzata per soddisfare le esigenze del cliente, raggiungere gli obiettivi di progetto e gestire le risorse.

Il progetto Partner Portal è stato commissionato dall’azienda statunitense Columbus Mckinnon Corporation (CMCO) dopo una fase di gara; Coolshop aveva già lavorato con CMCO per il progetto Apollo, il cui obiettivo era quello di apportare migliorie sul loro CPQ, utilizzato da distributori, venditori e utenti interni CMCO . Un applicativo CPQ (Configure, Price, Quote) consente agli utenti di configurare prodotti in base alle proprie esigenze, stabilirne il prezzo e di generare un preventivo.

Il caso di studio è descrittivo della metodologia messa in piedi da Coolshop, in quanto permette di illustrarne l’intero processo e il modo in cui è di aiuto per il Project Manager e l’organizzazione tutta.

Inoltre, è stato deciso di utilizzare un singolo caso di studio perché rappresentativo dei progetti gestiti in Coolshop, in quanto la metodologia adottata è standardizzata per ogni progetto.

Nei capitoli successivi sarà descritta l’azienda cliente e il Progetto Partner Portal.

4.1. DESCRIZIONE AZIENDA CLIENTE CMCO

Columbus McKinnon Corporation è un’azienda statunitense che si occupa della progettazione, produzione e vendita di attrezzature e tecnologie per la movimentazione dei materiali, per il sollevamento, la trazione e l’ancoraggio dei carichi come verricelli, catene, cavi, gru e sollevatori. Inoltre, offre servizi di assistenza tecnica, riparazione e manutenzione per garantire che le attrezzature funzionino in modo sicuro ed efficiente.

Il catalogo dell’azienda è composto da prodotti configurabili, pertanto il cliente ha la possibilità di decidere le caratteristiche del prodotto, ad esempio la sua capacità di sollevamento, e prodotti non configurabili, ad esempio i pezzi di ricambio.

Nel tempo CMCO ha acquisito diverse aziende concorrenti. Per scelte strategiche dell’azienda cliente, l’identità dei vari brand è rimasta divisa, quindi i clienti dell’azienda facevano riferimento a diverse sezioni di essa. Questa struttura organizzativa è riflessa sull’organizzazione dei loro applicativi, difatti esistono diversi Dealer Portal a seconda del brand di riferimento.

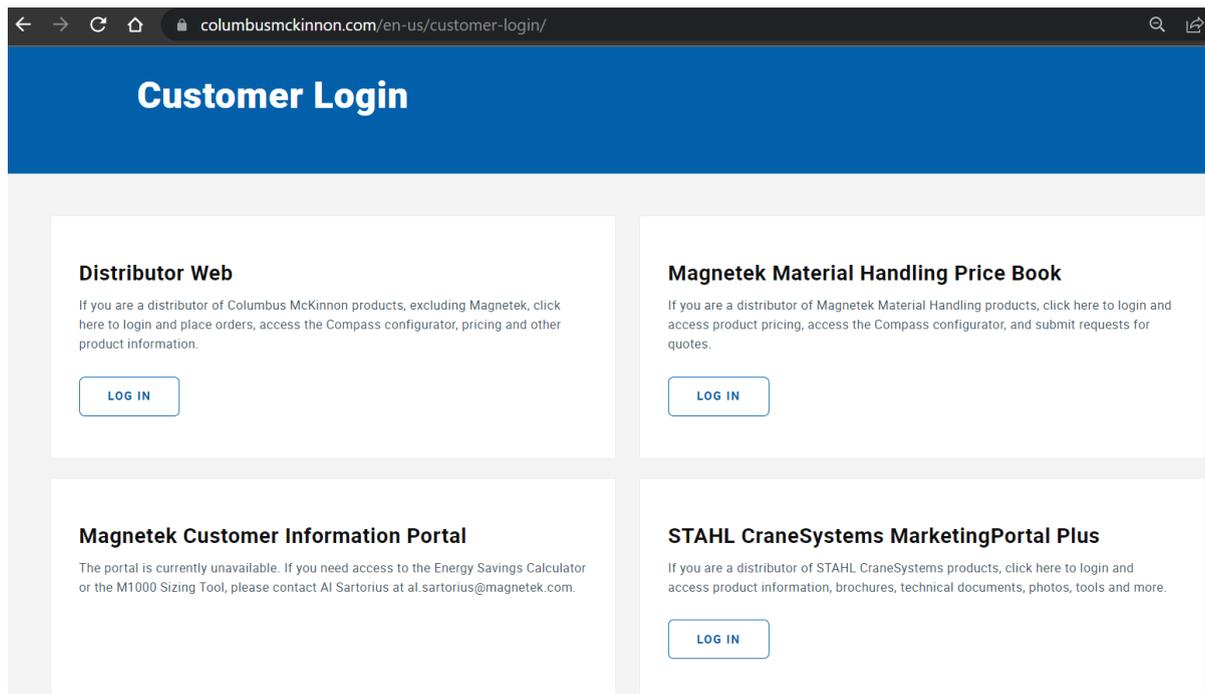


Figure 34: CMCO Entry Point Portali

Inoltre, la struttura organizzativa di CMCO è divisa anche per mercati: NA (North America), LATAM (LATin AMerica) e EMEA (Europe, Middle East and Africa). I brand non lavorano su tutti i mercati e non commercializzano in ogni mercato gli stessi prodotti.

Infine, l'azienda divide l'utenza all'interno dei propri siti web in tre utenze:

- Distributor: Compra da CMCO e la rivende al cliente finale.
- SalesMan: Venditore di CMCO che si interfaccia direttamente al cliente finale.
- Internal: utente interno di CMCO.

4.2. DESCRIZIONE DEL PROGETTO

In fase di gara, il requisito richiesto da CMCO era la costruzione di un Partner Portal che potesse convergere in un'unica esperienza i brand, i mercati e gli attori dell'azienda, che risultava fino ad allora piuttosto frammentata. Le aziende partecipanti sarebbero state giudicate sulla base di tempi, costi e User Experience ideata nell'offerta.

Il Partner Portal è un applicativo web riservato a due tipi di utenti: quelli interni, ad esempio gli specialisti di prodotto che utilizzano il software per validare le diverse configurazioni dei prodotti, e quelli esterni, partner dell'azienda. Questi ultimi sono divisi per ruolo, brand e area geografica, e l'applicazione consente loro di accedere ai registri di transazioni, risorse di marketing, informazioni sul prezzo e sulle vendite di prodotti e servizi, dettagli tecnici e supporto fornito dall'azienda.

Con il fine di svolgere la sua funzione professionale, il Partner Portal necessita di poggiarsi su piattaforme tecnologiche dotate di automazioni che accelerino azioni di routine, e di integrarsi con eventuali infrastrutture IT preesistenti e con app per estendere le proprie funzionalità.

Solitamente, gli utenti partner hanno a disposizione dashboard all'interno del Partner Portal che permettono di consultare e condividere le informazioni sul catalogo dei prodotti, come

documenti tecnici, contratti, listini prezzi, nonché di partecipare a programmi di formazione, webinar ed eventi.

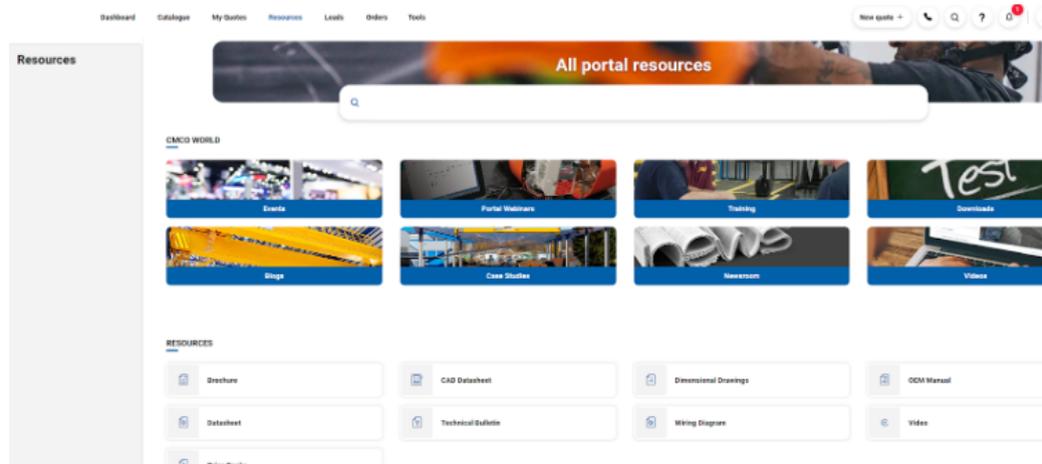


Figure 35: Pagina delle risorse Partner Portal

Infine, il partner portal deve essere un punto di riferimento per tutti i partner dell'azienda e li deve supportare con assistenza e coinvolgimento costanti; infatti, sono solitamente implementati strumenti di social networking, che risultano essere un ottimo mezzo per condividere notizie, report, casi di studio, normative e per diffondere a tappeto risorse di particolare rilevanza e urgenza.

4.2.1. DEFINIZIONE DEL PROGETTO

Come si legge all'interno del SOW firmato da CMCO, la soluzione proposta da Coolshop sarebbe partita dall'applicativo Compass, migliorandone le performance ed effettuando un restyle, e avrebbe aggiunto:

- Tutte le funzionalità presenti negli altri portali.
- Integrazione di un CMS (Optimizely) per il materiale di Marketing.
- Integrazione di un CRM (Salesforce) per approfondimenti commerciali.
- Customer Experience unica basata sull'analisi del comportamento degli utenti.
- Esperienza dell'utente "Seamless", in cui l'utente non si rende conto che le funzionalità appartengano a tools differenti (Configuratore, CMS, CRM...), immaginando che siano proprie del sito.

La pianificazione iniziale del progetto prevedeva un'unica Wave di 10 mesi, con una contingency di un mese, che sarebbe iniziata a metà novembre 2021 e terminata a metà settembre 2022.

La Wave sarebbe stata divisa in 5 Release, all'interno delle quali si sarebbero svolte attività sia lato Coolshop che lato CMCO. Ogni release sarebbe stata composta da una fase iniziale di design, durante la quale veniva redatta la documentazione e creati i Visuals per ogni funzionalità, seguita dalla fase di implementazione, per finire con una fase di testing, eseguita prima internamente e poi dal cliente.

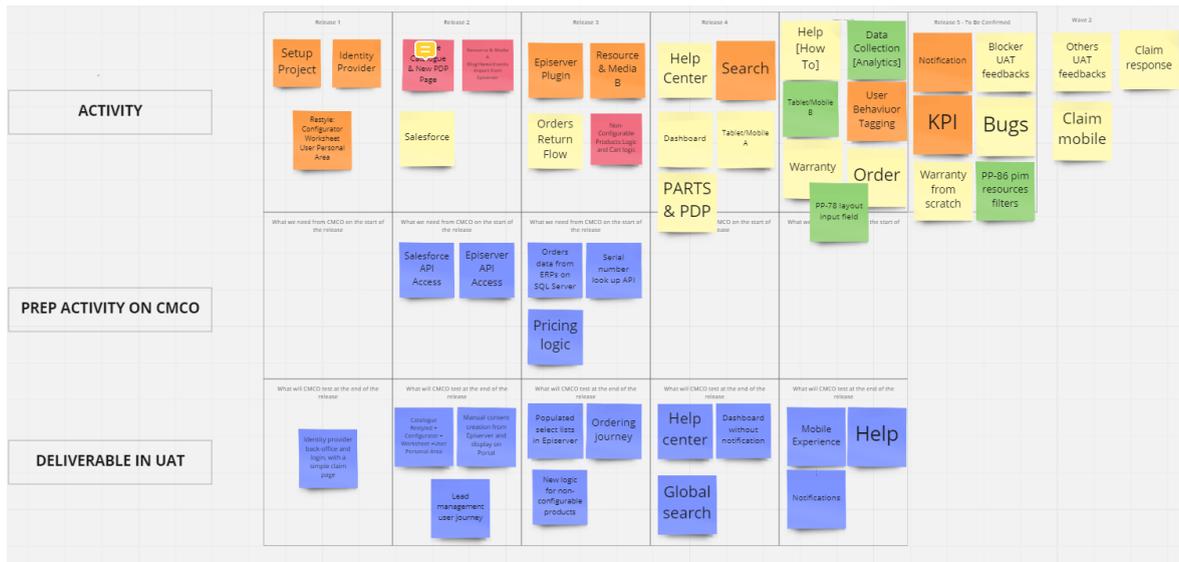


Figure 36: Project Releases

Il progetto si sarebbe concluso con il go live del Partner Portal e quattro settimane di Post Go Live Support.

Proprio a causa della lunga durata del progetto, è stato deciso che funzionalità da aggiungere in wave successive sarebbero state oggetto di futuri progetti, perchè eventuali soluzioni sarebbero potute diventare obsolete ancora prima della loro implementazione.



Figure 37: Prima Programmazione Gantt Chart

Coolshop pianificò un Workshop da CMCO dalla durata di una settimana, così da esplorare nel dettaglio tutte le funzionalità che il Partner Portal doveva contenere. Il workshop avrebbe avuto la durata di 5 giorni, così suddivisi:

- 2 giorni per la definizione della User Experience, attraverso interviste effettuate con i Partners, il team di Prodotto e il team di Marketing per ogni brand.
- 1 giorno per l'analisi dell'infrastruttura IT e decidere come sarebbe avvenuta l'integrazione con il sistema, da condurre con specialisti di infrastruttura
- 1 giorno per l'analisi di prodotti e configurazioni, da condurre con il team di prodotto
- 1 giorno per impostare la gestione del progetto, da condurre con Project Champion, il cui ruolo è spiegato in seguito.

Per garantire il successo del progetto, durante tutta la durata del progetto e in particolar modo durante i workshop, Coolshop ha richiesto al cliente la partecipazione di quattro figure chiave essenziali.

- Project Leader: una risorsa IS responsabile delle decisioni operative e tattiche durante le operazioni quotidiane.
- Project Champion: un IS leader, coinvolto nella definizione delle decisioni strategiche del progetto, responsabile del processo di approvazione tecnica del design della soluzione e del test di accettazione.
- Business Process Owner (BPO): leader del settore commerciale, responsabile del processo di configurazione e quotazione dei prodotti per i distributori. Sono responsabili del coordinamento di eventuali conflitti sul lato commerciale e del coinvolgimento di tutti i BPE interessati.
- Business Process Expert (BPE): Una o più figure esperte nel processo commerciale impattato dall'applicazione. Hanno la responsabilità di spiegare a Coolshop i processi da loro gestiti e di valutare gli impatti che il Solution Design proposto ha su di essi, consigliando il BPO sulla loro approvazione.

È facilmente intuibile che queste figure sarebbero state utili alla corretta integrazione del software fornito da Coolshop con i preesistenti sistemi informativi del cliente, ma anche alla definizione della corretta User Experience, definendo processi e funzionalità utili all'utente.

Il budget preventivato per il progetto era di 1060 MD (Man Days) e presentava un margine di variazione del $\pm 25\%$.

Le contingency di budget e durata del progetto sono state decise di comune accordo e risultavano vantaggiose per entrambe le parti; in questo modo, Coolshop metteva a disposizione per CMCO budget e tempo necessari per lo sviluppo di eventuali nuove funzionalità e Coolshop avrebbe ridotto il rischio di ritardare il rilascio del progetto. In ogni caso, contingency troppo larghe sarebbero risultate uno svantaggio per Coolshop, che preferisce di gran lunga terminare velocemente i progetti per allocare le risorse su di nuovi, aumentando così la redditività.

4.2.2. ESECUZIONE DEL PROGETTO

CMCO firmò il SOW un mese più tardi di quanto preventivato all'interno del SOW. Nonostante ciò, a causa di strategie commerciali di promozione del Portal, CMCO richiese di mantenere l'obiettivo di consegna del progetto entro il 26 Settembre 2022.

Come detto in precedenza, la metodologia di Project Management utilizzata da Coolshop per guidare i propri progetti fa uso di quella Waterfall per le attività di alto livello. Ciò ha permesso all'organizzazione di conoscere in anticipo l'allocazione delle proprie risorse tra i vari progetti e controllare la disponibilità per la richiesta del cliente, allocando un maggior numero di risorse per una durata ridotta.

Coolshop rispose affermativamente alla richiesta, modificando la timeline iniziale di progetto.

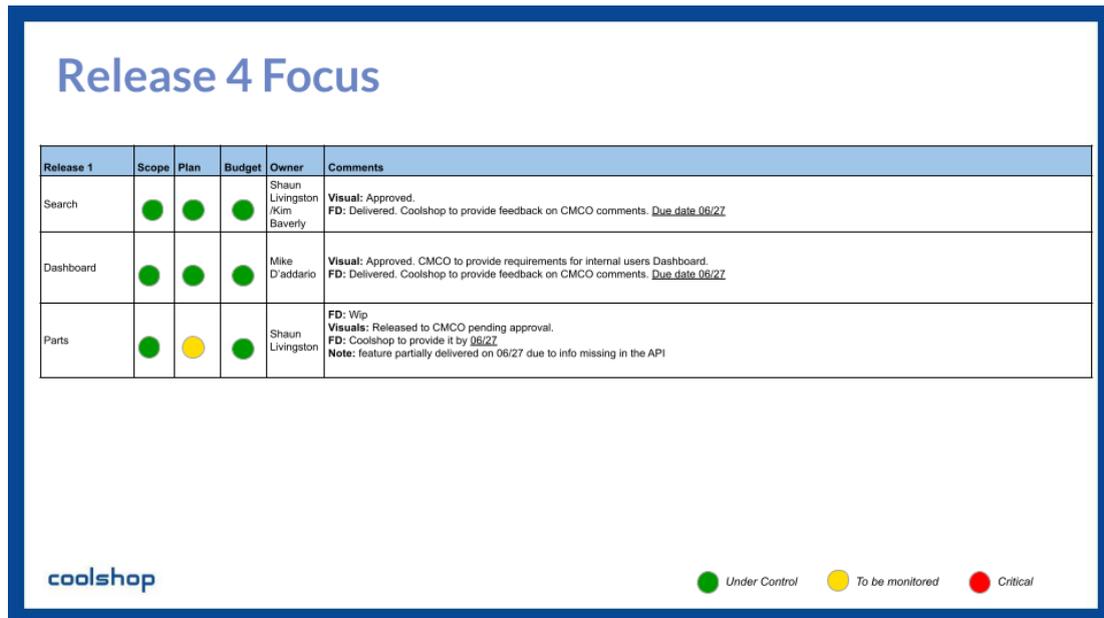


Figure 39: Presentazione settimanale - Monitoraggio

Inoltre, durante questi meeting Coolshop ha esposto l'effort utilizzato per i cambi di requisiti avvenuti durante l'esecuzione del progetto, definendo l'effort rimanente al cliente per eventuali nuove richieste.

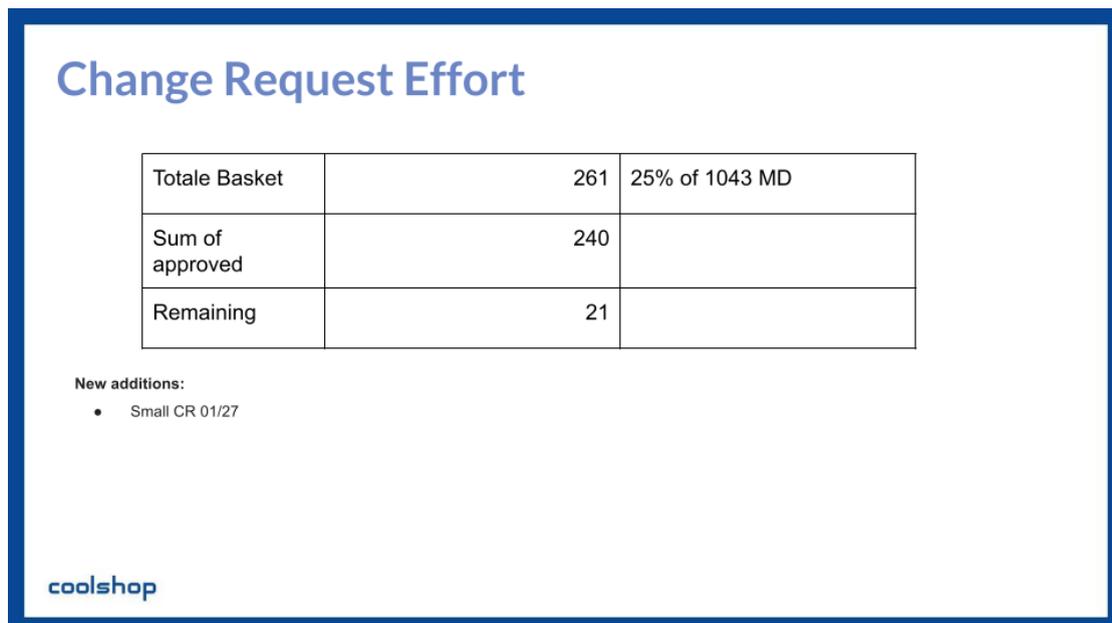


Figure 40: Meeting settimanale - Budget rimanente

Durante il periodo conclusivo del progetto, CMCO ha condotto una fase di test per far provare l'applicativo ad alcuni distributori selezionati. Da questa fase è emerso che gli utenti necessitassero di alcune funzionalità presenti sui vecchi portali, che CMCO aveva deciso di implementare in Wave successive; per questa ragione, CMCO ha richiesto lo sviluppo di nuove funzionalità prima del Go Live dell'applicativo.

La metodologia adottata da Coolshop è quindi risultata utile al cliente per fargli comprendere meglio quali fossero le loro reali esigenze.

La scelta di accettare o meno la richiesta era a totale discrezione di Coolshop, in quanto il budget di progetto era ormai terminato.

Anche in questa situazione la metodologia utilizzata da Coolshop è stata di aiuto all'organizzazione per prendere una decisione riguardo la richiesta del cliente. Attraverso una valutazione interna della disponibilità delle risorse, è stato deciso che la richiesta sarebbe stata accolta, ma sfruttando un gruppo ridotto del team di progetto e sostituendo il Project Manager. Per questa ragione, le nuove funzionalità sono state raccolte all'interno di una nuova release, che sarebbe iniziata al termine della release precedente.

In seguito a questa decisione, è stato stanziato un nuovo budget di 302 Man Days e rimandato il Go Live del progetto di 5 mesi.

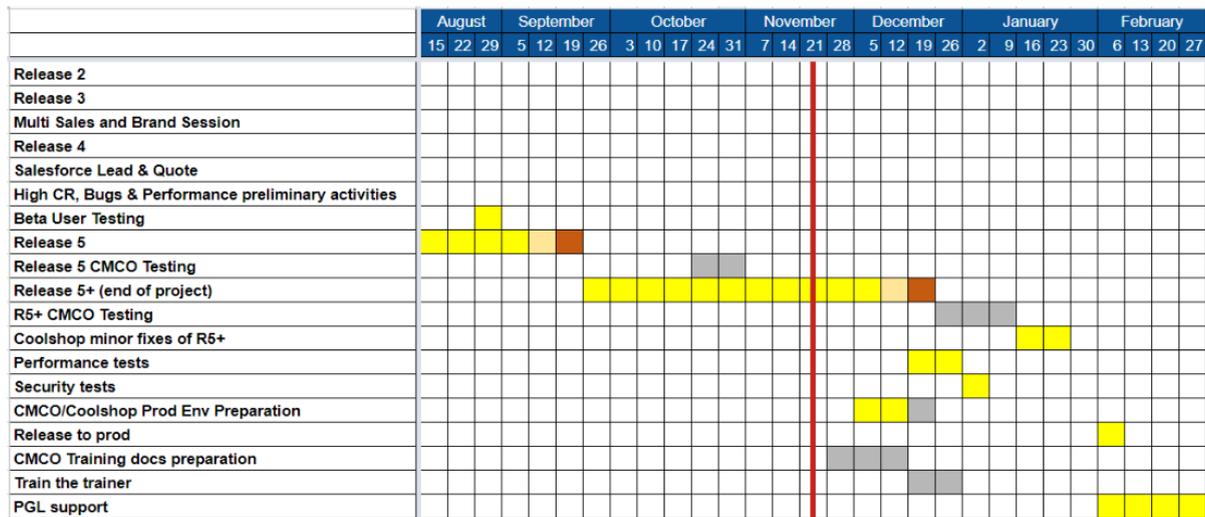


Figure 41: Gantt Chart Release 5+

L'ultima Release durò un mese in più rispetto quanto preventivato, per ragioni che saranno documentate successivamente nel capitolo riguardante le interviste.

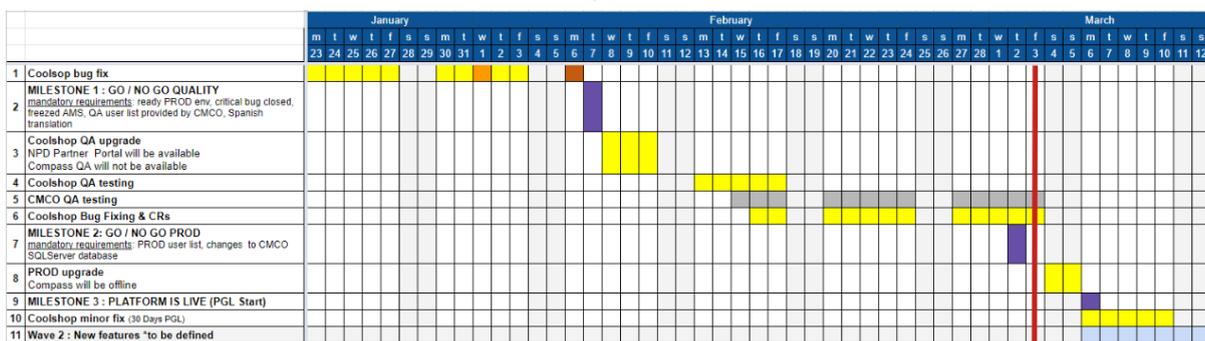


Figure 42: Gantt Chart Finale

Riepilogando, utilizzando la metodologia Coolshop il progetto è consistito in:

- 4 mesi di Discovery
- 1 mese di pianificazione di alto livello
- 15 mesi di esecuzione

4.2.3. RISULTATI

In questa sezione saranno brevemente descritti i risultati del progetto, in modo da introdurre il successivo capitolo riguardanti le interviste ai membri del team di progetto, dove saranno esplorate le motivazioni alla base dei risultati ottenuti.

Lo scope del progetto si è chiaramente allargato: non solo il cliente ha richiesto micro-evolutiva durante l'esecuzione del progetto, per le quali sono state usate tutte le giornate extra, ma ha richiesto un'intera nuova Release senza la quale il prodotto sarebbe stato considerato insoddisfacente.

È quindi chiaro che il cliente, durante lo sviluppo del progetto, grazie ad un prodotto funzionante che potesse essere utilizzato da utenti interni all'azienda e clienti campione da cui raccogliere le opinioni, ha compreso quali fossero le sue reali esigenze.

È evidente quindi che la fase iterativa della metodologia Coolshop è stata utile per ottenere un prodotto di buona qualità con tempistiche inferiori; se la metodologia fosse stata prettamente Waterfall, il cliente si sarebbe resa conto solo a prodotto completato che questo era manchevole di funzionalità importanti.

Per quanto riguarda il budget di progetto, come mostrato in figura 43, le giornate erogate da Coolshop risultano essere complessivamente maggiori rispetto quelle vendute a preventivo. Le motivazioni per le quali ciò è accaduto saranno dettagliate all'interno del capitolo delle interviste.

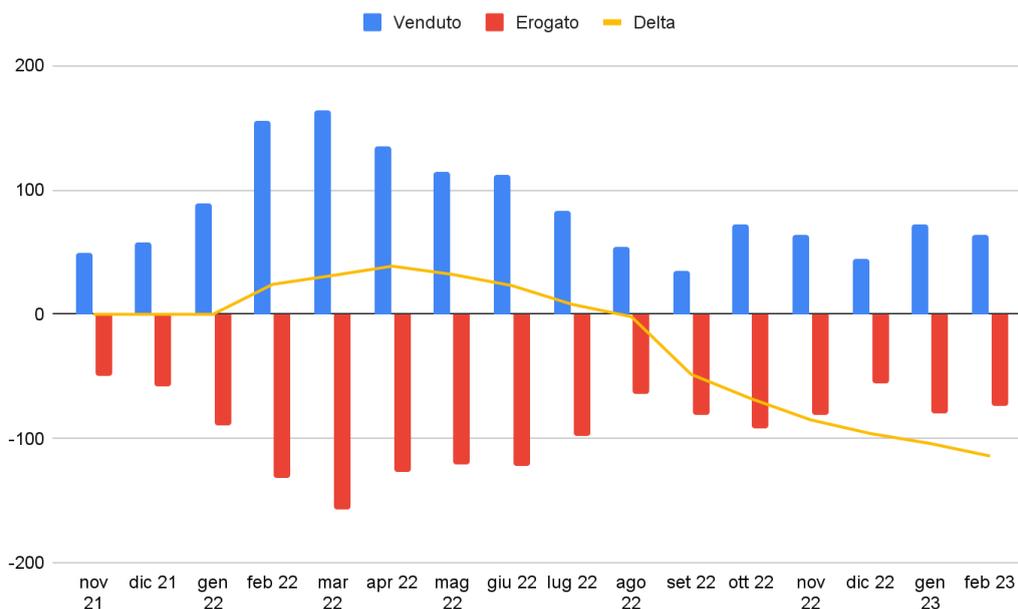


Figure 43: Budget Preventivato vs. Consuntivo

4.2.4. UTILIZZO DELLA METODOLOGIA DI PROJECT MANAGEMENT

Come descritto in precedenza, la metodologia adottata dal team di Coolshop risulta avere un approccio Waterfall per la pianificazione ad alto livello del progetto e uno più Agile a livello operativo.

L'utilizzo della metodologia Waterfall ad alto livello ha permesso a Coolshop di soddisfare le esigenze commerciali del cliente, fornendo una stima di massima del budget e dei tempi di consegna. Questo ha anche agevolato una comunicazione chiara e trasparente tra le due organizzazioni, che sapevano sempre cosa era previsto e quando doveva essere fatto. Inoltre, ha permesso a Coolshop di gestire con maggiore efficienza le risorse del progetto, regolamentando l'incremento o la riduzione delle risorse del team in base alle esigenze del progetto durante lo sviluppo; dal momento che per un determinato periodo di tempo sono state pianificate attività su un certo progetto, Coolshop blocca le risorse interne necessarie per fornire la capacità di effort promesse.

Difatti, dalla timeline è possibile notare come la fase iniziale, di costruzione del progetto, e quella finale, di smantellamento, necessitano di minor effort, e il cronoprogramma dettagliato permetteva di identificare esattamente quali tipi di risorse erano necessarie in ogni momento, in particolare per quanto riguarda le competenze tecnologiche. In questo modo Coolshop è stata in grado con grande anticipo di pianificare l'allocazione delle risorse al progetto.

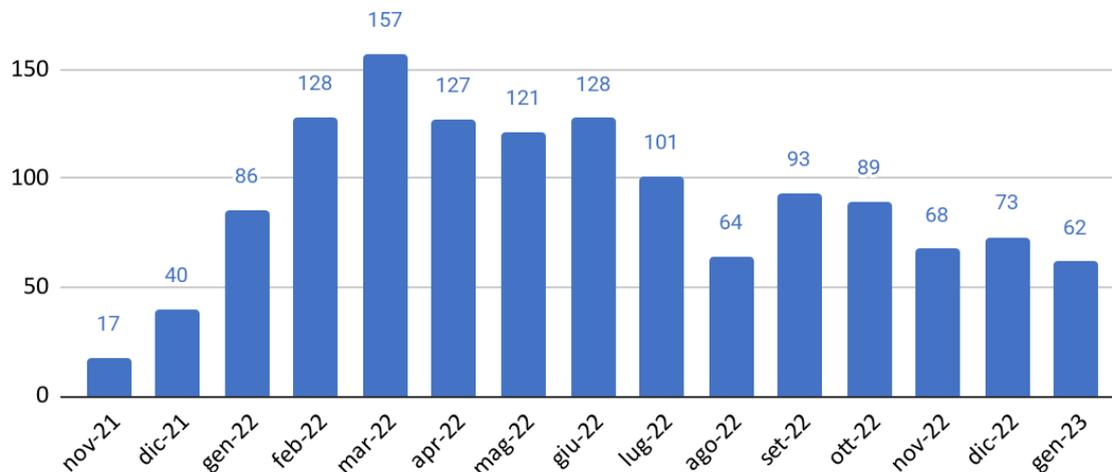


Figure 44: Man Days allocati per Mese

La rigidità della metodologia Waterfall è stata mitigata con l'assunzione che i piani iniziali sarebbero cambiati durante l'esecuzione del progetto. Questo ha permesso di prendere precauzione quali:

- la definizione del budget iniziale come una stima approssimativa e la costante informazione del cliente riguardo l'effort utilizzato, così che potesse prendere decisioni informate riguardo eventuali richieste di nuove funzionalità
- La definizione della timeline iniziale con una contingency ben precisa, che permettesse a Coolshop un'allocazione efficiente delle risorse e a CMCO di pronosticare quando sarebbe avvenuto il rilascio del prodotto completo.
- La pianificazione per release, che ha permesso di abbassare i costi di eventuali cambi di requisiti

L'utilizzo della metodologia Agile a livello operativo ha offerto notevoli vantaggi durante lo sviluppo del progetto. La suddivisione delle funzionalità in release diverse ha permesso di svolgere contemporaneamente le fasi di progettazione, implementazione e testing, permettendo un uso delle risorse più intensivo.

La costante verifica del codice durante la creazione ha permesso di identificare e risolvere rapidamente eventuali errori o problematiche, garantendo un applicativo affidabile e funzionante già alla prima release. Questo ha assicurato che le esigenze del cliente venissero soddisfatte.

Inoltre, la possibilità di avere subito software funzionante ha permesso al cliente di capire meglio quali erano le sue esigenze e richiedere modifiche che Coolshop ha potuto implementare grazie alla flessibilità concessa dalla metodologia Agile; proprio a questo fine è risultata utile la squadra del cliente richiesta da Coolshop: il continuo feedback ricevuto da personalità di reparti differenti ha permesso all'applicativo di ottenere una migliore User Experience.

Con il fine di rendere consistente l'utilizzo della metodologia Agile a livello operativo con i vincoli aziendali e imposti dalla metodologia Waterfall per la pianificazione ad alto livello, sono stati adoperati i seguenti accorgimenti:

- Processo di sviluppo ben definito: come descritto in precedenza, le risorse allocate ai progetti di Coolshop hanno un livello di expertise molto diverso tra di loro, motivo per il quale è sempre necessario che il nuovo codice introdotto nell'applicativo sia supervisionato dal Technical Leader di progetto, che ne possa confermare la qualità e la consistenza con le logiche precedentemente implementate.
- Precisa documentazione: all'inizio di ogni release sono stati redatti e condivisi i Functional Design, documenti che descrivono dettagliatamente le funzionalità che saranno implementate durante la fase di sviluppo. In tal modo, Coolshop e il cliente sono allineati su come una funzionalità sarà sviluppata, quindi eventuali futuri cambi di requisiti saranno tracciati come Change Request e sviluppati attraverso il rimanente budget di progetto.
- Tracciamento dell'effort utilizzato: ad ogni membro del team è stato richiesto di loggare, all'interno del software Jira, le ore utilizzate per il completamento di ogni task. In questo modo è possibile calcolare facilmente quale sia stato il budget effettivo utilizzato fino a quel momento.

Fix version / User	Σ	Nov 2021	Dec 2021	Jan 2022	Feb 2022	Mar 2022	Apr 2022	May 2022	Jun 2022	Jul 2022	Aug 2022	Sep 2022	Oct 2022	Nov 2022
Release 4	197h 20m			0.5	4.5	35.5	75.08	26.25	35.75	9.75		5	4	1
Alessandro	9h 30m					1	4.5		4					
Matteo	8h							1	7					
Oscar	87h 35m			0.5	2.5	6.5	29.83	13.75	24.75	9.75				
Roberto	2h				2									
Simone	7h 45m					4.5	2.75	0.5						
Jari	82h 30m					23.5	38	11				5	4	1
Release 5	187h 5m				1		1	24	59.33	42.25	30.75	24.75	4	
Federico	56h 5m							24	23.58	7.5	1			
Matteo	3h									1.5	1.5			
Oscar	19h 45m								0.5	7.25	2.25	9.75		
Simone	8h 30m								0.5	0.5	3	4	0.5	
Jari	99h 45m				1		1		34.75	25.5	23	11	3.5	
1.1	24h 30m			24.5										
Total	10946h 25m	139.17	320.25	684.17	1024.92	1258.17	1017.67	970	1021.08	806.33	511	744.42	714.08	541.92

Figure 45: Tracciamento effort Jira

5. APPROCCIO E METODO DI RICERCA

Come già definito nel sottocapitolo "Obiettivo della Tesi", il fine di questo lavoro di tesi è quello di validare la metodologia di project management utilizzata in Coolshop. Attraverso l'esplorazione dei vincoli di utilizzo, benefici e malefici, si vuole fornire un quadro completo della metodologia, cosicché possa essere utilizzata da aziende dalle caratteristiche simili in termini di dimensione, settore e complessità dei progetti.

Siccome non mi è stato possibile reperire un database contenente i risultati di progetti simili a quelli sviluppati in azienda con cui poter effettuare confronti, né ho avuto a disposizione di un campione abbastanza ampio di soggetti che lavorano nel settore a cui somministrare questionari, ho optato per un metodo di ricerca qualitativa basato su interviste semi-strutturate, che hanno permesso di tirare fuori costrutti, idee, insight e ampliare il discorso, essendo un approccio meno strutturato rispetto al questionario.

Ogni intervista è stata trascritta e analizzata attraverso la tecnica del coding, un processo di analisi dei dati che viene utilizzato per categorizzare e organizzare le informazioni raccolte dalle interviste (Corbin & Strauss, 1998).

Questo processo consiste nell'identificare i temi o le categorie emergenti dai dati raccolti e nell'assegnare codici o etichette alle informazioni per facilitare la successiva analisi e la sintesi dei risultati. Il coding delle interviste aiuta a trasformare le informazioni qualitative raccolte in dati quantitativi, rendendoli più facilmente analizzabili e comparabili (Andrea Vargiu, 2020).

Le persone intervistate sono stati membri del team di progetto, che hanno ricoperto ruoli diversi durante le varie fasi del progetto; le interviste sono state condotte da me, di persona, presso gli uffici Coolshop e hanno avuto una durata approssimativa di un'ora ciascuna.

Nei seguenti sottocapitoli saranno riportati e analizzati i temi emersi dalle interviste.

5.1. INTERVISTA CON PALADIN/PM - da discovery a Release 4

In questo sottocapitolo si analizzerà l'intervista del Paladin Senior che, come spiegato da lui durante il progetto, ha svolto questo ruolo dalla fase di discovery, fino alla fine della Release 4. Difatti, come illustrato in precedenza, durante l'esecuzione del progetto c'è stato un cambiamento della risorsa Paladin.

L'intervista può essere letta nell'allegato *Analisi_Intervista_Paladin_1*

5.1.1. Tematica 1: Passaggio di Consegna

Il passaggio di consegna tra le risorse Paladin Senior è stato valutato negativamente, a causa del contenuto del Knowledge Transfer, risultato manchevole di alcuni argomenti:

- Non tutte le informazioni sul progetto sono state discusse; in particolar modo si è fatto riferimento agli accordi presi tra le parti in momenti concitati del progetto, per i

quali mancava una corretta documentazione ed è risultato difficile tenerne traccia tra i vari argomenti da trattare.

- La concentrazione eccessiva sul prodotto durante il Knowledge Transfer ha portato ad una mancanza di attenzione alle informazioni relative al cliente, che avrebbero richiesto un trattamento più approfondito per ottenere una maggiore comprensione del progetto. Si consiglia l'utilizzo del SOW come base per l'acquisizione di queste informazioni.

Inoltre, il nuovo Paladin era una risorsa entrata di recente in Coolshop, quindi non era familiare con i processi interni del team. Questo ha portato a una mancanza di comprensione dei flussi di lavoro e delle dinamiche di comunicazione, rendendo più difficile il trasferimento di conoscenze.

D'altra parte, il vecchio Paladin non era avvezzo alle best practice dell'azienda, il che ha reso difficile la comprensione di ciò che era presente all'interno del progetto, motivo per il quale è stata necessaria la sua collaborazione anche dopo il Knowledge transfer.

Il cattivo passaggio di consegna ha complicato la gestione del progetto, rendendo il team inefficiente e comportando uno spreco di Budget.

5.1.2. Tematica 2: Budget di Progetto

Durante l'intervista sono state discusse le motivazioni per le quali sono state erogate più giornate di lavoro rispetto quelle vendute, causando un bilancio negativo per il progetto.

Come si può vedere nel grafico riportato in figura 43, verso il periodo di Agosto 2022 il bilancio del progetto ha raggiunto lo zero, per poi diminuire sempre di più. Ciò è accaduto perché, secondo i piani iniziali, lo sviluppo del prodotto doveva essere completato internamente in quel periodo.

Le motivazioni che hanno portato a tale esito sono molteplici.

In primo luogo, l'integrazione di nuovi tools sconosciuti a Coolshop ha reso molto difficile effettuare precise stime di effort necessario, che per tale ragione è successivamente risultato insufficiente.

Inoltre, le stime non hanno tenuto conto che i tools erano nuovi anche per il cliente, il che ha causato ingenti ritardi e inefficienze nella raccolta dei requisiti.

Una cattiva pianificazione iniziale delle risorse da allocare al team di progetto ha inizialmente causato una congestione del team, provocando ritardi nella consegna del progetto. Difatti, in figura 43 è possibile notare che il bilancio di progetto risultava positivo nel mese di Aprile; ciò a causa di un numero di risorse insufficiente, quando evidentemente il progetto necessitava di spendere maggior budget. Successivamente si è cercato di porre rimedio a questa situazione allocando più risorse per recuperare tempo, ma ciò ha comportato un consumo di budget inefficace.

In aggiunta, una parte del budget è stata spesa per analisi relative all'integrazione di nuove funzionalità che il cliente ha poi deciso di non acquistare, a causa dei costi eccessivi rispetto ai benefici attesi. La scelta dell'azienda di non far pagare al cliente il budget così consumato, ha evidenziato una discrepanza tra le esigenze dell'azienda e quelle del progetto, con la

conseguente spesa di budget per decisioni che erano più favorevoli all'azienda stessa. Per le stesse ragioni, è stato deciso di allocare nel team di progetto alcune risorse non necessarie che hanno comportato inefficienze, ma a cui è stata garantita una crescita personale di cui l'azienda ha beneficiato.

5.1.3. Tematica 3: Turnover dei Membri del Team

Come spiegato precedentemente, Coolshop, e quindi i suoi progetti, sono caratterizzati da un frequente turnover delle risorse.

Questo implica inefficienza per le risorse dev, che necessitano di effort per studiare il progetto, al fine di comprendere i processi interni all'applicativo, e di conseguenza in perdita di efficienza del Technical Leader, il quale deve necessariamente orientare le scelte tecniche delle risorse. Ciò è aggravato dalla frequente inesperienza dei developers, i quali hanno maggiore necessità di spiegazioni da parte del Technical Leader di progetto per quanto riguarda il funzionamento dell'applicativo.

La difficoltà di ingresso delle risorse è mitigata dalla documentazione sull'applicativo, la quale facilita l'accrescimento di conoscenze sui processi e funzionalità del prodotto.

Si ritiene che l'entrata di risorse all'interno del progetto è resa ancora più complicata dal fatto che l'applicativo è assimilabile ad un monolite, motivo per il quale per implementare una nuova funzionalità è necessario basarsi, e quindi conoscere, il funzionamento della restante struttura.

Ciò esclude la parte più nuova del prodotto, che essendo strutturata a microservizi, permette ai developers di implementare funzionalità pur non avendo una conoscenza approfondita del resto dell'applicativo

5.1.4. Tematica 4: Gestione del Team

La comunicazione del team è valutata molto positivamente.

L'utilizzo di stand up meeting ha aiutato il team a comprendere cosa stesse accadendo ogni giorno, permettendo ai developers di organizzarsi tra loro per scambiare informazioni su come portare avanti alcune evolutive, e al Project Manager di monitorare l'andamento degli sviluppi.

Durante il progetto sono state organizzate retrospettive nelle quali i membri del team potevano portare sul tavolo di discussione eventuali problematiche a cui cercare rimedio, ad esempio è stata ridotta la durata degli stand up meeting, così da renderli più efficienti; oppure proporre nuove idee che potessero migliorare il lavoro del team, ad esempio è stata introdotta la ricorrenza di una mail a fine settimana verso il team che spiegava cosa sarebbe successo la settimana successiva, con il fine di dare a tutti una migliore visione d'insieme. È stato puntualizzato dall'intervistato che le retrospettive siano uno strumento da utilizzare con parsimonia; dal momento che il team è risultato essere ingolfato, non c'era l'effort necessario per migliorare tutti gli aspetti portati sul tavolo della retrospettiva, e a lungo andare questo avrebbe potuto generare delusione all'interno del team.

Anche la comunicazione con il cliente è valutata positivamente e il team richiesto da Coolshop è risultata essere di aiuto. Sono state organizzate riunioni settimanali, con un team

più stretto, e mensili, svolte con stakeholders più larghi, ma si considera sarebbe stato efficace anche un breve meeting giornaliero per andare incontro all'inesperienza del PM del cliente, così da affrontare subito eventuali criticità e open point per entrambe le parti. Benché si reputi che le retrospettive con il cliente possano essere d'aiuto, il team non disponeva dell'effort necessario per portare eventuali miglioramenti proposti.

Si ritiene necessario migliorare l'allocazione del Paladin e Hero di progetto, in quanto ad ambedue le figure sono richieste una moltitudine di attività: al Paladin sono richiesti sia il ruolo di Analista Funzionale che di Project Manager, mentre l'Hero deve svolgere attività di Technical Leader, quindi seguire gli sviluppi, prendere decisioni tecniche, effettuare le stime, ma anche di developers senior. Ciò provoca una congestione nell'intero team, perchè può bloccare parte del team o comunque diminuire l'efficienza; ad esempio, a causa di ciò spesso è capitato che la descrizione dei task assegnati ai developers fosse poco chiara o incompleta.

5.1.5. Tematica 5: Organizzazione delle Release

La pianificazione di alto livello delle Release è stata giudicata inadeguata in quanto il tempo messo a disposizione per il Paladin e il Technical Leader per svolgere alcuni loro compiti è risultato insufficiente.

Ciò ha comportato che, per evitare di bloccare lo sviluppo del prodotto qualora i developers non avessero task assegnati, Paladin e Technical Leader hanno dovuto proseguire con l'analisi delle funzionalità, sacrificando le attività di code reviewing e testing.

5.2. INTERVISTA CON PALADIN/PM - da Release 5 a chiusura progetto

In questo sottocapitolo sarà analizzata l'intervista svolta con il Paladin subentrato a progetto in corso. Come spiegato dall'intervistato, è entrato nel team di progetto all'inizio della Release 2 come Analista Funzionale, con l'obiettivo di seguire le attività di integrazione tra l'applicativo e Salesforce, il CRM scelto dal cliente. Mentre, all'inizio della Release 5 è stato promosso a Project Manager, sostituendo la vecchia risorsa.

L'intervista può essere letta nell'allegato *Analisi_Intervista_Paladin_2*

5.2.1. Tematica 1: Gestione del Team

La comunicazione interna del team nel complesso è valutata positivamente.

Le richieste effettuate ai Developers sono risultate chiare e le priorità sono state modificate solo durante la fase di consegna, quando vi era necessità di maggiore attenzione alle richieste giornaliere del cliente, anche se questo poteva portare ad una diminuzione dell'efficienza.

C'è sicuramente da migliorare per quanto riguarda la gestione del Technical Leader, in quanto a causa del grande quantitativo di responsabilità, risulta essere il collo di bottiglia del team, per il quale sarebbe stata necessaria una figura di supporto.

Comunque, è stata valutata negativamente la soluzione di aggiungere un secondo Technical Leader al team di progetto, in quanto la figura mancava del background conoscitivo del progetto necessario per effettuare le corrette scelte tecniche.

Si ritiene sarebbe stato maggiormente utile una figura Dev con maggiore Seniority, in grado di sviluppare una funzionalità con poche indicazioni da parte del Technical Leader e rendergli più facile l'attività di code review, così da scaricarlo di impegni.

La gestione delle attività del cliente entro date prestabilite è stata complessa e non sempre rispettata, tuttavia la comunicazione con il cliente è considerata soddisfacente e ben strutturata. Durante la fase di consegna, tuttavia, la comunicazione è stata meno organizzata a causa dell'urgenza delle richieste.

5.2.2. Tematica 2: Integrazione CMS

L'attività per l'integrazione del CMS all'interno dell'applicativo è valutata negativamente.

Coolshop non aveva conoscenze pregresse sull'applicativo, quindi l'intera fase di studio del prodotto e analisi dei requisiti necessari all'integrazione sono state totalmente affidate alla risorsa. Ciò ha condotto ad una lavorazione a bassa efficienza, per cui anche se il functional design risultante è giudicato di ottimo livello, l'effort utilizzato ha decisamente superato la stima preventivata.

Anche il cliente non aveva conoscenze pregresse sull'applicativo, per il quale aveva previsto di affidarsi ad una società di consulenza, ma poi ha rescisso la collaborazione poco dopo. Questo ha provocato grandi problemi al cliente, il quale, anche dopo la consegna del progetto, non è riuscita ad ottenere i risultati sperati dalla piattaforma.

Inoltre, la mancanza di conoscenze dell'applicativo da parte del cliente ha provocato problemi nella collaborazione tra le parti. Il cliente non è stato in grado di fornire stime attendibili sulle proprie attività, il che ha comportato grandi difficoltà nella pianificazione delle fasi di sviluppo da parte di Coolshop, e ritardi nella fase di testing del cliente, portandolo ad accorgersi di errori di integrazione in netto ritardo.

La funzionalità è stata oggetto di riflessione per definire l'importanza per la metodologia di Coolshop di avere Functional Design ben definiti, così da siglare ciò che è stato concordato. Solo in questo modo è possibile definire le responsabilità tra le parti e avere una precisa distinzione tra un bug, da risolvere senza budget, e un Change Request, che consuma il budget di progetto.

5.2.3. Tematica 3: Passaggio di Consegna

Il passaggio di consegna tra il primo Paladin e il successivo è valutato negativamente.

La procedura di Knowledge Transfer delle informazioni di progetto non è standardizzata all'interno dell'azienda, quindi è stato compito delle figure Paladin organizzarsi a tal fine. Sono state organizzate delle sessioni durante le quali venivano spiegate le funzionalità dell'applicativo e, successivamente, la nuova figura ha preso in mano le comunicazioni con il cliente e con il team di progetto.

Sarebbe stato preferibile un processo di affiancamento, durante il quale le parti si sarebbero scambiate di ruolo, così da ottenere una conoscenza più approfondita dell'applicativo e della parte commerciale del progetto.

Inoltre, il passaggio di consegna è stato inficiato da una precedente documentazione manchevole:

- L'assenza o la non completezza di alcuni Functional Design non ha permesso la perfetta conoscenza di quali fossero le funzionalità presenti nell'applicativo e le loro caratteristiche.
- L'assenza di alcuni Testbook che confermassero l'accettazione delle funzionalità da parte del cliente ha ostacolato la conoscenza del loro stato di avanzamento nel progetto; motivo per il quale è stato necessario rilavorare alcune delle funzionalità in fasi inoltrate di progetto.

5.2.4. Tematica 4: Consegna del Progetto

La qualità del prodotto consegnato è valutata positivamente, dato che i bug segnalati dal cliente sono risultati essere pochi e di basso impatto, e l'applicativo risulta avere tutte le funzionalità nello scope di progetto. Inoltre, bisogna considerare che è presente un contratto di AMS che permette di mantenere alta la qualità dell'applicativo e implementare piccole funzionalità.

Ciononostante, l'applicativo è stato consegnato in ritardo rispetto a quanto pronosticato. Difatti, rispetto la stima iniziale in fase di gara il progetto è stato consegnato 5 mesi più tardi a causa dell'aggiunta della Release 5+. L'aggiunta di questa release durante l'esecuzione del progetto può essere ricondotta alle caratteristiche della metodologia Coolshop: l'integrazione della metodologia Agile per le attività di basso livello permette al cliente di raggiungere una migliore consapevolezza di ciò che è per lui necessario, cambiando le richieste iniziali e aggiungendo nuove funzionalità, ma ciò collide con la metodologia Waterfall nella fase di pianificazione, per il quale è difficile rispettare i piani iniziali e rimanere nell'intorno di budget prefissato. Inoltre, anche la Release 5+ è terminata tre settimane in ritardo, perchè il cliente ha terminato molto in ritardo i preparativi dei propri sistemi e Coolshop non è riuscita a vincolare il cliente a terminare i test dell'applicativo per tempo.

5.3. INTERVISTA CON HERO/TECHNICAL LEADER

In questo sottocapitolo sarà analizzata l'intervista con il Technical Leader del progetto, che ha svolto lo stesso ruolo nel progetto svolto precedentemente con lo stesso cliente. Per questa ragione, risulta essere l'unica persona all'interno del team di progetto a conoscere l'intero applicativo.

L'intervista può essere letta nell'allegato *Analisi_Intervista_Hero*

5.3.1. Tematica 1: Gestione del Team

La comunicazione interna del Team è risultata essere soddisfacente, in quanto l'intero team risultava a conoscenza di ciò che stava accadendo nel progetto e quali fossero le priorità. Risultano comunque da migliorare alcuni aspetti riguardo la comunicazione con il Paladin; infatti, a volte la descrizione dei task da parte del Paladin è risultata essere incompleta, a causa di una mancanza di struttura omologata per la descrizione dei task. Inoltre, la presenza di un intermediario può inevitabilmente degradare l'informazione espressa dal cliente.

Infine, il cambio del Paladin durante l'esecuzione del progetto, ha portato la figura a non avere un'overview completa sui processi dell'applicativo. Difatti, il passaggio di consegna del Paladin durante l'esecuzione del progetto è risultata essere insoddisfacente, in quanto si è ritenuto da migliorare il processo stesso; innanzitutto, una fase di affiancamento risulta essere necessaria affinché possano essere viste insieme un gran numero di casistiche. In secondo luogo, è necessaria una documentazione specifica, come un glossario in cui sono raccolti i termini specifici dell'applicativo, e un diagramma di flusso che permetta di spiegare in modo più agevole quali siano i processi del software.

Il continuo turnover delle risorse developers all'interno del team di progetto ha portato vantaggi e svantaggi.

Sicuramente avere personalità diverse all'interno del team ha portato i singoli a crescere maggiormente, condividendo visioni, logiche e skills diverse; questo ha però portato anche a delle inefficienze, dovute al fatto che solo la risorsa che ha sviluppato una certa funzionalità ne conoscerà le logiche al 100%. Il progetto deve quindi sopportare il costo del trasferimento di conoscenze, che può avvenire attraverso la compilazione di documenti tecnici, oppure con la lettura del codice scritto, attività che risulta difficile per figure con bassa esperienza. Inoltre, benché 1-to-1 e retrospective hanno permesso all'intero team di lavorare in modo più snello, creando dei flussi più efficaci, è difficile mantenere il team affiatato se le risorse al suo interno cambiano frequentemente.

L'interazione con i membri Jedi del team è stata valutata positivamente, in quanto ha portato ad ottenere un prodotto di alta qualità in termini di User Experience. Risulta però da migliorare il processo che consente di trovare un compromesso ottimale tra i requisiti tecnici proposti dal Technical Leader e quelli di User Experience dello Jedi, in modo da ridurre il numero di iterazioni richieste e velocizzare il processo decisionale.

5.3.2. Tematica 2: Ingolfamento della Risorsa

Come illustrato in precedenza, il Technical Leader risulta avere molti compiti: deve effettuare un'analisi tecnica dei task scritti dal Paladin, effettuare le scelte tecniche per lo sviluppo di funzionalità, effettuare le stime richieste dal cliente, effettuare la code review del lavoro delle risorse developers e spesso deve scrivere lui stesso codice. Questo carico di compiti lo ha reso spesso il collo di bottiglia del progetto.

Per snellire il suo carico, si è cercato di delegare alle risorse developers le scelte tecniche per le nuove funzionalità dell'applicativo, le quali non necessitano di grandi conoscenze pregresse riguardo l'intero sistema e possono essere trattate stand-alone. Questa soluzione

porta comunque delle inefficienze, in quanto si potrebbero re-implementare delle funzionalità già presenti all'interno dell'applicativo.

Inoltre, durante l'esecuzione del progetto è stato introdotto all'interno del team un altro Technical Leader, che potesse aiutare il primo nella gestione delle risorse developers e nell'effettuazione di alcune stime. Ciò ha sicuramente ridotto il suo carico di lavoro, ma ha causato anche diverse problematiche; difatti, la figura non aveva una conoscenza approfondita dell'applicativo, causando errori nell'effettuazione di alcune stime.

Infine, alcune funzionalità sono state rilasciate senza la code review del Technical Leader, ma con test più intensivi da parte del Paladin. Questo ha permesso di scaricare il lavoro del TL ottenendo un applicativo funzionante, ma ha comunque ridotto le performance di alcune funzionalità, diminuendo la qualità del prodotto.

5.3.3. Tematica 3: Budget di Progetto

Come anticipato, il bilancio economico del progetto è risultato negativo. Secondo il parere del Technical Leader, questo è stato causato da una sottostima delle funzionalità.

Le motivazioni per il quale ciò è accaduto sono varie. Innanzitutto molti strumenti che bisognava integrare non erano conosciuti dall'azienda, motivo per il quale non si avevano riferimenti a cui affidarsi.

In secondo luogo, le analisi su cui la stima doveva basarsi sono state effettuate ad un livello troppo alto, rendendo la stima più difficile da effettuare.

Non solo, l'analisi poco approfondita delle funzionalità ha costretto il team ad effettuare numerosi rework durante il progetto, poiché il cliente ha compreso quali fossero i suoi reali bisogni solo in seguito, comportando inefficienze e costi aggiuntivi.

5.4. INTERVISTA CON DEVELOPER

In questo sottocapitolo sarà analizzata l'intervista con uno dei developer del team di progetto, che, come spiega durante l'intervista, ha avuto il compito di analizzare i task scritti dal Paladin e Technical Leader e scriverne il codice per integrare la funzionalità descritta.

L'intervista può essere letta nell'allegato *Analisi_Intervista_Dev*

5.4.1. Tematica 1: Comunicazione Interna al Team

La comunicazione con il team è valutata positivamente; la risorsa è stata supportata efficacemente dal Technical Leader e dagli altri membri del team. In particolare, è risultato molto utile la cerimonia dello stand up meeting, durante il quale gli è stato possibile dichiarare su cosa stesse lavorando e gli venivano fornite indicazioni sui membri a cui chiedere per funzionalità che potevano impattare ciò che lui stava sviluppando. Inoltre, durante questa cerimonia potevano essere comunicati efficacemente eventuali cambi di priorità.

Si ritiene che sia da migliorare la struttura dei task scritta dal membro Paladin del team; difatti, spesso è mancata l'informazione del tempo stimato per lo sviluppo del task e, in particolar modo durante i periodi concitati, sono mancate alcune informazioni importanti per l'attività. Per questa ragione si consiglia la creazione di una struttura rigida per la compilazione dei task.

Le retrospective conseguite durante l'esecuzione del progetto sono state valutate utili momenti di confronto, ma è importante che siano seguiti da provvedimenti per applicare ciò che è risultante dalla riunione. Per questa ragione sono valutati più utili i 1-to-1, in quanto è più facile applicare accorgimenti nel rapporto tra singoli.

Infine, il passaggio di consegna tra Paladins è stato valutato molto positivamente, molto probabilmente perchè le comunicazioni dal Paladin verso i developers sono prima filtrate dal Technical Leader, il quale è impattato in primo luogo.

5.4.2. Tematica 2: Rapporto con Technical Leader

Il rapporto con il Technical Leader è risultato molto positivo.

Il developer ha ammesso che spesso, per portare avanti lo sviluppo di alcune funzionalità, soprattutto riguardanti la parte vecchia dell'applicativo, ha necessitato dell'aiuto del Technical Leader, in quanto solo lui disponeva delle informazioni necessarie per effettuare scelte tecniche efficaci. Questo comunque non ha bloccato le sue attività, in quanto aveva sempre un task diverso su cui poter cominciare a lavorare, in attesa che il Technical Leader si liberasse.

La risorsa è favorevole ad una maggiore distribuzione delle scelte tecniche, in quanto permetterebbero al Technical Leader di scaricare alcuni dei suoi compiti e renderebbe il lavoro dei developers più motivante, aumentando l'efficienza. In particolar modo, si ritiene che ciò possa essere fatto sulla parte nuova dell'applicativo, dove c'è un minore bisogno di conoscenze pregresse sul funzionamento del software.

Infine, il developer ha valutato positivamente le misure adottate dal team per gestire eventuali ritardi di sviluppo, dovuti a stima errate. Ha infatti notato una buona flessibilità del team nella definizione delle date di consegna delle funzionalità, permettendogli di mantenere sempre un carico di lavoro adeguato.

5.5. INTERVISTA CON JEDI/UX EXPERT

In questo sottocapitolo sarà analizzata l'intervista con lo Jedi del team di progetto, il quale si è occupato della definizione della User Experience e della successiva realizzazione della User Interface dell'applicativo.

L'intervista può essere letta nell'allegato *Analisi_Intervista_Jedi*

5.5.1. Tematica 1: Gestione del Team

Con il fine di realizzare una User Experience di alta qualità, ma contenuta nei costi di sviluppo, lo Jedi deve collaborare con l'Hero e il Paladin di progetto per raggiungere la soluzione ottimale. La cooperazione tra le tre figure è stata valutata buona, anche grazie alle retrospective svolte tra i membri, sia durante l'esecuzione del progetto, sia per la fase di gara e workshop. Queste con il tempo hanno aiutato a snellire il lavoro, riducendo il numero di volte che lo Jedi ha dovuto cambiare flussi e visuals.

La cooperazione tra le tre figure ha permesso di trovare ottime idee di system design, grazie alle quali è stato possibile implementare per funzionalità diverse, processi simili, ad esempio impaginazioni con le stesse regole, che ha apportato benefici sia dal punto di vista della UX, in quanto l'utente riconosce e si abitua allo stesso comportamento in punti diversi dell'applicativo, sia dal punto di vista tecnico, in quanto riduce notevolmente i tempi di implementazione.

Inoltre, le riunioni di coordinamento svolte tra Jedi, Paladin e Hero, hanno permesso di valutare volta per volta quale fosse lo stato del progetto, in termini di budget e tempi, andando eventualmente a modificare la User Experience per trovare soluzioni più facili da implementare, qualora il budget preventivato non fosse sufficiente.

5.5.2. Tematica 2: Allocazione sui Progetti

L'allocazione delle risorse Jedi sui progetti risulta essere molto diversa rispetto a quella delle altre figure aziendali; difatti, circa il 40% dell'effort Jedi sul progetto è stato utilizzato nei periodi di gara e workshop. Per questa ragione, le figure Jedi sono allocate su molti progetti, da qui la necessità di avere un'attenta pianificazione delle risorse sui progetti.

La strategia di allocazione delle risorse Jedi prevede una prima raccolta delle richieste Paladin per il mese successivo, per poi redigere un planning mensile delle attività. L'intervistato ha valutato molto buona la strategia, in quanto si riesce a mantenere un grado di flessibilità tale da poter soddisfare eventuali urgenze sopraggiunte.

5.5.3. Tematica 3: Fase di Gara

Per la fase di gara di questo progetto è stato utilizzato un approccio più sperimentale, affidando allo Jedi l'ideazione iniziale del prodotto, partendo dalle esigenze degli utenti e creando diversi flussi di funzionamento dell'applicativo.

I processi sono stati successivamente rivisti da Paladin e Hero, che ne hanno corretto rispettivamente gli aspetti funzionali e tecnici, in modo da avere un budget contenuto; ciò ha permesso l'ideazione di una User Experience più innovativa e meno vincolata.

La scelta di utilizzare questo approccio sperimentale è motivata dalla strategia commerciale di instaurare rapporti di lunga durata con i propri clienti; infatti, avendo già intrapreso rapporti con il cliente e conoscendo la sua attenzione per la User Experience, si è deciso di puntare maggiormente su quell'aspetto.

Ovviamente questa strategia necessita di un elevato numero di iterazioni del processo, in quanto Paladin e Hero di progetto hanno richiesto molte correzioni alla figura Jedi. Per questa ragione, invece di creare direttamente dei visuals che avrebbero rischiato di non

essere accettati, risultando in un grande spreco di effort, si è deciso di lavorare attraverso l'utilizzo di Wireframes, i quali sono meno costosi da produrre e permettono di tracciare ad alto livello il journey dell'utente sull'applicativo.

5.5.4. Tematica 4: Fase di Workshop

Durante il workshop, il membro Jedi del team di progetto ha collaborato con vari esponenti dei dipartimenti di CMCO; il suo compito è stato quello di trovare una User Experience per l'applicativo che potesse mettere d'accordo tutti gli stakeholders di progetto.

Lo Jedi ritiene che la fase di Workshop sia stata di successo, perchè ha potuto lavorare assieme ai dipendenti di CMCO, che meglio conoscono gli utenti finali dell'applicativo, per poter creare una User Experience che li potesse soddisfare.

6. RISULTATI

Come già illustrato in precedenza, le scelte della metodologia di project management dipendono dalle specifiche caratteristiche ed esigenze del progetto, dell'azienda cliente e di quella fornitrice.

Con il fine di fornire al lettore quali siano i contesti in cui può essere utile implementare la metodologia Coolshop per la gestione di un progetto, nel seguente capitolo saranno utilizzate le analisi effettuate sul caso di studio e sulle interviste ai membri del team di progetto per illustrare i contesti applicativi, i benefici e i malefici nell'utilizzo della metodologia ideata da Coolshop.

6.1. APPLICAZIONI

La metodologia sviluppata da Coolshop potrebbe prescindere il contesto della consulenza informatica, modificando ruoli ed interazioni al fine di essere adattata ad altri contesti; ad esempio, potrebbe essere utilizzata per lo sviluppo di un prodotto fisico o di un servizio in cui si ha l'esigenza di pianificare e programmare il proprio lavoro con largo anticipo, ma allo stesso tempo essere in grado di adattarsi alle esigenze e alle richieste di modifica di eventuali clienti durante lo sviluppo.

La metodologia può essere utilizzata per progetti nei quali il cliente non è conscio di tutti i possibili requisiti del prodotto finale; infatti, la metodologia permette di ottenere un Minimum Viable Product in tempi relativamente ristretti, che il cliente può utilizzare per capire quali siano le sue reali necessità, come è stato fatto nel [progetto Partner Portal](#) quando il cliente ha organizzato una fase di test con gli utenti, rendendosi conto che alcune funzionalità omesse erano importanti.

Infatti, l'intervista con il [primo](#) e il [secondo Paladin](#) ha fatto emergere come la metodologia necessita di un cliente disponibile a partecipare al progetto, che fornisca requisiti e valutazioni, testi il prodotto e svolga, se necessario, alcune attività per l'integrazione con i propri sistemi.

L'utilizzo della metodologia waterfall ad alto livello rende la metodologia Coolshop adatta per quei clienti che, per ragioni commerciali o strategiche, necessitano di conoscere una stima massima di tempi e costi del progetto.

Non solo, la pianificazione delle attività rende la metodologia adatta a progetti in cui il cliente deve svolgere attività essenziali allo sviluppo del prodotto o servizio, rendendo facile per il cliente vincolare determinate risorse in un certo periodo.

L'utilizzo della metodologia risulta efficace per aziende fornitrici che standardizzano la metodologia su un portfolio di progetti. Attraverso la pianificazione di lungo periodo delle attività per i vari progetti e l'utilizzo del file di staffing, è possibile conoscere e pianificare efficacemente l'allocazione delle risorse sui progetti.

La cosa è sottolineata durante le interviste dallo [Jedi](#), il quale afferma che la metodologia di Coolshop si caratterizza per la sua flessibilità, che consente a diverse figure professionali di lavorare su vari progetti senza essere vincolati ad una specifica allocazione. Questa

peculiarità ha permesso di superare alcune problematiche legate all'assegnazione delle risorse e di gestire i progetti in modo più fluido ed efficiente.

Come illustrato dal [secondo Paladin di progetto](#), la produzione di documentazione come Functional Design e Testbook permette alla metodologia di essere consigliata per aziende che tendono a instaurare rapporti di partnership di lunga durata, piuttosto che sviluppare progetti one-off.

Inoltre, come spiegato dal [primo Paladin intervistato](#), questa documentazione facilita il lavoro in contesti lavorativi caratterizzati da alto turn over, in quanto permette un più facile inserimento alle risorse più Junior, le quali possono farne uso per accrescere la loro conoscenza riguardo il prodotto o servizio. D'altronde, la metodologia è adatta per aziende fornitrici, e quindi per progetti, in cui vi è una grande disparità di skills tra le varie risorse developers, per le quali le risorse senior indirizzano e revisionano le varie attività.

L'utilizzo della metodologia è rischiosa quando applicata per prodotti o servizi che utilizzano tecnologie non conosciute dall'azienda fornitrice. Come emerso dalla discussione con l'[Hero](#) e il [Paladin](#) di progetto, stimare l'effort di attività mai svolte precedentemente potrebbe essere davvero complicato, compromettendo il budget di progetto. Il rischio, però, potrebbe essere mitigato attraverso l'utilizzo di accordi commerciali o approfondendo maggiormente l'analisi dei requisiti.

In conclusione, la tabella seguente mette a confronto i contesti applicativi della metodologia Coolshop con quelli delle metodologie tradizionali, Waterfall e Agile

Caratteristica	Waterfall	Agile	Coolshop
Documentazione	Dettagliata su tutti gli aspetti del progetto	Richiesto solo il necessario	Dettagliata su molti aspetti del progetto
Cliente	Coinvolto solo durante la fase iniziale	Coinvolto durante tutta la durata del progetto	Coinvolto durante tutta la durata del progetto
Membri del Team	Alta numerosità, Turnover frequenti	Bassa Numerosità, Esperti, collocati nello stesso luogo, team consolidato	Bassa Numerosità, Expertise variabile, Turnover Frequenti
Dimensioni del Progetto	grande	piccolo	Medio
Azienda	Grande	Piccola, Smart	Piccola, Smart
Membri del Team	Molti	Pochi	Pochi
Requisiti	Chiari, basso tasso di cambiamenti	Non chiari, frequenti cambiamenti	chiari ad alto livello, con cambiamenti durante l'esecuzione
Pianificazione	Lineare	Iterativa	Lineare ad alto livello, Iterativa a basso livello

Enfasi	Processo	Persone	Processo
Stile del Management	Autocratico	Distribuito	Autocratico
Misure di Performance	Conformità al piano	Valore apportato	Conformità al piano e Valore apportato

Table 2: Confronto tra metodologie tradizionali e Coolshop

6.2. VANTAGGI

Sono di seguito descritti i vantaggi apportati dall'adozione della metodologia di Coolshop.

- **Qualità del Prodotto**

L'utilizzo della metodologia permette di assicurarsi che il prodotto rispetti la qualità promessa.

Dall'intervista con il [secondo Paladin](#) è emerso che l'esistenza di fasi di testing alla fine di ogni Release, da svolgere sia internamente che lato cliente, congiuntamente a un periodo di Post Go live e alla tendenza a prolungare i rapporti attraverso un contratto di AMS, permette di accertarsi che l'applicativo sia privo di bug e si mantenga stabile.

In aggiunta, la continua integrazione del cliente come membro attivo del progetto, il rilascio di software funzionante ad ogni release, permettono al cliente di scoprire quali siano le sue reali esigenze, permettendo di sviluppare un prodotto che possa davvero avere valore per lui.

Infine, come spiegato nell'intervista con lo [Jedi](#), la metodologia permette di mantenere flessibile l'investimento sulla qualità, aumentandolo o riducendolo quando necessario.

- **Monitoraggio del Tempo di progetto**

La metodologia waterfall ad alto livello, con l'utilizzo dei Gantt Chart, permette di conoscere quale dovrebbe essere lo stato dei lavori e notare lo scostamento con ciò che è stato performato.

- **Comunicazione del Team**

Come sottolineato da tutti i partecipanti alle interviste, la metodologia permette un'ottima comunicazione interna e verso il cliente.

La cerimonia dello stand up meeting e l'utilizzo di Jira permette al team di essere coordinato e di facilitare lo scambio di informazioni.

Inoltre, l'utilizzo di retrospective e 1-to-1 permettono al team di migliorarsi e di risolvere eventuali conflitti.

- **User Experience**

Come spiegato dallo [Jedi](#), l'organizzazione di Workshop all'inizio del progetto permette di creare una User Experience di ottima qualità grazie all'intervento dei dipendenti del cliente, i quali conoscono meglio gli utenti finali.

Inoltre, il rilascio di software funzionante, pronto per essere testato dal cliente, permette al team di modificare la User Experience in base alle esigenze dell'utente finale.

- **Informazione condivisa**

Gli obiettivi del progetto sono condivisi con ogni stakeholders per tutta la durata del progetto.

Quindi, benché durante il ciclo di vita del progetto le priorità e i requisiti possono facilmente cambiare, lo scope del progetto rimane sempre ben chiaro a tutti.

6.3. SVANTAGGI

- **Necessaria molta documentazione**

A differenza di ciò che accade in Agile, la metodologia di Coolshop necessita di molta documentazione; questo non solo consuma budget di progetto, ma causa un ulteriore impegno per il team di progetto, causando problematiche nel caso in cui l'attività di documentazione non fosse svolta correttamente, come testimoniato dal [secondo Paladin](#).

I motivi per cui è necessaria molta documentazione sono vari:

- La vendita del progetto a prezzo fisso, con funzionalità prezzate sulla base di una stima, vincolano il team di progetto e il cliente a trovare un accordo che definisca in modo chiaro cosa sia la funzionalità e quando è implementata correttamente.
- Gli accordi tra le parti sono spesso riformulati e, come espresso dal [primo Paladin di progetto](#), necessitano di ulteriore documentazione.

- **Difficoltà nel produrre stime affidabili per implementazioni innovative**

L'intervista con [l'Hero di progetto](#) ha evidenziato la difficoltà nel produrre una stima affidabile dell'effort necessario per sviluppare funzionalità altamente innovative per il team.

Questo rappresenta un rischio che potrebbe impattare il budget del progetto, nel caso in cui la stima dovesse effettivamente rivelarsi errata.

- **Membri del team soggetti ad ingolfamento**

Come illustrato in precedenza, la metodologia è adatta per aziende fornitrici, e quindi per progetti in cui vi è una grande disparità di skills tra le varie risorse, motivo per il quale le risorse senior indirizzano e revisionano lo sviluppo delle attività.

Per questa ragione le risorse senior rischiano spesso di essere soggette ad ingolfamento, che genera inefficienze in tutta l'esecuzione del progetto, come illustrato dal [primo Paladin](#).

Una possibile soluzione a questa problematica è suggerita [dall'Hero](#) e consiste nell'allocare più risorse senior dall'inizio del progetto, anche se sotto utilizzate.

- **Mancanza di un processo per il Passaggio di Consegna**

La metodologia Coolshop non prevede una procedura standardizzata per il passaggio di consegna tra membri senior del team, lasciando ai soggetti la libertà di organizzare il Knowledge Transfer a loro discrezione.

Tale approccio può comportare il rischio di inefficienze all'interno del team, con conseguente perdita di budget e tempo, in quanto potrebbero perdersi informazioni chiave.

Pertanto, un'adeguata definizione di una procedura standardizzata per il passaggio di conoscenze potrebbe mitigare il rischio di causare inefficienze al team.

7. CONCLUSIONI, PROSPETTIVE FUTURE E RIFLESSIONI

Come illustrato nel primo capitolo del presente lavoro di tesi, i progetti sono uno strumento utile al fine di apportare valore alle aziende. Però, affinché un progetto possa raggiungere il successo, è importante scegliere la corretta metodologia per la sua gestione.

Nel corso del presente lavoro di tesi è stata illustrata la metodologia di gestione dei progetti ideata dall'azienda Coolshop, descrivendone il contesto in cui è nata, i ruoli e i processi di lavoro.

La metodologia risulta essere un ibrido tra le metodologie tradizionali, Waterfall e Agile, di cui è stata fatta una ricerca bibliografica al fine di comprendere appieno l'utilizzo fatto nella metodologia di Coolshop.

Con il fine di validare la metodologia, cercando in quali contesti potrebbe essere utilizzata, è stato utilizzato il progetto Partner Portal come caso di studio tipico per l'azienda.

Del progetto è stato raccontato il contesto, quindi le caratteristiche dell'azienda cliente, le varie fasi del suo ciclo di vita, i risultati ottenuti alla fine del progetto e il modo in cui la metodologia è stata implementata.

Successivamente, per ampliare il discorso e capire quali possano essere i contesti applicativi, i benefici e i malefici apportati dalla metodologia al di fuori del singolo caso di studio, è stato adottato un metodo di ricerca qualitativa basato su interviste semi-strutturate rivolte ai membri del team di progetto.

Infine, sono stati estrapolati i risultati emergenti dalle interviste, esplorando i possibili contesti applicativi e le possibili implicazioni che l'utilizzo della metodologia può portare. Nel capitolo [Risultati](#), sono stati illustrati i contesti in cui potrebbe essere applicata la metodologia Coolshop, aiutando il progetto a raggiungere il successo.

Con il fine di validare i risultati ottenuti in questa tesi, ritengo possa essere interessante implementare la metodologia per la gestione di un progetto al di fuori delle mura di Coolshop, valutandone gli impatti sul progetto e sui membri del team.

In conclusione, il presente lavoro di tesi mi ha permesso di approfondire la mia conoscenza sulle metodologie di gestione dei progetti tradizionali, scoprendone caratteristiche e strumenti, e studiare la metodologia inventata da Coolshop.

La metodologia è risultata essere adatta alla gestione dei progetti aziendali e il team di progetto si è dimostrato soddisfatto del suo utilizzo. Allo stesso tempo però, il suo studio mi ha permesso di portare alla luce alcune inefficienze, analizzate nel capitolo [Svantaggi](#), a cui cercherò di porre rimedio nell'ambiente lavorativo.

References

- Agile: la chiave per la digital transformation e la crescita aziendale.* (2020, August 25).
Mia-Platform. Retrieved December 20, 2022, from
<https://blog.mia-platform.eu/it/agile-la-chiave-per-la-digital-transformation-e-la-crescita-a-aziendale>
- Agile Manifesto: le basi da cui partire.* (n.d.). Rhubbit srl. Retrieved December 8, 2022, from
<https://www.rhubbit.it/agile-manifesto-le-basi-da-cui-partire/>
- Anderson, D. J., & Reinertsen, D. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.
- Business Agility.* (n.d.). Felice Pescatore. Retrieved December 8, 2022, from
<https://www.felicepescatore.it/business-agility>
- Corbin, J., & Strauss, A. (1998). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications.
- De Marco, A. (2018). *Project Management for Facility Constructions: A Guide for Engineers and Architects*. Springer International Publishing.
- Filippetti, A. (n.d.). *Le regole del gioco: i Ruoli Scrum*. AgilePlaza. Retrieved December 29, 2022, from <https://agileplaza.it/ruoli-scrum/>
- Filippetti, A. (2022, March 16). *Le regole del gioco: gli Artefatti Scrum*. AgilePlaza. Retrieved December 29, 2022, from <https://agileplaza.it/artefatti-scrum/>
- Filippetti, A. (2022, June 12). *Le regole del gioco: gli Eventi Scrum*. AgilePlaza. Retrieved December 29, 2022, from <https://agileplaza.it/eventi-scrum/>
- 5 Principles for an Effective Kanban System -.* (2018, August 5). OpEx Learning. Retrieved December 20, 2022, from
<https://opexlearning.com/resources/5-principles-for-an-effective-kanban-system/26661/>
- 4 Core Kanban Principles and 6 Practices (2022).* (n.d.). Teamhood. Retrieved December 21, 2022, from <https://teamhood.com/kanban-resources/kanban-principles/>

A Guide to the Project Management Body of Knowledge. (2013). Project Management Institute, Incorporated.

History: The Agile Manifesto. (n.d.). Manifesto for Agile Software Development. Retrieved December 26, 2022, from <https://agilemanifesto.org/history.html>

ICB4 - IPMA Italy. Individual competence baseline for project management (I. Ghisletti & A. Mantovani, Trans.). (2017). ANIMP Servizi.

Ilmi, M. A., Pradana, F., & Hayuhardhika Nugraha Putra, W. (2020). Software Project Management Systems Using Kanban Method in the CV. Primavisi Globalindo. *Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, 4, 215–231. <https://doi.org/10.29407/intensif.v4i2.14320>

I principi sottostanti al Manifesto Agile. (n.d.). Manifesto for Agile Software Development. Retrieved December 26, 2022, from <https://agilemanifesto.org/iso/it/principles.html>

ISO. (n.d.). 21500:2012. *Guidance on project management.*

Kisielnicki, J., & Misiak, A. M. (2017). Effectiveness of Agile Compared to Waterfall Implementation Methods in IT Projects: Analysis Based on Business Intelligence Projects. *Foundations of Management*, 9, 273–286. 10.1515/fman-2017-0021

Laoyan, S. (n.d.). *Tutto quello che c'è da sapere sulla metodologia waterfall per la gestione dei progetti* • Asana. Asana. Retrieved November 19, 2022, from <https://asana.com/it/resources/waterfall-project-management-methodology>

Lewis, H. R. (Ed.). (2021). *Ideas that Created the Future: Classic Papers of Computer Science.* MIT Press.

Metodo Kanban nel project management | Project Management Center. (n.d.). HumanWare. Retrieved December 20, 2022, from <https://www.humanwareonline.com/project-management/center/metodo-kanban-nel-project-management/>

Metodo Kanban nel project management | Project Management Center. (n.d.). HumanWare. Retrieved December 20, 2022, from

<https://www.humanwareonline.com/project-management/center/metodo-kanban-nel-proiect-management/>

Metodologia Project Management | Project Management Center. (n.d.). HumanWare.

Retrieved November 16, 2022, from

<https://www.humanwareonline.com/project-management/center/metodologia-di-proiect-management/>

Metodologie di gestione dei progetti: i dodici framework più usati • Asana. (2022, June 13).

Asana. Retrieved November 16, 2022, from

<https://asana.com/it/resources/project-management-methodologies>

Plaza, A. (n.d.). Formazione, Consulenza, Coaching Agile & DevOps - AgilePlaza Home.

Retrieved December 15, 2022, from <https://agileplaza.it>

Project Management Institute, Inc. (2017). *AGILE PRACTICE GUIDE*.

Schwaber, K. (2022, August 16). *What are Epics and Features?* Scrum.org. Retrieved

January 19, 2023, from

<https://www.scrum.org/resources/blog/what-are-epics-and-features>

Sistema Kanban: una panoramica della metodologia Kanban. (2021, September 17). Talent

Garden. Retrieved December 21, 2022, from

<https://talentgarden.org/it/project-management/kanban-system-an-overview-of-the-kanban-methodology/>

Špundak, M. (2014). Mixed agile/traditional project management methodology – reality or

illusion? 10.1016/j.sbspro.2014.03.105

Sutherland, J., & Schwaber, K. (n.d.). *The 2020 Scrum Guide TM*. Scrum Guides. Retrieved

December 12, 2022, from <https://scrumguides.org/scrum-guide.html#scrum-theory>

Sutherland, J., & Schwaber, K. (n.d.). *The 2020 Scrum Guide TM*. Scrum Guides. Retrieved

December 15, 2022, from <https://scrumguides.org/scrum-guide.html>

What is a project schedule? | AIPM. (2021, June 22). Australian Institute of Project

Management. Retrieved November 23, 2022, from

<https://aipm.com.au/blog/what-is-a-project-schedule/>

When to Use Waterfall vs. Agile. (2019, May 17). Macadamian. Retrieved December 20, 2022, from <https://www.macadamian.com/learn/when-to-use-waterfall-vs-agile/>

Andrea Vargiu. (2020, September 1). Lezione 16 La codifica delle interviste [Video]. YouTube. <https://www.youtube.com/watch?v=IDpJVmy0CgQ>