

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Gestionale – LM 31

Collegio di Ingegneria Gestionale



**Politecnico
di Torino**

Tesi di Laurea Magistrale

**Waterfall Project Management vs. Agile Project Management:
metodologie a confronto in contesto aziendale**

Relatore:

Prof. Carlo Rafele

Candidato:

Chiara Caso

Tutore Aziendale:

Giovanni Campolo

Anno Accademico 2022-2023

Sommario

INTRODUZIONE.....	1
CAPITOLO 1: Introduzione al Project Management e analisi teorica delle metodologie Waterfall e Agile.....	3
1.1. Introduzione al Project Management, al significato di progetto e alla figura del Project Manager.....	3
1.2. Introduzione al Program Management, al Project Portfolio Management e alle rispettive figure chiave.....	7
1.3. Monitoraggio e controllo.....	9
1.4. Caratteristiche della metodologia Waterfall.....	12
1.4.1. Le fasi della metodologia Waterfall.....	13
1.5. Caratteristiche della modalità Agile.....	15
1.5.1. Manifesto Agile e dodici principi.....	15
1.5.2. Scrum.....	17
1.5.2.1. Ruoli o responsabilità.....	18
1.5.2.2. Cerimonie.....	19
1.5.2.3. Artefatti.....	20
1.6. Confronto tra Waterfall e Agile.....	22
1.6.1. Il framework Cynefin e la scelta della metodologia progettuale più adatta.....	23
CAPITOLO 2: Descrizione di Reply e di come applica Waterfall e Agile.....	27
2.1. Reply S.p.A e Cluster DCX.....	27
2.1.1. Ripartizione dei Progetti nella Business Unit di Cluster DCX.....	29
2.2. Come Reply applica la metodologia Waterfall.....	32
2.2.1. Analisi.....	35
2.2.2. Design.....	36
2.2.3. Build.....	36
2.2.4. Deploy.....	37
2.3. Come Reply applica Agile.....	37
2.3.1. Offerta.....	38
2.3.2. Esecuzione.....	39
2.3.3. Template Agile utilizzati da Reply.....	42
CAPITOLO 3: Development Issue Management.....	48
3.1. Struttura del progetto DIM & Power Platform.....	48

3.1.1. Prima fase del progetto: consegna del Minimum Viable Product in modalità Waterfall	49
3.1.2. Seconda fase del progetto: miglioramenti in Agile	51
3.2. Descrizione del processo di Development Issue Management	54
3.3. Microsoft Power Platform	58
CAPITOLO 4: Analisi dei risultati	62
4.1. Analisi dei costi.....	62
4.2. Analisi dei tempi	69
4.3. Analisi della qualità	75
4.4 Analisi delle risorse.....	78
CAPITOLO 5: CONCLUSIONI ED EVOLUZIONI FUTURE.....	82
BIBLIOGRAFIA	88
SITOGRAFIA	89

INDICE DELLE FIGURE

Figura 1.1 Cambio della qualità di un progetto con impatto su tempi e costi.....	4
Figura 1.2 Portfolio, Program e Project Management	8
Figura 1.3 Curve ad S.....	11
Figura 1.4 Le sei fasi della metodologia Waterfall.....	13
Figura 1.5 Framework Scrum	17
Figura 1.6 BurnDown, BurnUp e Velocity chart.....	22
Figura 1.7 Waterfall vs Agile	22
Figura 1.8 Cynefin Framework.....	24
Figura 2.1 Ripartizione Waterfall/Agile per il Cliente 1	30
Figura 2.2 Ripartizione Waterfall/Agile per il Cliente 2	30
Figura 2.3 Approccio Waterfall	34
Figura 2.4 Agile.....	38
Figura 2.5 Template WoW	43
Figura 2.6 Template DoD	44
Figura 2.7 Template Product Backlog.....	45
Figura 2.8 Template Project Agreement.....	47
Figura 3.1 DevOps Azure.....	52
Figura 3.2 Dashboard DIM	55
Figura 3.3 Processo DIM.....	56
Figura 3.4 DIM steps	57
Figura 3.5 DIM Process Phase	57
Figura 3.6 Microsoft Power Platform	58
Figura 4.1 Conto Economico – MVP.....	63
Figura 4.2 Costi diretti - MVP.....	65
Figura 4.3 Conto Economico – Agile	66
Figura 4.4 Costi diretti - Agile	68
Figura 4.5 Gantt di progetto.....	69
Figura 4.6 Velocity chart.....	73
Figura 4.7 Burdndown chart e Burnup chart	73
Figura 4.8 Cumulative flow diagram	73

Figura 4.9 KPI - nuovi utenti per settimana	77
Figura 5.1 Solution Design - confidencial	83

INDICE DELLE TABELLE

Tabella 1 Indici di performance Earned Value Analysis	10
Tabella 2 Andamento degli Sprint	79
Tabella 3 Differenze Waterfall e Agile	87

INTRODUZIONE

Questo lavoro di tesi è stato portato avanti in contesto aziendale e, in particolare, presso l'azienda Reply. L'affiancamento al team ha permesso di analizzare un progetto effettuato per un cliente del settore automotive. Il fine ultimo di questa progettualità è quello di fornire una piattaforma Customer Relationship Management (CRM) integrata con quelle *Low Code – No Code* basate su tecnologia Microsoft. La commessa analizzata è stata divisa in due sotto commesse: la prima portata avanti secondo la modalità Waterfall e la seconda svolta in Agile. L'obiettivo di questa tesi è dunque quello di analizzare e confrontare in termini di tempi, costi e qualità i vantaggi di una modalità progettuale rispetto all'altra. La parte iniziale della tesi ha lo scopo di fornire una panoramica del mondo legato al Project Management, descrivendo in linea teorica le metodologie progettuali che sono state applicate in questo contesto; sarà, inoltre, effettuato un confronto tra le due modalità indicandone le principali differenze. Altresì fondamentale è la descrizione del contesto aziendale in cui è stato svolto questo lavoro, con un focus più dettagliato sulla modalità con cui, l'azienda Reply, applica la metodologia Waterfall e quella Agile. Si passerà poi a una descrizione dettagliata del processo e del prodotto offerto, ovvero uno strumento in grado di segnalare e gestire in maniera semplice ed efficace le anomalie che emergono in fase di sviluppo di un nuovo prodotto. Lo strumento appena descritto è denominato Development Issue Management (DIM). Saranno poi analizzati tutti gli eventi salienti che si sono verificati durante l'esecuzione del progetto, in entrambe le fasi progettuali. Infine, si effettuerà un'analisi dei risultati ottenuti in entrambe le sotto commesse. Sarà posta particolare attenzione agli eventi che si sono verificati durante l'esecuzione del progetto, specificando come questi abbiano influenzato il risultato finale sia da un punto di vista di tempistiche, che da un punto di vista economico. Lo scopo della tesi, dunque, è quello di fornire al lettore una fotografia delle due modalità progettuali, andando ad analizzare le motivazioni per cui, nell'ultimo periodo, il

trend di diffusione della metodologia Agile è crescente, a sfavore di quello della metodologia Waterfall.

CAPITOLO 1: Introduzione al Project Management e analisi teorica delle metodologie Waterfall e Agile

1.1. Introduzione al Project Management, al significato di progetto e alla figura del Project Manager

Il termine “Project Management”, o gestione dei progetti, definisce la gestione di un’impresa complessa, unica, di durata limitata, rivolta al raggiungimento degli obiettivi di progetto mediante un processo continuo di controllo e pianificazione di risorse differenziate, rispettando i vincoli di tempi, costi e qualità imposti. Bisogna specificare che “*di durata limitata*” sta ad indicare che, nel progetto, è prevista una data di inizio e una data di fine, a differenza di altre attività per cui potrebbe non essere prevista una fine; inoltre, ogni progetto è “*unico*” nei suoi elementi, nonostante potrebbero esistere progetti strutturalmente simili tra di loro. È importante, dunque, definire cosa sia un **progetto**, ovvero la risposta articolata a un bisogno o la soluzione a un problema. Le progettualità possono essere di due tipologie: interne all’azienda, come ad esempio progetti di ricerca e sviluppo di nuovi prodotti, di ingegnerizzazione di nuovi processi etc., oppure esterne all’azienda e in tal caso si può parlare di commesse per la fornitura a terzi di un prodotto o di un servizio. Difatti, questa tipologia di progettualità nasce dalla richiesta di un cliente esterno. All’inizio di un nuovo progetto devono essere definiti gli obiettivi; questa azione è molto importante poiché, la corretta definizione degli obiettivi, influenza in modo rilevante la riuscita del progetto. Ogni obiettivo deve superare il così detto test **SMART** ^[1]:

- **Specific (Specifico)**: se gli obiettivi sono troppo vasti o imprecisi, è probabile che sia più difficile raggiungerli, poiché se la definizione è troppo ampia, gli obiettivi diventano difficili da misurare. Se si è troppo vaghi, dunque, si rischia di commettere diversi errori e di incorrere in fraintendimenti.
- **Measurable (Misurabile)**: occorre poter quantificare i dati o indicare in che punto ci si trova rispetto all'obiettivo da raggiungere. Inoltre, definendo

obiettivi misurabili, ogni membro del team potrà valutare le proprie prestazioni.

- **Achievable (Raggiungibile):** è importante considerare gli obiettivi finali e pianificare obiettivi che siano raggiungibili, tenendo conto dei limiti interni ed esterni. Il raggiungimento degli obiettivi deve essere sostenibile.
- **Realistic (Realistico):** gli obiettivi devono essere, inoltre, realistici e pertinenti. Fissare obiettivi completamente inaccessibili non aumenterà l'impegno del team; allo stesso tempo però, obiettivi troppo facili non favoriranno il miglioramento dei risultati.
- **Time-bound (Definito nel tempo):** è importantissimo, infine, definire una tempistica per raggiungere gli obiettivi, in quanto un obiettivo definito nel tempo potrà prevedere un punto di inizio e uno di fine.

Dopo aver definito gli obiettivi, bisogna procedere con la stima del progetto in modo da quantificare il tempo necessario per la realizzazione dello stesso e i costi che dovranno essere sostenuti. Tutto questo perché ogni progetto, è soggetto, come detto in precedenza, a vincoli in termini di tempi, costi e qualità (o scopo), che formano il così detto **“Triangolo dei vincoli di progetto”**, in cui ogni lato rappresenta, appunto, un vincolo. Ogni vincolo è quindi correlato agli altri due, motivo per cui modificandone uno, si andrà automaticamente a modificare almeno uno dei restanti come riportato in *Figura 1.1*.

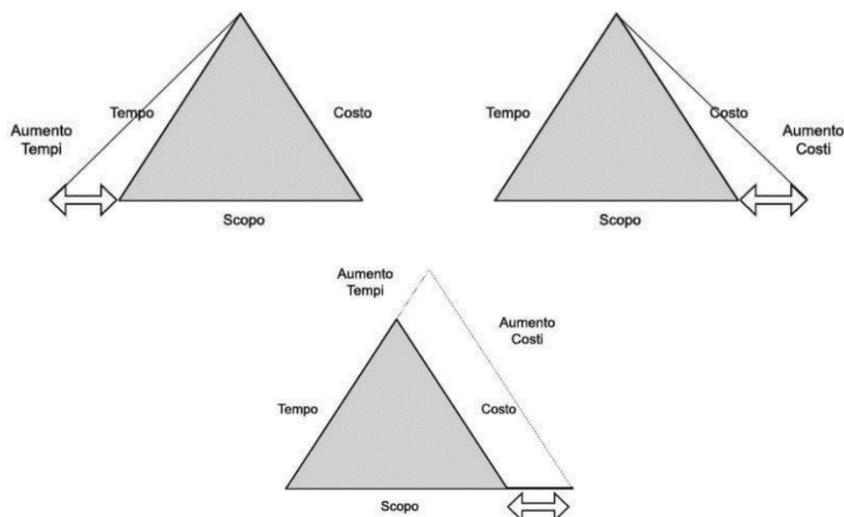


Figura 1.1 Cambio della qualità di un progetto con impatto su tempi e costi

In particolare, se si decidesse di migliorare l'aspetto della qualità, inevitabilmente ci saranno delle conseguenze su tempi e costi; ad esempio, se il vincolo dei costi è il più stringente, e quindi non può essere modificato, si andranno ad allungare i tempi. Al contrario, se il vincolo più stringente è quello del tempo, si andranno ad aumentare i costi, poiché ad esempio, si potrebbero impiegare più risorse, facendo crashing sul progetto, in modo da riuscire comunque a completarlo entro le scadenze. In generale:

- la qualità risulta essere direttamente proporzionale agli altri due vincoli;
- il tempo e il costo risultano essere inversamente proporzionali.

In un progetto, per stabilire la durata totale e i relativi costi, bisogna tener conto della pianificazione adottata, del parallelismo che si vuole o si può dare alle attività che compongono il progetto stesso e del numero di risorse adottate. I passi da seguire per effettuare una pianificazione possono essere così schematizzati:

- Suddividere tutte le attività, livello per livello, spingendosi al grado di dettaglio necessario per avere un controllo adeguato, andando a formare la così detta Work Breakdown Structure (WBS);
- Assegnare ogni attività, precedentemente individuata, alle risorse andando a formare la Resource Breakdown Structure (RBS);
- Associare ad ogni attività i costi di commessa, andando così a formare la Cost Breakdown Structure (CBS);
- Effettuare un diagramma di Gantt in cui vengono rappresentate tutte le attività con le rispettive durate;
- Analizzare i possibili rischi che potrebbero verificarsi durante l'esecuzione del progetto, in modo da mitigarne alcuni, trasferire quelli che possono essere trasferiti e, in generale, minimizzare l'impatto che i rischi potrebbero avere sul raggiungimento degli obiettivi del progetto;
- Determinare il costo totale del progetto in cui dovranno essere considerati i costi del personale, del materiale e dei servizi associati ad ogni task.

L'attività di gestione di un progetto, è in capo al **Project Manager**, ovvero colui che ha il compito di coordinare e controllare le varie fasi che compongono un progetto, con l'obiettivo di massimizzare la probabilità di successo del progetto stesso. Esso può partecipare all'esecuzione delle attività con il resto del team e ha il compito di verificare e garantire il rispetto dei vincoli di progetto (tempi, costi, qualità). Nello specifico, per quanto riguarda l'aspetto tempi, il Project Manager deve verificare la corretta schedulazione delle varie attività, nel rispetto delle tempistiche che sono state fornite dal cliente. Per quanto concerne i costi, invece, è necessario sottolineare che il Project Manager deve curare solo il Cash Flow del progetto, di cui risulta essere il responsabile. Infine, il Project Manager deve controllare che vengano rispettati gli standard qualitativi previsti dall'azienda e deve verificarne la coerenza con quanto richiesto dal cliente. In definitiva, quindi, si può affermare che il Project Manager nella quotidianità abbia come compiti principali quelli di:

- badare all'organizzazione e alla pianificazione delle varie attività;
- controllare che tutto venga eseguito secondo i piani;
- prendere decisioni inerenti quella progettualità;
- gestire i contratti con le varie funzioni aziendali in modo che queste ultime svolgano il lavoro concordato entro i limiti di tempo prefissati;
- prevedere in anticipo se ci potranno essere scostamenti significativi dagli obiettivi concordati o se questi ultimi siano difficili da raggiungere.

Ciò significa che l'intervento del Project Manager è principalmente di tipo organizzativo e non tecnico, come può essere quello delle altre figure all'interno del team. È infatti importante la capacità di coordinare il lavoro del team, in modo che l'esecuzione del progetto proceda senza rallentamenti improvvisi, nel rispetto dei vincoli del progetto stesso. Si può dire, in definitiva, che il Project Manager è tenuto a ricoprire diversi ruoli come: leader, motivatore, modello da seguire per il resto del team, negoziatore e attivatore.

1.2. Introduzione al Program Management, al Project Portfolio Management e alle rispettive figure chiave

Mentre il Project Manager ricopre tutte le funzioni elencate precedentemente, esiste un'altra figura definita **Program Manager** la quale svolge mansioni simili a quelle di un Project Manager, ma su più progetti. Difatti, qualora dovessero esservi progetti con finalità simili o in qualche modo coordinati, si parlerà di **Program Management** o, in italiano, gestione dei programmi. Si definisce **programma** un'organizzazione temporanea di attività e di più progetti i quali, se sono gestiti in modo coordinato, generano dei benefici che non potrebbero essere ottenuti se si gestissero i progetti individualmente. Esistono diversi criteri per definire un programma ad esempio: in base alla destinazione del mercato, alla dimensione media dei progetti, al tipo di prodotto, all'area geografica, al sistema di fornitura o al tipo di industria cliente. Un altro criterio di definizione è la grandezza e la portata del progetto; difatti, si potrebbero considerare tutti i progetti medio-piccoli come facenti parte di un unico programma, mentre i progetti complessi o di grandi dimensioni, come un programma separato. Infine, le somiglianze tra i progetti possono essere riscontrate anche nel tipo di sistema di consegna e di contratto: tutti i progetti consegnati con contratti turn key, ad esempio, possono far parte di un'unica famiglia che necessita degli stessi standard, processi e risorse. Un programma di successo porta vantaggi strategici e crescita organizzativa all'azienda, piuttosto che un unico risultato tangibile. In definitiva, si può affermare che la disciplina del Program Management si propone di raggiungere gli obiettivi strategici generali dell'organizzazione, supportando i Project Manager nel processo di completamento dei progetti a loro assegnati. I Program Manager, hanno anche il compito di colmare il divario tra la strategia aziendale e l'esecuzione dei progetti. A tal fine, la loro azione deve fornire ai Project Manager le risorse e gli strumenti necessari per portare a termine progetti simili all'interno di un programma [2]. Esiste, inoltre, un'altra entità al di sopra dei programmi ovvero il cosiddetto

portafoglio di progetti. Un portfolio (o portafoglio), infatti, è un insieme di programmi e/o progetti non necessariamente legati tra loro, che sono sviluppati insieme sia per garantire un uso ottimale delle risorse dell'organizzazione, sia per minimizzare il rischio del portafoglio. La disciplina che si occupa di gestire il portafoglio di progetti è nota come **Project Portfolio Management (PPM)**. Si tratta di un metodo che consente alle aziende di selezionare i progetti più importanti e di assegnare le risorse in modo strategico, per consentire il raggiungimento degli obiettivi aziendali e per massimizzare il Return On Investment (ROI), così da far aumentare il valore per gli azionisti e per il sistema di stakeholder (clienti, fornitori, committenti etc.). Il portfolio è gestito attraverso un processo di pianificazione, esecuzione, monitoraggio e controllo dei progetti, al fine di garantire che essi rimangano allineati con gli obiettivi aziendali, così che la competitività dell'azienda sul mercato possa aumentare. Un aspetto importante del PPM è la comunicazione e la collaborazione tra tutti gli attori coinvolti, come ad esempio la direzione aziendale, i responsabili dei progetti e le risorse. Inoltre, il Project Portfolio Management richiede un'adeguata gestione dei rischi e un continuo monitoraggio dei progressi per garantire che i progetti rimangano allineati. Difatti, la gestione del portafoglio progetti (PPM) può essere considerato un ponte tra la strategia aziendale e la gestione dei progetti stessa ^[3]. Come è stato detto in precedenza, dunque, si può considerare un legame "gerarchico" tra portfolio, programmi e progetti in quanto un portafoglio può essere un insieme di programmi che a sua volta risulta essere un insieme di progetti coordinati tra loro (*Figura 1.2*).

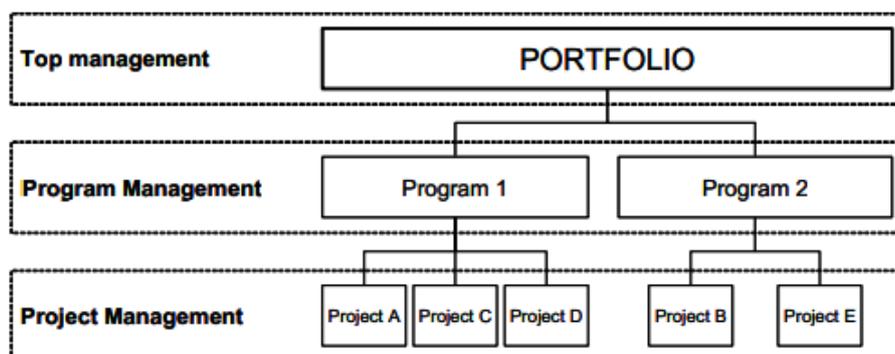


Figura 1.2 Portfolio, Program e Project Management

1.3. Monitoraggio e controllo

Un'attività molto importante legata al mondo del Project Management è quella del monitoraggio in quanto, durante l'esecuzione dei progetti, è probabile che si verifichino dei disallineamenti rispetto alle previsioni di tempo, costo e qualità che sono state effettuate nelle fasi precedenti. Per tale ragione il Project Manager ha il compito di monitorare lo stato di avanzamento del progetto, in modo da ridurre eventuali ritardi o spese extra budget. Difatti, il **monitoraggio** è un metodo di gestione che permette di verificare l'effettivo stato di avanzamento del progetto e le prestazioni in termini di tempi e costi. Oltre all'azione di monitoraggio, il Project Manager deve svolgere anche quella di controllo. Il **controllo** è un processo ricorrente che prevede la selezione di potenziali azioni correttive e preventive, ed eventualmente la riprogrammazione e la riorganizzazione del progetto attraverso la definizione dei nuovi obiettivi. Le misure dell'avanzamento di un progetto si basano sul consumo effettivo delle risorse a una certa data, ovvero il tempo impiegato rispetto a quello preventivato e il costo sostenuto fino a quel momento, definito anche **Actual Cost (AC)**. Il progetto deve essere monitorato dalla fase successiva a quella di pianificazione per tutta la durata del progetto stesso. Il primo obiettivo del monitoraggio è quello di tracciare in tempo reale i progressi del progetto e questo è reso possibile grazie alla matrice WBS/CBS precedentemente realizzata. Difatti, l'avanzamento dell'attività è definito come il rapporto percentuale tra la quantità eseguita e quella totale programmata. Per effettuare un'analisi accurata e veritiera, però, è necessario tenere in considerazione lo stato di avanzamento del progetto. Per tale motivo, si è sviluppata la cosiddetta **Earned Value Analysis (EVA)**. Si tratta, infatti, di un metodo utilizzato per misurare le prestazioni di un progetto che integra i costi, le scadenze e il lavoro svolto attribuendo a ognuno un valore monetario. Essa si basa su 3 valori chiave:

- **Budget Cost of Work Schedule (BCWS)**, ovvero il costo pianificato del lavoro previsto in un determinato periodo di tempo;

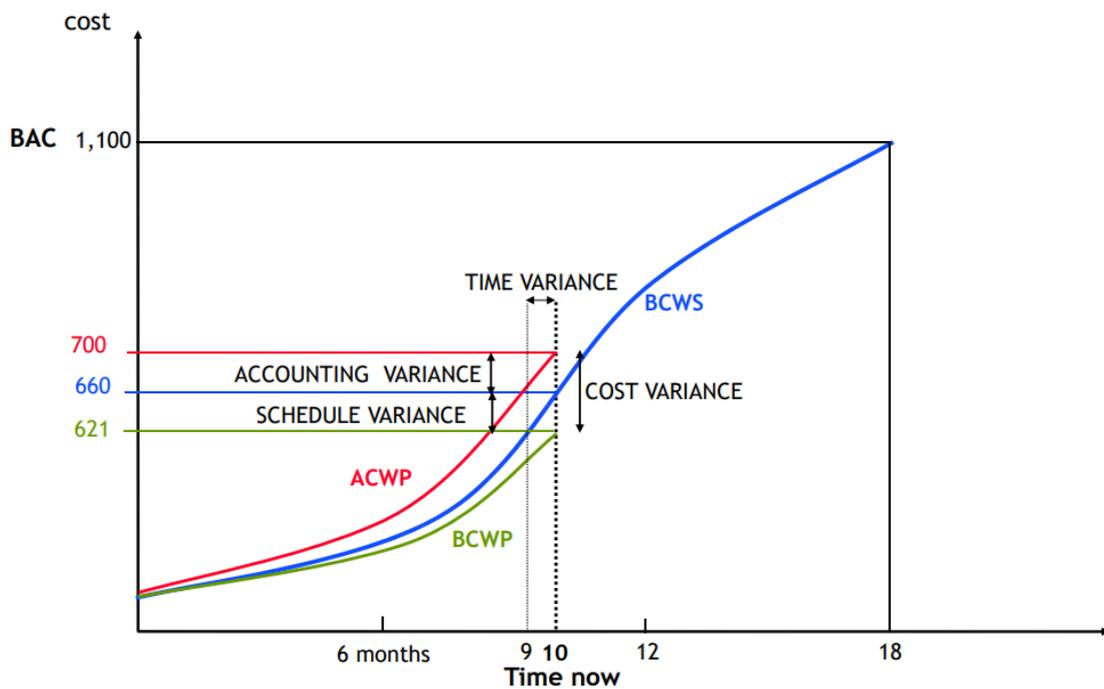
- **Actual Cost of Work Performed (ACWP)**, ovvero il costo effettivamente sostenuto per realizzare il lavoro fino a quel momento;
- **Budgeted Cost of Work Performed (BCWP)**, ovvero il valore di budget del lavoro svolto fino a quel momento.

L'Earned Value Analysis, dunque, permette di rappresentare gli scostamenti rispetto alle previsioni sia in termini di ritardo/anticipo sullo stato di avanzamento del progetto, sia in termini di costi unitari del prodotto maggiori/minori. Questa tipologia di analisi degli scostamenti permette un approfondimento accurato delle azioni preventive o correttive che possono essere attuate. Nell'EVA è possibile utilizzare una serie di indici di performance per valutare le prestazioni del progetto sia per quanto concerne i costi che per quanto concerne i tempi. In particolare, in riferimento ai costi è possibile trovare sia il **Cost Variance (CV)** = $BCWP - ACWP$, che il **Cost Index (CI)** = $\frac{BCWP}{ACWP}$. Questi servono a confrontare il costo del lavoro preventivato con quello effettivamente sostenuto. Nello specifico un valore di $CI < 1$, indica che il progetto è in *overrun*, ovvero c'è una perdita di valore; un $CI = 0$ indica che il progetto è in *linea con il budget*, e infine, un $CI > 1$ indica che il progetto è in *underrun*, con un guadagno di valore. Lo stesso discorso può essere effettuato per i tempi. In particolare, il primo indice è definito **Schedule Variance (SV)** = $BCWP - BCWS$, e l'altro è definito **Schedule Index (SI)** = $\frac{BCWP}{BCWS}$. In questo caso, un valore di $SI < 1$ indica un ritardo, un valore di $SI = 0$ indica che si è in linea con i tempi, mentre un valore di $SI > 1$ indica un anticipo.

	CI < 1	CI = 0	CI > 1
SI < 1	Progetto in ritardo e in perdita di valore	Progetto in ritardo ma in linea con i costi	Progetto in ritardo ma in risparmio
SI = 0	Progetto in linea con i tempi ma in perdita di valore	Progetto in linea con tempi e costi	Progetto in linea con i tempi e in risparmio
SI > 1	Progetto in anticipo ma in perdita di valore	Progetto in anticipo e in linea con i costi	Progetto in anticipo e in risparmio

Tabella 1 Indici di performance Earned Value Analysis

Inoltre, gli scostamenti di risorse, costi e tempi possono essere rappresentati in un grafico tempi/costi utilizzando le cosiddette curve a S. Nel grafico sono rappresentate le curve relative a BCWS, ACWP e BCWP. Ciò consente una rapida comprensione grafica dello stato del progetto che permette di suggerire azioni correttive alla strategia del progetto (Figura 1.3).



BAC = Budget at Completion (costo a budget complessivo)

Figura 1.3 Curve ad S

Dopo aver analizzato la situazione del progetto fino a un determinato momento, è possibile anche calcolare la stima a finire sia dei tempi che dei costi di progetto. Esistono due approcci per calcolare il **Cost Estimate At Completion (CEAC)**: uno più ottimistico, in quanto presuppone che dal momento in cui viene calcolato, fino alla fine del progetto non ci saranno sfioramenti di costi. Difatti, con questo primo approccio il costo totale è dato dal costo sostenuto e dalle stime a completamento, ovvero: $CEAC = ACWP + (BAC - BCWP) = BAC - (BCWP - ACWP) = BAC - CV$. Il secondo approccio, invece, presuppone che nel resto del progetto si verifichino

almeno le prestazioni avvenute fino a quel momento se non vengono intraprese le azioni correttive. In questo caso $CEAC = ACWP + \frac{BAC-BCWP}{CI} = \frac{BAC}{CI}$.

Quest'ultimo approccio può essere utilizzato anche per quanto riguarda la stima dei tempi. Pertanto, il **Time Estimated at Completion (TEAC)** sarà espresso come durata totale del progetto diviso lo Schedule Index, ovvero $TEAC = \frac{Budget\ Duration}{Schedule\ Index}$.

Riportando sul grafico delle curve ad S queste stime, sarà possibile individuare questi scostamenti anche graficamente.

1.4. Caratteristiche della metodologia Waterfall

Tra le metodologie più diffuse legate al mondo del project management, c'è sicuramente la Waterfall, anche conosciuta come metodologia tradizionale, la quale si basa su un approccio logico – temporale in cui le attività sono sequenziali. Difatti, vi è una successione di fasi del progetto e, generalmente, ogni fase termina prima che inizi la successiva. La consegna del prodotto finito al cliente è prevista soltanto nel momento in cui saranno concluse tutte le attività di progetto. Questo modello si è sviluppato nel mondo dell'industria manifatturiera ed edile, in cui vi era la necessità di terminare una fase prima di iniziare la successiva. Nel 1970 questa metodologia è stata formalizzata dal Dr. Winston Royce con il suo noto articolo *“Managing the Development of Large Software Systems”*, il quale ha condiviso le sue opinioni inerenti al modello Waterfall ^[4]. Tale metodologia, legata al mondo del project management, è definita **Project-based**: questo vuol dire che si pone molta attenzione alla fase di definizione del progetto e di tutti i suoi aspetti, come ad esempio i requisiti e le attività che lo compongono. Lo svolgimento dei progetti seguendo la modalità Waterfall, prevede la pianificazione di ogni fase nei minimi particolari, dall'inizio alla fine del progetto, in modo da fornire stabilità e struttura all'esecuzione del progetto stesso, e in modo da rendere prevedibili le risorse che dovranno essere adottate. Inoltre, come è stato già detto in precedenza, in progetti di questo tipo si passa da una fase all'altra in maniera discendente, non permettendo

quindi alcuna sovrapposizione tra le diverse fasi. Per tale motivo, è importante che i requisiti siano chiari sin dalle prime fasi, in quanto non saranno prese in considerazione modifiche ai requisiti durante lo svolgimento del progetto. La modalità Waterfall attua una strategia di tipo **predittivo** che utilizza fasi specifiche per controllare il processo. Risulta essere rigida ai cambiamenti ma, un approccio di questo tipo, ha maggiore probabilità di generare il risultato previsto. In definitiva, la Waterfall si adatta molto bene a progettualità che hanno un obiettivo ben chiaro e definito, in modo da rendere più lineare il processo di pianificazione.

1.4.1. Le fasi della metodologia Waterfall

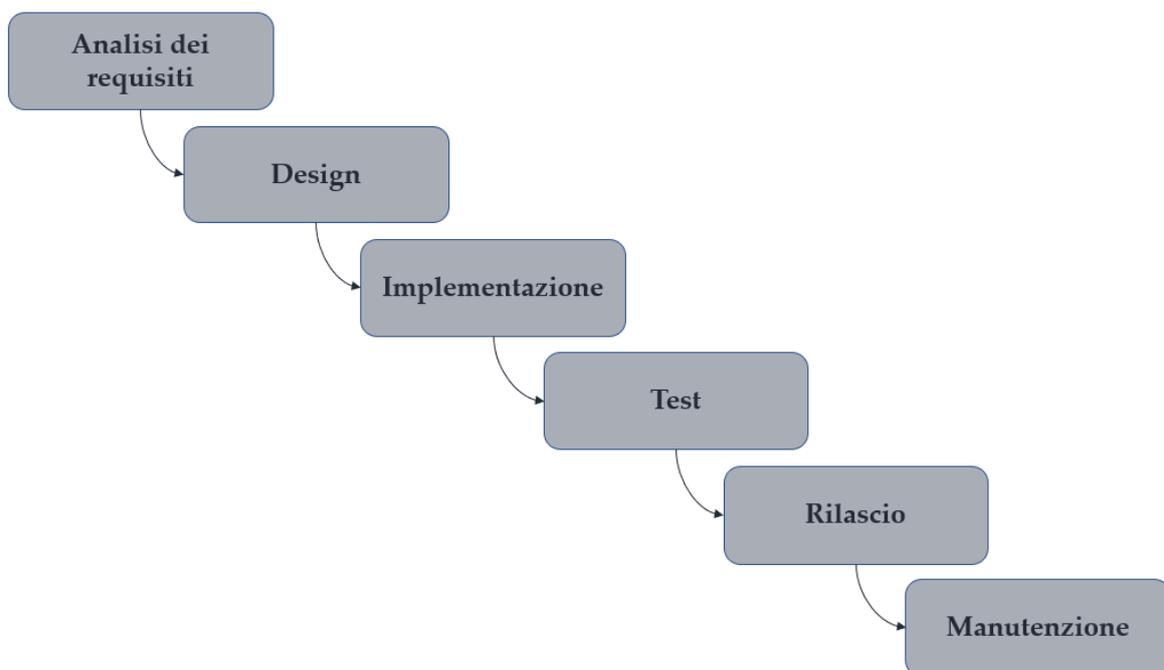


Figura 1.4 Le sei fasi della metodologia Waterfall

Il workflow di un progetto svolto in modalità Waterfall, segue sei step da eseguire in maniera sequenziale (Figura 1.4):

- Analisi dei requisiti: questa è la fase sicuramente più delicata in quanto vanno analizzati e raccolti i requisiti che poi serviranno a determinare gli obiettivi del progetto, i quali difficilmente potranno essere modificati durante lo

svolgimento del progetto stesso. Questa fase è composta da due attività: la stesura dei requisiti, in modo da determinare esattamente cosa farà il sistema e lo studio di fattibilità. Si prevede, dunque, di effettuare la stesura dell'analisi funzionale, che ha il fine ultimo di delineare i risultati del progetto, in modo da limitare le possibili conflittualità tra le parti. Per tale motivo, è importante che in questa fase partecipino sia il cliente sia l'organizzazione che dovrà gestire il progetto. Alla fine di questo step, i requisiti del progetto dovrebbero essere chiari e specificati nel documento dei requisiti che verrà condiviso con tutto il team.

- Design: in questa fase, a partire dall'output della fase di analisi dei requisiti, ovvero l'analisi funzionale, si definisce l'architettura del sistema che dovrà essere implementato. Sarà necessario ricorrere a un livello di dettaglio maggiore rispetto a quello utilizzato nella fase precedente per definire l'architettura di alto livello poiché, in questa fase, è definita l'architettura dei singoli moduli e viene descritto il progetto in tutte le sue operazioni e le relative relazioni.
- Implementazione: il team procede con lo sviluppo di quanto è stato definito nei documenti ottenuti nelle fasi precedenti, implementando tutti i moduli che sono stati individuati e concordati. Questa fase risulterà conclusa non appena tutti i moduli saranno stati implementati.
- Test: dopo la fase di implementazione, il team deve verificare che tutti i moduli funzionino correttamente, che siano allineati ai requisiti analizzati e che, una volta uniti, continuino a funzionare.
- Rilascio: il prodotto finale viene rilasciato al cliente. Questa fase risulta essere molto delicata, in quanto potrebbero emergere delle problematiche con il cliente che non sono emerse in fase di analisi dei requisiti, creando quindi delle incomprensioni sulla definizione del prodotto finito.

- Manutenzione: in questa fase tendenzialmente si svolgono dei piccoli miglioramenti al sistema, oppure potrebbe verificarsi la necessità di effettuare interventi di manutenzione, prevedendo quindi l'intervento del team.

1.5. Caratteristiche della modalità Agile

Un'altra delle metodologie legate all'ambito del project management è l'Agile, un approccio che si concentra sulla capacità di un'organizzazione di rispondere ed adattarsi in maniera rapida ai cambiamenti degli obiettivi. Questo modello si è sviluppato principalmente per far fronte alle esigenze dei progetti legati allo sviluppo del software, ed è stato successivamente sperimentato in ambito di progetti dinamici e rischiosi, in cui la metodologia tradizionale non riusciva ad essere completamente efficace. Il metodo Agile si basa su cicli di iterazione, chiamati Sprint, che hanno una durata variabile da una a più settimane, i quali iniziano con una sessione di pianificazione e terminano con la presentazione del lavoro svolto. Le consegne frequenti hanno l'obiettivo di raccogliere feedback dalle parti interessate e di dare la possibilità al team di allinearsi ai nuovi requisiti richiesti dal cliente. Il termine Agile è stato utilizzato per la prima volta nel 2001, quando è stato ideato il "*Manifesto Agile*" [5].

1.5.1. Manifesto Agile e dodici principi

- Individui e interazioni più che processi e strumenti: ovvero l'importanza di affidarsi alle persone e alla loro capacità di adattarsi e risolvere i problemi, piuttosto che concentrarsi su processi e strumenti standardizzati. Il lavoro di squadra, la comunicazione e la collaborazione sono alla base della gestione Agile.

- Risultati utili del progetto rispetto a una documentazione completa: ciò significa che viene prodotta solo la documentazione necessaria, concentrandosi sulla consegna tempestiva e frequente del lavoro svolto.
- Collaborazione con il cliente piuttosto che negoziazione del contratto: bisogna considerare gli stakeholder aziendali come parte del team e invitarli a partecipare attivamente e proattivamente, comprendendo ciò che ritengono importante e ricevendo feedback costanti sul lavoro svolto e sugli obiettivi raggiunti, è più importante della negoziazione del contratto per il successo del progetto. Il coinvolgimento degli stakeholder aziendali nel corso del progetto, offre l'opportunità di aumentare la soddisfazione del cliente e del valore aziendale.
- Rispondere ai cambiamenti piuttosto che seguire un piano: nonostante l'analisi dettagliata e la pianificazione iniziale, il risultato del progetto può evolversi nel tempo, per cui è opportuno apportare modifiche controllate e organizzate per allinearsi ai nuovi requisiti.

Dietro il Manifesto Agile vi sono 12 principi che possono essere sintetizzati come segue:

1. La soddisfazione del cliente è la massima priorità del progetto.
2. Bisogna adattarsi ai requisiti in continua evoluzione, anche a stadi avanzati del ciclo di sviluppo, a favore del vantaggio competitivo del cliente.
3. Consegnare frequentemente un risultato di valore con una preferenza per i tempi più brevi, avendo l'opportunità di raccogliere feedback e di allinearsi alla visione degli stakeholder.
4. Gli stakeholder aziendali e il team devono collaborare frequentemente.
5. Costruire i progetti attorno a persone motivate, fornire loro l'ambiente e il supporto di cui hanno bisogno e fidarsi di loro per portare a termine il lavoro.

6. Il metodo più efficiente ed efficace per trasmettere informazioni a un team e all'interno di esso è la conversazione faccia a faccia.
7. Il risultato del progetto è la misura principale del progresso.
8. I processi agili promuovono lo sviluppo sostenibile consentendo, agli sviluppatori e agli attori coinvolti, di mantenere un ritmo costante indefinitamente.
9. L'attenzione continua all'eccellenza tecnica e alla buona progettazione amplifica l'agilità.
10. Semplicità: l'arte di massimizzare la quantità di lavoro non svolto è essenziale.
11. Le architetture, i requisiti e i progetti migliori emergono da team auto-organizzati.
12. A intervalli regolari, il team riflette su come diventare più efficace, quindi regola e adatta il proprio comportamento di conseguenza.

Esistono diversi framework per applicare l'Agile come ad esempio Kanban, Adaptive software development, Extreme Programming e Scrum, il quale risulta essere quello più diffuso.

1.5.2. Scrum

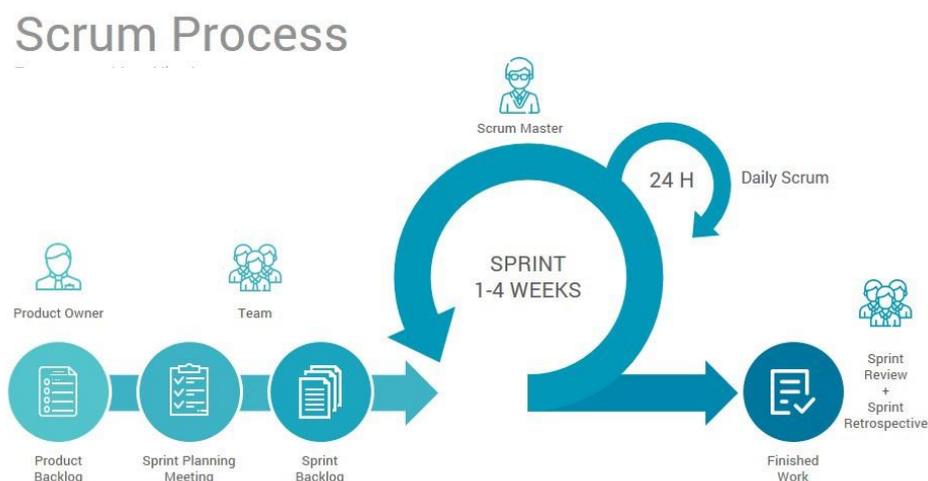


Figura 1.5 Framework Scrum

Il framework Scrum permette la gestione incrementale e iterativa di un progetto (Figura 1.5). Si basa sull'empirismo il quale, come pilastri fondamentali, riconosce la **trasparenza, l'ispezione e l'adattamento**. Questo framework è composto da diverse sottocomponenti, quali ruoli, artefatti e cerimonie, che saranno analizzati nei successivi paragrafi.

1.5.2.1. Ruoli o responsabilità

Le figure principali oltre allo Scrum Team sono lo Scrum Master e il Product Owner ^[6]; andando più nello specifico, ogni attore ha un determinato ruolo/responsabilità. In primo luogo, vi è il **Product Owner**, il quale rappresenta gli interessi del committente e ha come obiettivo quello di massimizzare il Return On Investment (ROI) per quanto riguarda progetti esterni, e quello di massimizzare il profitto per quanto riguarda progetti interni. Esso ha anche il compito di creare il Product Backlog, ovvero la lista di tutte le attività che devono essere svolte per il progetto, assegnando ad ogni task un valore di priorità, che tiene conto del valore di business generato dall'attività stessa, delle strategie applicate e di altri fattori. Un'altra figura rilevante, è quella dello **Scrum Master**, il quale ha il compito di risolvere gli impedimenti che si verificano durante lo svolgimento delle attività. Esso può essere paragonato a un coach che motiva il team, detta il passo di marcia e fa sì che siano evitate sotto e sovra allocazioni. Infine, vi è lo **Scrum Team**, composto dai così detti **Developers** (o sviluppatori), ovvero coloro che svolgono tutte le task presenti nel Product Backlog, e quindi realizzano il prodotto richiesto dal Product Owner. Il team è organizzato in modo da massimizzare la produttività, la flessibilità e la creatività. Un fattore molto importante in Scrum è che il team sia *autogestito*. Difatti, non vi è la necessità di rivolgersi a figure esterne per quanto concerne la gestione del progetto, in quanto, il team al proprio interno, ha tutte le capacità e competenze per decidere quali strategie bisogna attuare al fine di completare il lavoro. Proprio per tale ragione, gli sviluppatori hanno capacità su un

ampio numero di materie come: testing, front end, back end e così via. Tendenzialmente il team non deve superare le 10 persone, in quanto si è visto come team più piccoli risultano essere più produttivi.

1.5.2.2. Cerimonie

Come detto in precedenza, il lavoro in Scrum è organizzato in Sprint di durata variabile. Per ogni Sprint sono previste 4 cerimonie, nello specifico:

- Sprint Planning ^[7]: si svolge all'inizio di ogni iterazione e ha come fine ultimo quello di pianificare il lavoro che deve essere svolto entro il termine dello Sprint. La durata del meeting è variabile, ed è subordinata alla durata dello Sprint. Generalmente, per Sprint di 2 settimane, la durata varia dalle 2 alle 4 ore. In fase di pianificazione, può essere determinato l'effort necessario, in termini di ore, per portare avanti i singoli task. Durante questa fase bisogna chiedersi perché lo Sprint è di valore, cosa può essere completato e come si svolgeranno le attività che sono state selezionate.
- Daily Scrum ^[8]: si tratta di una pianificazione giornaliera di quello che deve essere svolto per raggiungere lo Sprint Goal. Questo meeting ha una durata di circa 15 minuti. È obbligatoria la presenza del Team, mentre risulta essere facoltativa quella dello Scrum Master e del Product Owner. Durante la riunione, bisogna aggiornarsi su quanto è stato svolto nella giornata precedente, decidere cosa si svolgerà nell'arco della giornata e individuare quali sono gli eventuali impedimenti che rallentano il lavoro.
- Sprint Review: il fine di questo meeting è quello di analizzare il risultato dello Sprint in modo da individuare eventuali migliorie negli Sprint successivi. Sono tenuti a partecipare sia gli Stakeholder che il Team Scrum ed è tenuta secondo toni informali. Per uno Sprint di 2 settimane, questa riunione, può durare fino a 2 ore. È importante che non sia questa la sede in cui il Product Owner vede per la prima volta le User Story che sono state svolte. Durante

questo meeting può essere analizzato anche il lavoro non svolto o quello che non si è riuscito a portare a termine per lo Sprint, in modo che si possano ricevere feedback e possa essere incluso nei successivi Sprint.

- Sprint Retrospective ^[9]: si svolge alla fine di ogni Sprint e ha come fine ultimo quello di raccogliere i feedback e migliorare l'esecuzione degli Sprint successivi. A questo meeting partecipa tutto lo Scrum Team, e ha una durata di circa un'ora e mezza per uno sprint di 2 settimane. Durante questa riunione tutti i membri del team devono porsi tre domande, ovvero: cosa è andato bene nello Sprint, cosa può essere migliorato e cosa il team si impegna a migliorare nello Sprint successivo. Una volta che ogni membro ha esposto le risposte alle prime due domande si passerà a una fase di votazione al termine della quale, le criticità con il maggior numero di voti, saranno gli Action Point ovvero ciò che il team si impegna a migliorare nello Sprint successivo.

1.5.2.3. Artefatti

In Scrum non si producono documenti, bensì artefatti che hanno il fine ultimo di misurare i progressi del team e di massimizzare la trasparenza. Gli artefatti previsti sono: il Product Backlog, lo Sprint Backlog e l'Increment. Il **Product Backlog** è l'insieme di tutte le attività necessarie da svolgere per raggiungere l'obiettivo del progetto. Queste attività sono chiamate *Product Backlog Item* e, ad ognuna di esse, il Product Owner assegna la priorità con cui devono essere svolte, tenendo conto di diversi fattori come ad esempio il valore economico che esse generano, il rispettivo rischio etc. Gli Item del Product Backlog sono scritti sotto forma di User Story, ovvero una descrizione informale di ciò che il team dovrà sviluppare. Il responsabile del Product Backlog è il Product Owner, che risulta essere l'unico attore in grado di modificare la priorità degli Items. Bisogna sottolineare che il Product Backlog, può essere modificato durante tutto lo svolgimento del progetto, ma è importante che ogni modifica sia comunicata allo

Scrum Team. Un altro artefatto è lo **Sprint Backlog**, ovvero un sotto insieme di Items che devono essere svolti in un determinato Sprint. In questo caso il responsabile sarà lo Scrum Team, che potrà decidere quali saranno gli Items su cui si andrà a lavorare, tenendo conto della velocità media dello Sprint precedente e provando a restare in linea con quel valore. Infine vi è l'**Increment**, ovvero cosa è stato svolto durante uno Sprint. Tendenzialmente quest'ultimo dovrebbe coincidere con lo Sprint Backlog ma, qualora il team dovesse incontrare impedimenti durante l'esecuzione dello Sprint, i due artefatti risulteranno diversi. In ogni caso l'incremento dovrà fornire un prodotto utilizzabile. Oltre agli artefatti sopracitati, esistono dei grafici che permettono di monitorare in maniera semplice e immediata l'andamento dello sprint. Essi sono di tre tipologie principali, ovvero: BurnUp Chart, Burn Down Chart e Velocity Chart. Nello specifico, il **BurnUp Chart** permette di visualizzare il lavoro che è stato svolto rispetto al lavoro totale. In questo tipo di grafico è rappresentato sia il lavoro totale che deve essere svolto per raggiungere l'obiettivo di progetto (Total effort), sia la previsione di quando questo lavoro dovrebbe essere eseguito (Ideal BurnUp) sia il lavoro effettivamente svolto (Team Effort). Sull'asse delle ascisse sono presenti le iterazioni, o sprint, mentre sull'asse delle ordinate è possibile trovare i punteggi assegnati alle User story. Il **Burn Down Chart** è complementare a quello appena descritto, in quanto indica il lavoro che manca per raggiungere l'obiettivo dello Sprint. È strutturalmente uguale al BurnUp Chart ma, mentre dal primo ci si aspetta un andamento crescente, quest'ultimo avrà un andamento decrescente. In questo caso è rappresentata sia la previsione del lavoro che deve essere svolto (Ideal Burn Down), sia il lavoro rimanente (Remaining Effort). Il **velocity chart**, infine, indica la velocità con cui vengono completate le storie; difatti ha lo scopo di contare quanti story points si riescono a sviluppare in un determinato sprint. Nella *Figura 1.6* sono rappresentati degli esempi dei grafici appena descritti.



Figura 1.6 BurnDown, BurnUp e Velocity chart

1.6. Confronto tra Waterfall e Agile

Tra le due modalità progettuali analizzate, è possibile trovare numerose differenze. La principale può essere riscontrata nel fatto che il modello Waterfall risulta essere **predittivo**; difatti in primis bisogna fissare l'obiettivo di progetto o lo scopo da raggiungere. Successivamente, backward, si definisce il piano di lavoro che permette di designare le stime inerenti a costi e tempo come illustrato in *Figura 1.7*.



Figura 1.7 Waterfall vs Agile

Il modello Agile al contrario, è definito **adattivo**, ovvero, dopo aver stabilito tempi e costi, si cerca di massimizzare il numero di requisiti che possono essere sviluppati in quel determinato Sprint. Un'altra differenza molto importante è data dal fatto

che, come è stato riportato nei paragrafi precedenti, la modalità Waterfall risulta essere molto rigida ai cambiamenti per via della sua struttura lineare e ben organizzata che vede il susseguirsi delle diverse fasi; la metodologia Agile, invece, è molto conosciuta per la sua flessibilità e adattabilità ai cambiamenti. Questo è reso possibile soprattutto dalla sua struttura in Sprint di breve durata. Difatti, a differenza della Waterfall, si tratta di un framework iterativo in cui gli Sprint sono di breve durata e possono essere definiti nuovi obiettivi e nuove strategie all'inizio di ogni Sprint, mentre nella modalità tradizionale, dopo l'elaborazione del business plan non è prevista la possibilità di cambiare obiettivi e scadenze. Per tale motivo la metodologia classica si applica molto bene a progetti ben inquadrati, con obiettivi fissi. Un'altra differenza può essere riscontrata nel fatto che la modalità Agile presenta una struttura ciclica, ovvero tutte le fasi si ripetono per ogni Sprint, mentre in Waterfall ogni fase è eseguita una ed una sola volta. Di conseguenza anche la fase di test, per quanto concerne i progetti svolti in Agile è ripetuta alla fine degli sviluppi di ogni Sprint, mentre secondo la metodologia tradizionale deve essere eseguita solo alla fine del progetto; questa può essere identificata come una delle cause principali per cui si verificano ritardi in quanto, non testando mai il prodotto, si arriva alla fine del progetto senza avere alcun tipo di riscontro sull'effettivo funzionamento di quanto è stato svolto. Infine, un'altra differenza sostanziale risiede nella composizione del team; difatti, per le progettualità svolte seguendo la metodologia Waterfall, sono preferibili team di grandi dimensioni che coinvolgono diverse funzioni aziendali. Secondo i principi dell'Agile, invece, sono preferibili team di piccole dimensioni, specializzati, in modo che il coordinamento risulti più facile ed efficace.

1.6.1. Il framework Cynefin e la scelta della metodologia progettuale più adatta

Al fine di comprendere meglio quale metodologia progettuale sia più adatta a una determinata progettualità, Dave Snowden nei primi anni 2000, ha ideato una

matrice, nota anche come CYNEFIN Framework ^[10], la quale ha il fine ultimo di aiutare i Project Manager a categorizzare i progetti. Snowden, con questo framework, ha teorizzato la presenza di cinque diversi ambienti: semplice, complicato, complesso, caotico e disordine come è possibile vedere in *Figura 1.8*.

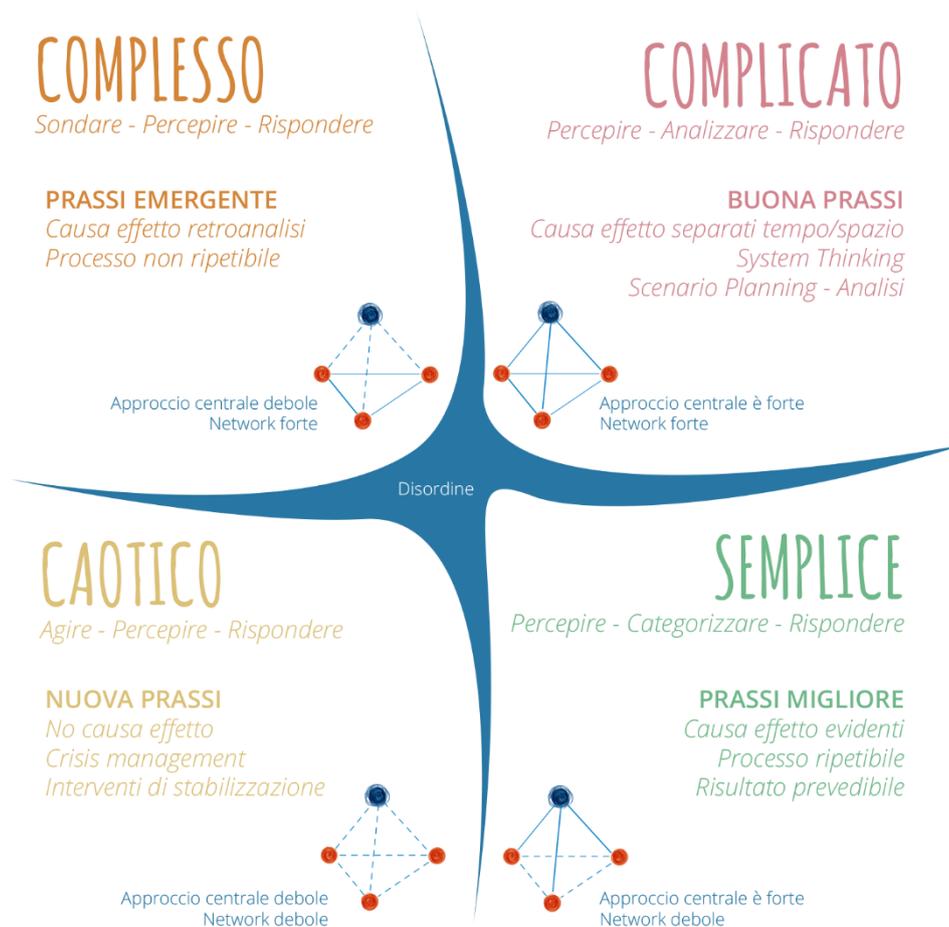


Figura 1.8 Cynefin Framework

- **Semplice:** in questo dominio è riportato tutto ciò che si conosce, per cui vi sono delle relazioni causa – effetto che sono note, prevedibili e ripetibili. Un esempio di progetto che può essere categorizzato in questo ambiente è quello della costruzione di una casa in un contesto edilizio. Difatti, sebbene ogni progetto è diverso dagli altri, le relazioni causa – effetto sono ben chiare. In questo ambiente l’approccio che si può adottare è quello di

percepire – categorizzare – rispondere ovvero percepire le particolarità di questa progettualità, categorizzarle e rispondere. In questo dominio è pertinente il concetto di “Best Practice”.

- Complicato: in questo dominio sono ancora presenti le relazioni causa – effetto, ma non sono prevedibili come nel caso precedente. Difatti, in questo ambiente vi è la necessità di analizzare tali relazioni poiché non risultano essere evidenti. Un esempio di progettualità che può essere considerata complicata è la costruzione di un ponte, in quanto bisognerà studiare il contesto in cui esso deve essere costruito, il traffico che dovrà sostenere, i corsi d’acqua che dovrà attraversare etc. L’approccio che può essere utilizzato in questo caso è *percepire – analizzare – rispondere* e la gestione del progetto sarà considerata “Buona Pratica”.

In questi primi due ambienti l’approccio risulta essere di tipo **predittivo**, e pertanto sono contesti in cui si riesce a pianificare e schedulare il progetto in dettaglio. Per tale motivo la metodologia progettuale che è preferibile adottare è la **Waterfall**.

- Complesso: in questo dominio le relazioni causa – effetto sono imprevedibili. Questo ambiente è costituito da alcune caratteristiche come ad esempio un’alta volatilità, data dal fatto che i bisogni possono cambiare velocemente. Inoltre, vi è un’elevata incertezza in quanto non è facile prevedere cosa accadrà in futuro. Un esempio di progetto potrebbe essere quello di un prodotto disruptive. L’approccio utilizzato in questo dominio è *sperimentare – percepire – rispondere*, in quanto si tratta di un approccio empirico in cui sarà necessario sperimentare e testare azioni per definire dettagliatamente lo scopo del lavoro. Maggiore è la complessità, maggiore sarà la flessibilità del team per rispondere ai cambiamenti.
- Caotico: in questo dominio non esistono relazioni causa – effetto. Si adatta particolarmente a progetti con un tasso elevato di innovazione e con un’alta

propensione al rischio. L'approccio utilizzato in questo dominio è *agire – percepire – rispondere*.

Questi due ambienti, invece, sono caratterizzati da un approccio **adattivo**. Si tratta di contesti in cui la responsabilità è condivisa, e le decisioni sono distribuite tra gli attori in modo da fronteggiare il repentino cambiamento delle condizioni a contorno e degli obiettivi progettuali. Pertanto, la comunità di Program, Project e Portfolio Management per gestire questi domini cerca di attuare strategie di sviluppo iterativo come **Agile** o **Concurrent Engineering**.

Infine, è possibile trovare l'ultimo ambiente, il disordine, in cui viene collocato tutto ciò che non riesce ad essere categorizzato negli altri ambienti. È importante sottolineare che è possibile oltrepassare i confini tra i diversi ambienti, e dunque, un progetto che inizialmente era stato categorizzato come complesso può diventare semplice e viceversa.

CAPITOLO 2: Descrizione di Reply e di come applica Waterfall e Agile

2.1. Reply S.p.A e Cluster DCX

Questo lavoro di tesi è stato svolto presso l'azienda Reply. Si tratta di una società nata e fondata a Torino nel 1996, che offre servizi di **consulenza** in diversi settori quali IoT, CX & Digital Commerce, Cloud Computing, Data World, Quantum Computing e tanti altri ancora. Reply opera in diversi mercati, in particolare quello *Automotive, Energy & Utility, Financial, Manufacturing, Logistics, Healthcare, Retail & Consumer Products e Telco & Media*. In ogni segmento citato, il gruppo concilia competenze del settore specifico e esperienza nei diversi servizi offerti. Difatti, oltre ad offrire un servizio di consulenza, Reply offre anche servizi di **System Integration** per poter sfruttare nel migliore dei modi possibili le potenzialità di soluzioni tecnologiche innovative. Offre anche un servizio di **Digital Services**, ovvero propone soluzioni innovative basate sui nuovi trend digitali e sui nuovi canali di comunicazione. Con più di 12.500 persone in 16 Paesi distribuiti tra Europa, Africa, Asia, America e Oceania, l'azienda si propone di rendere concreta la rivoluzione digitale attraverso competenze verticali e una profonda passione per l'innovazione. Come è riportato nel resoconto al 30 settembre 2022, nei primi nove mesi dell'anno, il fatturato del Gruppo Reply, ha raggiunto i 1.355,7 milioni di Euro. Tale risultato risulta essere in crescita del 26,3% rispetto a quello registrato al 30 settembre dell'anno precedente, che risultava essere pari a 1.073,4 milioni di Euro ^[1]. Durante questo periodo, sono risultati positivi tutti gli indicatori. Difatti, L'EBITDA consolidato è stato di 218,6 milioni di Euro, in crescita del 19,3% rispetto a quello relativo a settembre 2021. Di conseguenza, anche l'EBIT, relativo ai mesi da gennaio a settembre, è risultato essere in crescita del 24,5%, rispetto ai 147,5 milioni di Euro a settembre 2021, per un valore pari a 183,6 milioni di Euro. Infine, l'utile ante imposte relativo a questo periodo è stato di 169,1 milioni di Euro, un risultato positivo rispetto ai 150 milioni di Euro del rispettivo raggiunto nel 2021. Per tutte

queste ragioni elencate, la posizione finanziaria del Gruppo Reply al 30 settembre 2022 risulta essere positiva per un totale di 133,9 milioni di Euro, riuscendo a raggiungere risultati positivi sia in termini di marginalità che di fatturato. È importante sottolineare che Reply, è in continua espansione; difatti, da inizio 2022 il gruppo ha esteso la propria presenza a livello internazionale, ponendo particolare attenzione a tre diversi mercati: *Americano, Francese e Tedesco*. Per quanto concerne la struttura dell'azienda, bisogna specificare che, il Gruppo Reply è organizzato in tante company verticali che trattano tecnologie differenti. Ogni company affianca i principali attori industriali, aiutandoli nella definizione di modelli di business abilitati dai nuovi paradigmi tecnologici e di comunicazione digitale. In particolare, la stesura della tesi è stata effettuata nella company Cluster DCX (Dynamics Customer eXperience), che è specializzata in consulenza, system integration e servizi digitali. Difatti, Cluster offre piattaforme Customer Relationship Management (CRM), integrate con quelle *Low Code – No Code* basate su tecnologia Microsoft, arricchite con analisi dei dati, Artificial Intelligence (AI) e moderne tecnologie di lavoro. Questa scelta, trova le motivazioni nel fatto che Cluster Reply risulta essere un partner Microsoft certificato a livello Gold e, pertanto, sviluppa il proprio business in stretta collaborazione con le filiali locali di Microsoft. Parlare di CRM vuol dire parlare di una progettualità a 360°, difatti sono eseguiti tutti gli step legati all'ambito del project management: raccolta dei requisiti, analisi, gestione di tempi/costi/qualità etc. A sua volta, la company Cluster è suddivisa in Business Unit, ognuna delle quali offre soluzioni CRM in un diverso settore del mercato. In particolare, i principali clienti della Business Unit con cui è stato portato avanti il progetto, e dunque il lavoro di tesi, appartengono al settore automotive, sia in contesto dinamico che internazionale. Reply, inoltre, nel 2021 è stata posizionata come Leader di mercato da Gartner, che è una società di ricerca e consulenza leader a livello mondiale, nel Magic Quadrant for CRM and Customer Experience Implementation Services. Il report ha valutato 16 fornitori di servizi di implementazione CRM e CX in tutto il mondo ^[12]. L'approccio di *“Human Centric*

Design” utilizzato da Reply per affrontare una progettualità, consente alla società stessa di comprendere, progettare e implementare soluzioni complesse di CRM e Customer Experience, le quali combinano efficacemente l'analisi del Customer Journey, la User eXperience (UX) design, la definizione dei processi, le piattaforme CRM, le architetture tecnologiche e l'analisi dei dati.

2.1.1. Ripartizione dei Progetti nella Business Unit di Cluster DCX

Prima di analizzare come il Gruppo Reply applica la metodologia progettuale Waterfall e quella Agile, è importante analizzare come sono gestiti i progetti in Cluster Reply, e più in particolare nella Business Unit in cui è stato portato avanti questo lavoro di tesi. Bisogna tener presente che questa scelta non dipende unicamente dall'azienda fornitrice, ma dipende molto dall'organizzazione e dalla cultura aziendale dei clienti. Nello specifico, risultano essere attive 22 progettualità di cui 20 con dei clienti del settore automotive appartenenti allo stesso gruppo aziendale che saranno indicati come “*cliente 1*”, e gli altri due progetti con un altro cliente, indicato come “*cliente 2*”. Andando a suddividere i progetti in base al cliente, si può notare come il contesto aziendale influisca sulla modalità progettuale scelta dal cliente. Difatti, il “*cliente 1*” con cui sono portate avanti la maggior parte delle progettualità ovvero il 91% circa, risulta avere uno stampo industriale in cui le relazioni causa – effetto sono note. Inoltre, il cliente è molto legato alle routine aziendali e, fino a questo momento, poco propenso al cambiamento delle stesse. Per tali ragioni, nella classificazione del Cynefin Framework descritta nel *paragrafo 1.6.1* si posiziona nel dominio definito “*semplice*” in cui si preferisce utilizzare un approccio classico. Difatti, questo cliente ha sempre preferito che i progetti fossero gestiti in modalità Waterfall, e solo in questo momento ha deciso di sperimentare la modalità progettuale Agile. In questo contesto di **cultura aziendale**, dunque, solo il 5% delle progettualità è portato avanti in modalità Agile, mentre il 95% è svolto con la metodologia Waterfall (*Figura 2.1*).

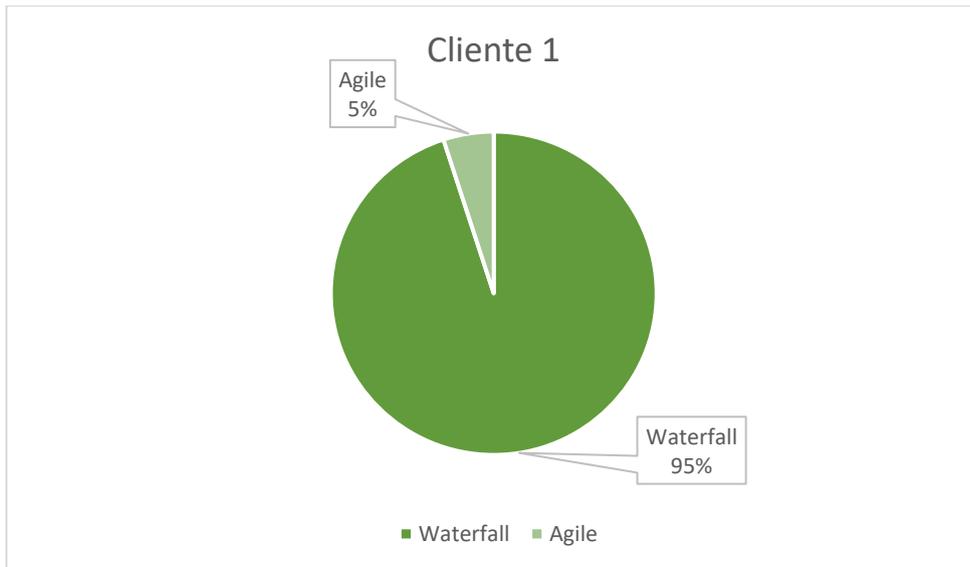


Figura 2.1 Ripartizione Waterfall/Agile per il Cliente 1

Il "cliente 2", invece, si trova in un contesto totalmente diverso da quello descritto precedentemente. Difatti, questa azienda risulta avere uno stampo decisamente più innovativo e risulta essere molto propensa e flessibile ai cambiamenti. Difatti, il 100% dei progetti portati avanti per questa azienda sono svolti in modalità Agile (Figura 2.2).

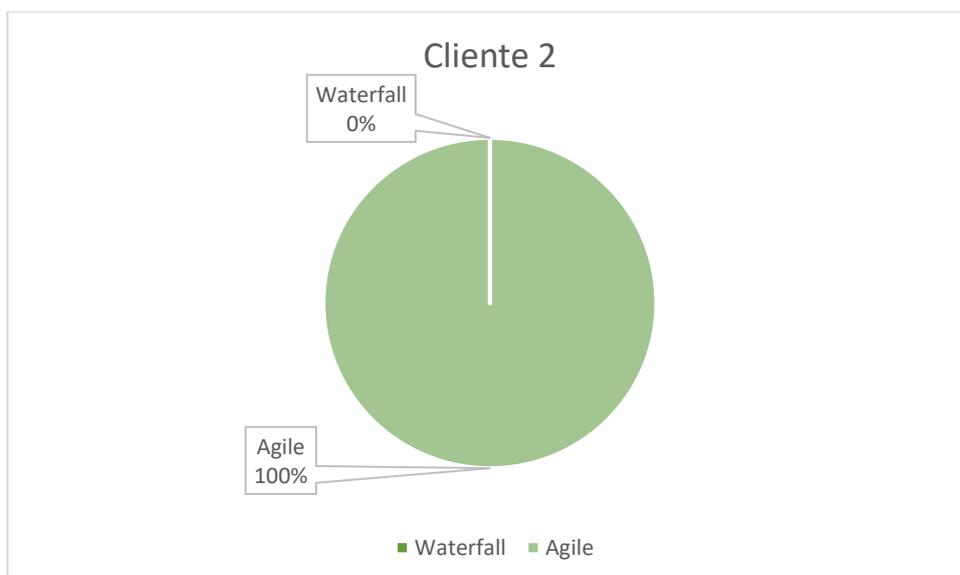


Figura 2.2 Ripartizione Waterfall/Agile per il Cliente 2

Da questi dati si evince come, uno dei fattori determinanti nella decisione della modalità progettuale risulta essere la cultura aziendale. Nonostante ciò, sarà analizzata nei capitoli successivi la risposta del “cliente 1” alla metodologia Agile. A valle di questa panoramica, è utile specificare che la gestione delle progettualità cambia tra le diverse Business Unit, poiché si trattano clienti appartenenti a settori diversi, e varia anche tra le diverse company di Reply. Pertanto, l’analisi di quanto accade nella Business Unit descritta, non rispecchia l’andamento del gruppo Reply nella sua interezza, il quale, per questioni di privacy non può essere descritto. Si può inoltre sottolineare che, all’interno della Business Unit vi sono un totale di 46 dipendenti, di cui:

- 3 Scrum Master;
- 9 Project Manager.

È importante porre l’attenzione sul fatto che le figure di Project Manager e Scrum Master non sempre coincidono. Questo perché guardando le caratteristiche generali e le mansioni dei due ruoli è possibile notare come questi siano molto differenti. Difatti, la caratteristica principale che è richiesta a un Project Manager è quella di essere il **leader** del progetto e del team, risultando la figura che ha la responsabilità sul raggiungimento degli obiettivi del progetto stesso. Il Project Manager, dunque, ha il compito di guidare la pianificazione delle attività per l’esecuzione del progetto, e far in modo che i vincoli di tempi, costi e qualità siano rispettati. Lo Scrum Master, invece, risulta essere un **coach**, responsabile dell’efficacia dello Scrum Team. Questo è reso possibile permettendo al Team di migliorare le proprie pratiche definite nel framework Scrum. Lo Scrum Master ad esempio si pone l’obiettivo di:

- allenare i membri del team all’autogestione e alla cross-funzionalità;
- aiutare lo Scrum Team a concentrarsi nel creare Increment di grande valore che incontrino la Definition of Done;
- eliminare gli impedimenti all’avanzamento dello Scrum Team.

Difatti, lo Scrum Master non ha il compito di gestire i componenti del team assegnando loro task giornalmente, in quanto il development team in un contesto Agile si autogestisce, ma dovrebbe fluidificare la comunicazione e gestire i conflitti. Inoltre, si pone come obiettivo principale quello di promuovere la teoria Scrum per far sì che sia compresa sia all'interno del team che all'esterno. In definitiva, il Project Manager è considerato colui che gestisce aspetti come tempi, costi e qualità, mentre lo Scrum Master è colui che cerca di mettere d'accordo le varie parti e promuove la teoria Scrum all'interno dell'organizzazione. Solo nel caso in cui si abbia la **doppia certificazione**, ovvero sia quella per essere Scrum Master, che quella per essere un Project Manager, una risorsa può ricoprire entrambi i ruoli. Difatti, nel caso specifico della progettualità oggetto di tesi, ci si è ritrovati in questa situazione; la figura che nella sotto commessa svolta in modalità Waterfall risultava essere il Project Manager, nella seconda fase progettuale ha ricoperto il ruolo di Scrum Master.

2.2. Come Reply applica la metodologia Waterfall

Tra le varie metodologie legate al mondo del Project Management, l'approccio Waterfall è il più utilizzato. In questo approccio, le fasi e le sotto fasi di lavoro sono eseguite in sequenza, e ogni fase è svolta una sola volta seguendo un modello a cascata. Le fasi, dunque, si susseguono in ordine temporale e, il ritorno a una fase precedente, è previsto solo nel caso di azioni che attivano una revisione. La consegna del sistema è prevista solo al termine della fase di Deployment ed è quindi unica, nel senso che avverrà nel momento in cui tutte le funzioni incluse nei requisiti dell'utente sono disponibili. Nonostante la modalità Waterfall risulti essere rigida, è lasciata alla discrezione del Project Manager una certa flessibilità nel passaggio da una fase all'altra, il quale si farà carico di eventuali costi di gestione e si assumerà i rischi ad essi associati. In Reply, l'approccio Waterfall viene applicato quando:

- I requisiti dell'utente sono consolidati, chiari e dettagliati e la probabilità che ci siano modifiche durante il progetto è limitata;
- Il progetto è di piccole/medie dimensioni con un profilo di rischio relativamente basso;
- La durata del progetto non è eccessiva e la verifica di fattibilità sarà effettuata in tempi relativamente brevi.

In questi progetti il team risulta essere piccolo e, i fattori di rischio legati alla tecnologia utilizzata, ai tempi di consegna, ai parametri finanziari/contrattuali o al rapporto con il cliente, sono relativamente bassi. Bisogna specificare che il contesto di riferimento della metodologia descritta è il **progetto software**, ovvero l'insieme delle attività che trasformano i requisiti dell'utente in un sistema software utilizzato nell'ambiente di produzione. In questo senso, la creazione di progetti non comporta solo lo sviluppo di funzioni, ma comprende una serie di attività complesse regolate da accordi contrattuali e finanziari stabiliti e consolidati, nonché una serie di attività tecniche per la trasformazione e l'elaborazione del software e per l'ingegnerizzazione della soluzione. L'applicazione del modello, e più specificamente l'avvio delle attività di progetto, dipendono dal soddisfacimento di una serie di condizioni di partenza e di ipotesi di riferimento. In particolare, all'inizio del progetto devono essere disponibili le seguenti informazioni:

- I requisiti dell'utente: sono normalmente definiti dal cliente e possono essere resi disponibili sotto forma di un unico documento strutturato come ad esempio un documento di specifiche, oppure possono essere contenuti in diversi documenti di progetto. Prima di iniziare il progetto, il Project Manager deve verificare che tutti i requisiti siano coerenti, completi e condivisi dagli stakeholder;
- Le fasi del progetto e il piano di rilascio: devono essere indicati sia i principali obiettivi in termini di tempistica come ad esempio la data di inizio del

progetto, la data di GoLive, etc., sia il piano di rilascio stabilito per le diverse versioni del software (release);

- Eventuali standard e linee guida imposte dal cliente: sia la metodologia di riferimento, che gli standard operativi da applicare all'intero progetto devono essere noti e resi disponibili;
- Termini di accettazione/verifica del sistema: devono, infine, essere descritte le procedure di accettazione e verifica del sistema. Ad esempio, se la verifica è prevista o meno in produzione, come deve essere eseguito il test di accettazione, etc.

Quando tutte queste informazioni sono note e condivise con tutti gli attori coinvolti, inizia lo svolgimento del progetto. Il ciclo di vita del progetto software è costituito da una serie di fasi e sotto fasi, all'interno delle quali è previsto il completamento di una serie di attività che portano alla creazione dei deliverable di lavoro corrispondenti. Di seguito, è riportato uno schema delle fasi previste dell'azienda quando applica l'approccio Waterfall (*Figura 2.3*).

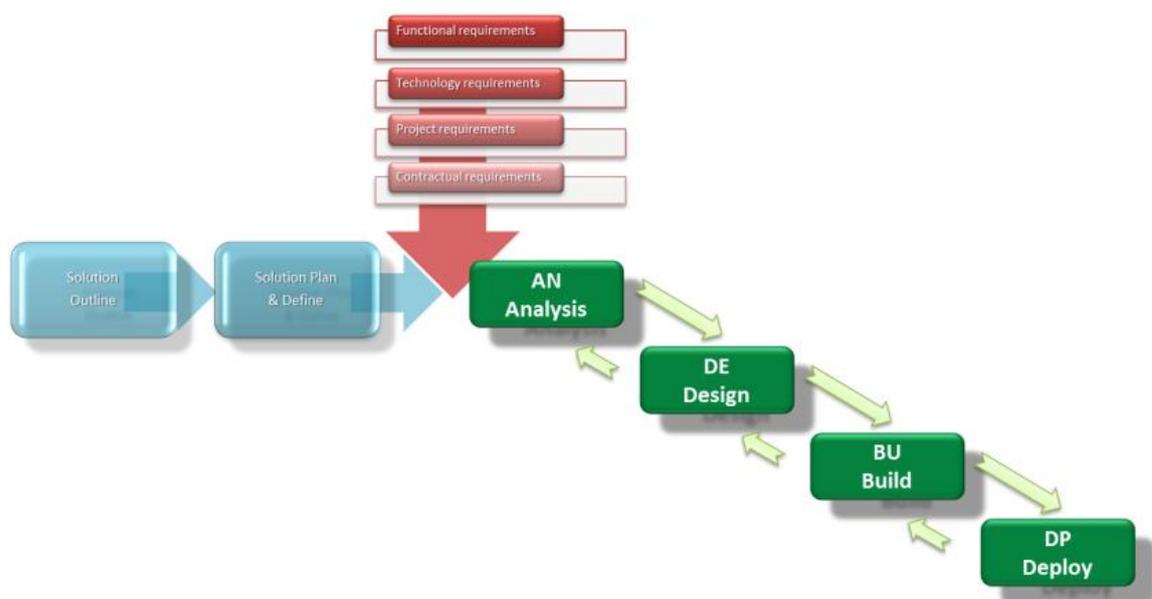


Figura 2.3 Approccio Waterfall

2.2.1. Analisi

La fase di **Analisi** ha come scopo quello di definire una serie completa e coerente di requisiti software che siano in grado di soddisfare le richieste del cliente, considerate l'input di questa prima fase. Bisognerà analizzare tali requisiti per riuscire a creare modelli di progetto software in modo da descrivere ciò che il sistema deve fare secondo diverse prospettive. La fase di Analisi è una fase ampia e complessa ed è pertanto suddivisa in una serie di sotto fasi:

- Descrizione del processo: bisognerà rappresentare un modello all'interno del quale saranno mappati e descritti i processi aziendali coinvolti nella creazione del progetto software. In questa prima sotto fase sarà sviluppato il *process flow* in dettaglio e saranno analizzate le diverse interazioni con le funzioni, gli attori coinvolti e i servizi forniti dal sistema;
- Macroanalisi: questa fase prevede la descrizione del sistema attraverso due diagrammi ovvero il *Data Flow Diagram* (DFD), il quale descrive il funzionamento del sistema mediante il flusso concettuale dei dati scambiati tra i vari processi applicativi e l'*Application Flow Diagram* (AFD), il quale descrive il funzionamento del sistema attraverso il flusso applicativo tra funzioni, servizi interni ed esterni rilevanti per lo sviluppo del sistema.
- Analisi funzionale: in questa sotto fase, si descrive in maniera dettagliata il comportamento del sistema in termini di requisiti software. Si tratta della specifica tecnica più importante in quanto è l'oggetto della revisione da parte del Cliente ed è il riferimento principale per la progettazione dei test del sistema.

In definitiva, si può affermare che il fine ultimo della fase di Analisi è quello di descrivere e definire cosa dovrà fare il sistema

2.2.2. Design

Lo scopo della fase di **Progettazione** (DE) è quello di descrivere il processo dal punto di vista tecnico. In sostanza, si riferisce alla fase di progettazione tecnica, che precede le successive attività di codifica, configurazione e assemblaggio. L'attività principale di questa fase è la progettazione tecnica dei componenti e dei modelli definiti nella precedente fase di Analisi. Per ogni componente dell'applicazione, bisogna ottenere specifiche tecniche sufficientemente dettagliate che possano fungere da input per gli sviluppatori nella successiva fase di Build. Questa fase ha dunque come input generale tutto ciò che viene prodotto nella fase di Analisi. L'output principale previsto per questa fase è l'*Analisi Tecnica*, la quale dipende strettamente dal modello architettonico stabilito. Inoltre, deve essere sviluppato anche il modello di DB logico, il quale traduce il modello concettuale in un modello di relazioni che può essere implementato in uno specifico sistema di gestione di database.

2.2.3. Build

Lo scopo della fase **Build** è quello di completare le attività di coding, configurazione e assemblaggio, in modo che il progetto software sia poi creato, testato e verificato, in relazione ai Requisiti del Software stabiliti nella fase di Analisi Funzionale. È importante notare come la struttura della fase di Build dipenda strettamente dal tipo di attività svolta, ovvero dalla richiesta del cliente. Nello specifico, possono esserci casi in cui bisogna sviluppare del nuovo codice per soddisfare i requisiti del cliente, e casi in cui la richiesta risulta essere più standard. In questi casi è possibile procedere alla realizzazione del software mettendo insieme dei componenti già esistenti. La fase di Build ha come input generale tutto ciò che viene prodotto nella fase di Design che la precede. L'output di questa fase, sarà la piattaforma funzionante e testata nell'ambiente di sviluppo. Difatti, è necessario che siano effettuati i test per identificare eventuali bug del sistema.

2.2.4. Deploy

Lo scopo della fase di **Deploy** è rendere il software operativo nell'ambiente di produzione. L'input di riferimento per la fase di Deployment è quanto prodotto durante la fase di Build. I prerequisiti per il Deployment sono due: il primo è che il sistema abbia effettivamente superato i test di base effettuati nella fase di Build; il secondo è che la verifica di quanto è stato prodotto, svolta insieme al Cliente, abbia avuto esito positivo. A questo punto può iniziare il **Go-Live**, che identifica l'inizio del periodo di Roll-Out, durante il quale il sistema software viene progressivamente reso disponibile e utilizzabile nella pratica. L'output di questa fase prevede che il software sia rilasciato nell'ambiente di produzione e che sia effettivamente funzionante. In alcuni casi, è previsto anche il rilascio di un *manuale*, il quale deve contenere tutte le informazioni necessarie agli utenti per utilizzare correttamente il sistema e per diagnosticare le principali problematiche. Anche in questa fase è necessario che siano effettuati i test per verificare il corretto funzionamento del sistema ed identificare eventuali bug. Sarà solo in questo momento che il cliente avrà modo di osservare quanto è stato svolto. Pertanto, se dovessero essere richieste delle modifiche in questo momento, si potrebbero generare ritardi nella consegna.

2.3. Come Reply applica Agile

Reply sostiene l'utilizzo del framework Scrum come approccio Agile per lo sviluppo iterativo e incrementale del software. Il termine "Agile" è inteso come termine generico per descrivere la natura iterativa della fornitura, mentre "Scrum" è un approccio esplicito allo sviluppo del software. In un tipico modello di rapporto con i potenziali clienti, Reply suddivide il processo di gestione di un progetto in tre macro step: offerta, esecuzione e chiusura, come riportato in *Figura 2.4*. Nel primo step, sono previste due fasi: proposta e offerta.

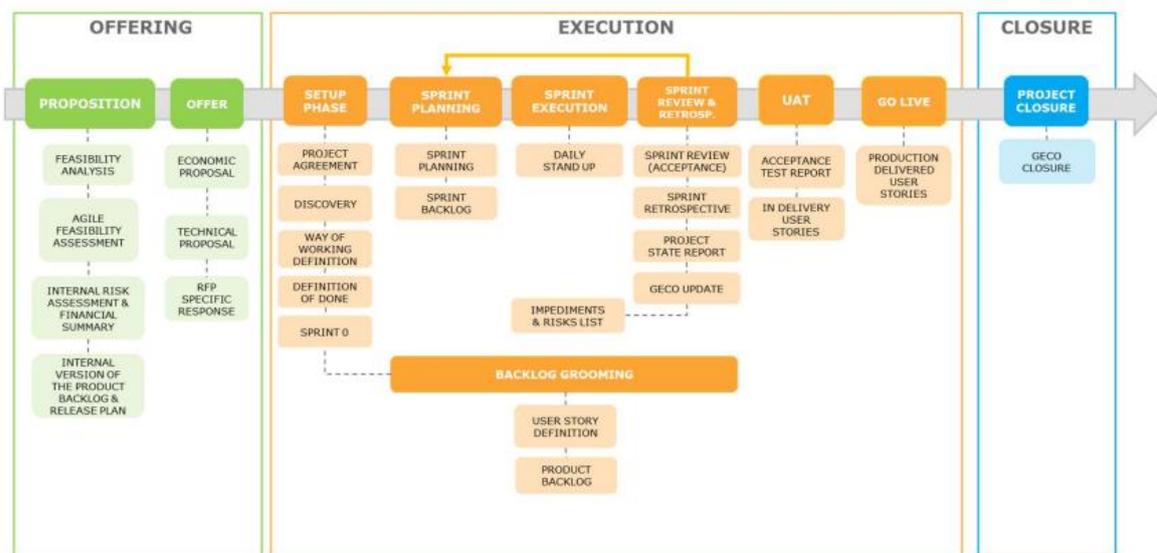


Figura 2.4 Agile

2.3.1. Offerta

Si inizia normalmente con la ricezione della **Request For Proposal (RFP)** inviata dal cliente. L'RFP, anche nota come richiesta di proposta, è un documento utilizzato da molte organizzazioni per ricevere le offerte di beni o servizi da potenziali venditori. A questa segue un'analisi di fattibilità, effettuata da Reply, che ha lo scopo di fornire una risposta completa al cliente ed è guidata da Bid Manager. Questa attività comprende: l'analisi dei requisiti, la risposta alla soluzione tecnica, una stima iniziale di alto livello di profili e tempistiche, ed infine, un'analisi dei rischi e delle dipendenze. Successivamente deve essere effettuata "un'analisi di fattibilità Agile", in cui si analizza l'opportunità di consegnare il progetto seguendo il framework Scrum. Questo viene reso possibile dalla matrice di fattibilità/rischio Agile, la quale mira a fornire un approccio comune all'interno di Reply per valutare in che misura possa valere la pena passare alla metodologia Agile nell'affrontare un'opportunità di progetto. Infine, sarà effettuata una valutazione interna dei rischi e il riepilogo finanziario. Talvolta può essere creata una versione iniziale del Product Backlog e del Release Plan; il Bid Manager prepara un Product Backlog iniziale che si allinea alle caratteristiche chiave da sviluppare, idealmente con la partecipazione del

Cliente. Dopo tutte le attività di analisi sopra citate, si procede con l'**offerta**. In questa fase il Bid Manager presenta al cliente: un'offerta economica, un'offerta tecnica in cui è descritta la soluzione da un punto di vista tecnico e la risposta all'RFP con l'elenco di tutti i deliverable previsti nell'offerta. Una volta che il cliente accetta formalmente l'offerta, si passa allo step dell'esecuzione.

2.3.2. Esecuzione

È importante sottolineare che, a differenza del modello teorico Agile spiegato nel paragrafo 1.5, nella metodologia Agile applicata da Reply è prevista una fase iniziale, definita **Fase di Setup** che a sua volta è composta da diverse sotto fasi:

- Impostazione del progetto: il Delivery Manager divide l'ordine in sottordini. Per ogni sottordine, il Delivery Manager definisce il budget complessivo, le previsioni per i tre mesi successivi e vincola il team Scrum al sottordine stesso;
- Discovery: per i progetti di grandi dimensioni, o se richiesto dal cliente, viene stabilita una "fase di scoperta" che ha lo scopo di definire i requisiti (attraverso workshop congiunti con il cliente), dettagliare il progetto architettonico, definire il piano generale e il prezzo prima dell'inizio della costruzione. Le fasi di scoperta presentano una durata variabile, che in genere risulta essere compresa tra le 8 e le 12 settimane;
- Kick off Meeting: l'incontro iniziale di kick-off, gestito dallo Scrum Master, ha lo scopo di definire lo scenario dell'intero sviluppo, stabilire le regole di base per le modalità di lavoro dello Sprint, concordare la definition of Done e stabilire le priorità per lo Sprint immediato. L'output di questa riunione è il documento di *Project Agreement*, in cui sono elencati l'elenco dei deliverables, la definizione del team Scrum e l'obiettivo di progetto.
- Definizione delle modalità di lavoro: viene stilato un documento che illustra l'intero ciclo Scrum, in cui sono definiti diversi aspetti come la durata degli Sprint, le cerimonie, il processo di accettazione, i ruoli e le responsabilità. È un

documento di lavoro e può essere modificato attraverso azioni correttive durante le Sprint Retrospective o in qualsiasi momento se il team Scrum, compreso il Product Owner, è d'accordo sulla modifica;

- Definition of Done: si tratta di un documento che definisce i criteri di qualità in base ai quali la consegna è ritenuta accettabile per essere considerata "DONE". È un documento di lavoro e, come quello precedentemente descritto, può essere modificato attraverso azioni correttive durante le Sprint Retrospective o in qualsiasi momento se il Team Scrum, compreso il Product Owner, è d'accordo sulla modifica;
- Sprint 0: in alcuni casi è possibile trovare uno "*Sprint 0*" il quale risulta essere importante ed è orientato all'avvio dell'intero processo di sviluppo attraverso la creazione della Build Pipeline, l'impostazione degli ambienti, gli strumenti di sviluppo, la struttura iniziale del codice, gli strumenti e i framework di test, il repository del software, la connettività VPN del cliente etc. In sostanza, in questo sprint viene effettuato tutto ciò che serve per essere pronti all'implementazione. La durata dipende dalle dimensioni del progetto; in genere, per un progetto medio/grande si tratta di un ciclo di sprint più lungo del normale, che va fino alle 4 - 6 settimane di durata. Per un progetto di piccole dimensioni, invece, la durata è la stessa degli altri Sprint. Anche se viene chiamato "Sprint 0", questa fase opzionale differisce dagli Sprint standard e non può essere considerata un vero e proprio Sprint perché non produce un incremento di prodotto ragionevole.

Dopo questa fase di setup, si parte con l'esecuzione vera e propria del progetto. Il team esegue ogni Sprint come descritto nel framework Scrum, senza modifiche alle cerimonie (pianificazione, stand-up giornaliero, demo e retrospective). Tipicamente si inizia effettuando lo **Sprint Planning** meeting: questo evento è un momento di pianificazione per il team, insieme al Product Owner, che ha lo scopo di stabilire un piano di consegna per il ciclo di Sprint che sta iniziando. È importante notare che

solo il team può accettare e rifiutare le User Story non pronte per lo sviluppo. Questa cerimonia di pianificazione tipicamente ha questi obiettivi:

- Confermare lo scopo dello Sprint con il Product Owner;
- Assicurare la corretta comprensione del Product Backlog e delle user story;
- Rivedere la stima del lavoro;
- Determinare, in base alla capacità e alla velocità, cosa può essere consegnato;
- Finalizzare lo Sprint Backlog;

A questo punto può iniziare l'**esecuzione dello Sprint**: questa macro-attività riguarda tutto il lavoro svolto dal team per raggiungere l'obiettivo finale. Le pratiche e gli eventi che si verificano all'interno di uno Sprint sono definiti attraverso i documenti "*Ways of Working*" e "*Definition of Done*". Questa fase comprende anche i test di integrazione/sistema; nello specifico, il team di sviluppo esegue tutti i test tecnici per convalidare le User Stories in vista della consegna, secondo gli accordi di progetto. Si passa poi alla **Sprint Review**, ovvero una riunione per mostrare ciò che è stato realizzato nell'ultimo Sprint. A seconda delle modalità di lavoro, la Sprint Review può essere un evento singolo o multiplo e può essere utilizzata come parte della Progressive Acceptance. In genere avviene l'ultimo giorno dello Sprint. Successivamente viene effettuata la **Sprint Retrospective**, che si svolge dopo lo sprint e idealmente prima della successiva sessione di pianificazione dello sprint. È una riunione interna del team, in cui si riflette sull'andamento dell'ultimo sprint e ha l'obiettivo di identificare eventuali azioni correttive da intraprendere per migliorare alcuni aspetti per lo sprint successivo. Esistono diverse forme di presentazione e strumenti che possono essere utilizzati, ma la cosa principale è che il team si concentri solo su alcuni elementi (2 o 3) da migliorare per lo sprint successivo. Alla fine di ogni sprint, il Delivery Manager o, in caso di team piccoli lo Scrum Master, aggiorna il sottordine in base a quanto accaduto nello Sprint stesso. Un'altra fase dell'esecuzione sono gli **User Acceptance Test (UAT)** mediante i quali il prodotto viene testato per verificare se

soddisfa i requisiti funzionali e non, così che si possa successivamente procedere con la consegna in produzione. Il responsabile dell'esecuzione dell'UAT è il Product Owner che può essere aiutato da un team dedicato di tester o da un gruppo di utenti reali. È possibile eseguire questa fase di UAT all'interno di ogni ciclo di sprint, e dunque come parte dello sprint stesso, oppure separatamente attraverso una sessione ad hoc, tipica dei grandi progetti. Prima di procedere con la consegna bisogna ottenere l'*autorizzazione formale* da parte del Product Owner o degli stakeholder; successivamente il team avvia il processo di rilascio in produzione. Dopo la fase di Go Live si passa allo step di **Chiusura**. Lo Scrum Master, dunque, rilascia lo Scrum Team al termine delle attività di sviluppo. Saranno raccolte le conoscenze sviluppate nel progetto, per condividerle eventualmente con i progetti futuri, come ad esempio:

- pratiche di sviluppo che hanno responsabilizzato il team e gli hanno permesso di gestire l'incertezza di Agile;
- conoscenza di strumenti di successo, introdotti per il progetto e non già patrimonio dell'azienda;
- librerie di recente sviluppo, esperienza su API e framework, frammenti di codice degni di nota per applicare i pattern;
- evidenze storiche di consegna come Burndown chart, Velocity chart, BurnUp chart etc.

2.3.3. Template Agile utilizzati da Reply

Di seguito sono riportati alcuni dei template più importanti che Reply utilizza per la gestione dei progetti in Agile. Saranno riportati rispettivamente nelle *Figure 2.5, 2.6, 2.7 e 2.8* i template del Ways of Working (WoW), Definition of Done (DoD), Product Backlog e il documento di Project Agreement.

Sprint Cycle Event	2 weeks Time/Day	Attendance	Output
Sprint Planning	WK1 – Monday @9.00am/11.00am	Scrum Team*, Product Owner	Sprint Goal, Sprint Backlog ready, Sprint Started.
Daily Stand-up	Daily @9.30am/9.45am	Developer*, Scrum Master, Product Owner	Updated Sprint Board
Sprint Review	WK2 – Friday @ 9.00am/11.00am	Scrum Team, Product Owner & other Customer Stakeholders	Demonstration of completed functionality. Inspection & Adaptation.
Sprint Retrospective	WK2 – Friday @ 2.00pm/3.00pm	Scrum Team, Product Owner	Potentially PBI.
Release	WK2 – Friday @1.00pm →	Developer*, Scrum Master, Product Owner	Product Increment release.

Figura 2.5 Template WoW

#	Description	Status
1	Standards: Source Code has been written in accordance to agreed Coding Practices.	OK
2	CheckIn: Source Code, Unit Tests & QA Tests have been checked into agreed sw repository.	OK
3	Static Analysis: Static Analysis has been run and no Blocker or Critical Issues found.	OK
4	Build: Source Code caught up with the main code branch builds cleanly no errors.	OK
5	Unit Testing: Unit testing is completed and runs cleanly and adheres to agreed criteria.	NOK (55% cond, 70% func)
6	QA Testing: All QA Manual or Automated Tests run without errors.	OK
7	NFR: Performance and Security based testing conducted and no major discrepancies observed	OK
8	Tech Documentation: All appropriate technical documentation updated	OK
9	Integration Testing: 'Nearest Neighbour' testing against other customer work streams has been conducted.	OK
10	Defects: All Critical & Major issues have been resolved, no significant technical Debt outstanding.	OK

Figura 2.6 Template DoD

Summary			
Total Estimate	25	Sprint Number	2
Total Accepted User Stories	4	Total Left User Stories	6

Epic N°	User Story N°	Description	Business Value	Estimate	Priority	Status	Sprint Execution	Acceptance Date
1	1.1	As a not logged user I want to be able to register a new account so that I could log-in	7	5	1	Accepted	1	07/12/2022
1	1.2	As a registered not logged user I want to be able to log-in so that I could receive notifications	7	4	2	Pending	1	
1	1.5	As a registered not logged user I want to be able to recover my password so that I could log-in when i lost it	3	1	5	Rejected	2	07/12/2022
2	2.1	As a logged User I want to change the settings of the notification so that I could easily manage what type of notifications i want receive	6	4	3	Accepted	2	09/01/2023

Figura 2.7 Template Product Backlog

È importante sottolineare che le User Story presenti nella *Figura 4.5* risultano essere del tutto esemplificative di quanto si tiene traccia con questo documento.

Project Goal

The client needs to:

- *Have a tool able to report and manage in a simple way the Issues that emerge during the development phases of a new product.*
- *Provide fast support through tool implementation to facilitate harmonized reporting and ensure an overall view on Issues Management status.*
- *Set a common methodology for Issues assessment to facilitate the identification of priorities of intervention.*
- *Provide a unique source of input for the solvers and support who has to collect the issues and populate the tracking files.*

Project Management Framework - SCRUM

SCRUM Framework

Agreements over SCRUM

Sprint Duration	<i>The duration of a scrum iteration is two weeks</i>
Definition of Done	<i>The definition for which a user story can be considered DONE for example "all UAT are performed correctly", this definition can be refined for each user story where necessary.</i>

Deliverables List

List of deliverables in the current project:

- *Project Agreements (REY-AGILE-RM)*
- *Ways of Working (REY-AGILE-WOW)*
- *Definition of Done (REY-AGILE-DOD)*
- *Product Backlog (REY-AGILE-PB)*
- *User Story Definition (REY-AGILE-USD)*
- *Project State Report*

Team Composition		
Product Owner	Name	<i>Customer's Manager</i>
	Mail	<i>mail</i>
	Phone Number	<i>phone number</i>
Scrum Master	Name	<i>Laura Tornatore (Reply)</i>
	Mail	<i>mail</i>
	Phone Number	<i>phone number</i>
Team Developer	Name	<i>Reply's Employee</i>
	Mail	<i>mail</i>
	Phone Number	<i>phone number</i>
Team Developer	Name	<i>Reply's Employee</i>
	Mail	<i>mail</i>
	Phone Number	<i>phone number</i>

Figura 2.8 Template Project Agreement

Nel template sopra raffigurato, per motivi di privacy non sono indicati i contatti e i nominativi dei componenti del team.

CAPITOLO 3: Development Issue Management

3.1. Struttura del progetto DIM & Power Platform

Il progetto di cui tratterà questo lavoro di tesi è il Development Issue Management, realizzato per un cliente del settore automotive. Il cliente ha deciso di avviare un progetto per poter disporre di uno strumento in grado di segnalare e gestire in maniera semplice ed efficace le anomalie che emergono in fase di sviluppo di un nuovo prodotto. Prima di procedere con la descrizione del prodotto offerto, è importante specificare la struttura del progetto stesso. In particolare, si tratta di un'unica commessa, composta da 2 sotto commesse: la prima si pone come obiettivo la consegna del **Minimum Viable Product** (MVP), la quale è stata svolta seguendo i principi della metodologia Waterfall; la seconda fase, invece, ha come fine ultimo quello di adattare continuamente l'offerta alle esigenze del cliente stesso. Questa seconda fase è svolta attraverso Sprint di due settimane, e dunque in modalità Agile. Con MVP si intende la prima versione del prodotto che serve a validare la soluzione. Viene utilizzato inizialmente solo da alcuni utenti identificati dal cliente, definiti anche Key User, che possono poi fornire feedback sul funzionamento del prodotto stesso. È buona prassi che l'MVP abbia poche features implementate; questo perché l'obiettivo dell'MVP è quello di capire se la soluzione suscita l'interesse e l'entusiasmo dei clienti, al punto da spingerli ad acquistare e consigliare il prodotto o il servizio progettato. Nel caso specifico di questa progettualità l'MVP è considerata la prima versione della piattaforma DIM mediante la quale potevano essere trattate solo alcune delle tipologie di Issues che il cliente riscontra in fase di sviluppo. Inoltre, è importante specificare che il processo implementato nell'MVP era leggermente diverso da quello implementato nei successivi sprint svolti in modalità Agile, soprattutto per quanto concerne le responsabilità dei diversi attori coinvolti.

3.1.1. Prima fase del progetto: consegna del Minimum Viable Product in modalità Waterfall

La prima fase, avente come obiettivo finale la consegna dell'MVP è stata gestita mediante un contratto Turn Key (TK), seguendo le fasi del ciclo di vita del software ovvero raccolta dei requisiti, implementazione, test e messa in produzione. In un contratto di questo tipo, l'appaltatore esegue l'intero lavoro di progettazione e implementazione e ha il compito di finanziare a breve termine il progetto, e dunque durante tutto il periodo di costruzione. Al momento della messa in produzione del prodotto finale, l'appaltatore deve effettuare il pagamento. In alcuni casi, tuttavia, è possibile negoziare un piccolo anticipo o una modalità di pagamento rateizzata, secondo la quale ogni tranche di pagamento è effettuata quando le milestone principali dell'ambito di lavoro vengono completate e ne viene verificata la conformità qualitativa. Questo metodo dà al committente il tempo per raccogliere fondi durante l'esecuzione del progetto stesso. La scelta di procedere in questa prima sotto commessa con la modalità Waterfall, trova le principali motivazioni di seguito. In questo modo:

- Si pongono le basi del progetto in quanto si prevede la creazione rapida di uno strumento funzionante in un ambiente di produzione;
- Si prova a ridurre il rischio associato al progetto, in quanto, con un MVP e non con una pianificazione a lungo termine, si riesce a comprendere subito il comportamento del prodotto, e si può ricalibrare di conseguenza, avendo una forma auto consistente su cui basare il feedback;
- Durante le prime settimane di utilizzo, ci si rende conto dell'evoluzione che si vuole privilegiare, gestita con un successivo approccio Agile.

Questa progettualità è nata intorno al mese di luglio, nel momento in cui è stata presentata la richiesta dal cliente, unitamente a un documento in cui erano elencati per grandi linee i requisiti richiesti ed era descritto il processo da gestire. Da quel momento in poi, ci sono stati diversi incontri con il cliente per riuscire a

comprendere a fondo i requisiti e il processo stesso. Successivamente, dopo uno studio di analisi di fattibilità da parte del team, è stato sviluppato e presentato al business un *Proof of Concept* (PoC) il quale aveva sia l'obiettivo di mostrare al cliente la fattibilità di quanto proposto sia l'obiettivo di mostrare come potesse essere il risultato finale. Dopo la presentazione del PoC, i rapporti tra il cliente e il committente sono continuati attraverso una serie di incontri in cui si andava ad analizzare le diverse proposte inerenti all'impostazione del progetto, indicando il rispettivo valore economico. Successivamente, con precisione il 17 ottobre 2022, è iniziata l'esecuzione del progetto a seguito di un **kick - off meeting** effettuato con il cliente stesso. Nello specifico, in questa riunione sono stati presentati in maniera definitiva gli obiettivi di progetto, gli approcci progettuali utilizzati, il team di progetto, l'architettura del sistema ed è stato anche presentato il Gantt effettuato per questa prima macro-fase progettuale. A questo punto sono iniziate le attività di sviluppo per portare a termine le funzionalità previste e, le figure funzionali all'interno del team hanno creato e formalizzato un documento chiamato *Analisi Funzionale*, in cui sono stati descritti diversi aspetti del progetto, quali:

- Lo scopo e gli obiettivi del progetto;
- La roadmap del progetto;
- Gli attori coinvolti e i permessi ad essi attribuiti;
- Il data model;
- La descrizione del processo gestito attraverso la piattaforma;
- Gli strumenti di reportistica presenti sul DIM.

Questo documento, una volta completato, è stato controllato ed approvato dal cliente, cosicché potesse essere utilizzato come riferimento principale nelle attività successive di sviluppo. Nello scheduling effettuato sono state previste cinque settimane per lo sviluppo vero e proprio della piattaforma, al termine delle quali è iniziata la fase di test. Oltre ai test effettuati dal team di progetto, a fine novembre circa, sono state effettuate tre sessioni di User Acceptance Test (UAT) in cui, oltre al

team di sviluppo, hanno partecipato anche dei Key User individuati dal cliente. In queste tre sessioni sono state mostrate tutte le funzionalità sviluppate del DIM, e gli utenti hanno dunque potuto vedere effettivamente per la prima volta la piattaforma. Da questi test, purtroppo è emerso che il processo su cui sono state fatte le valutazioni iniziali, e su cui si basava il documento di Analisi Funzionale approvato dal cliente, non era del tutto corretto e dunque a seguito di una richiesta effettuata dal cliente si è posticipato il Go Live previsto per fine novembre. Pertanto, è stato necessario svolgere nuovamente tutte le fasi previste dalla metodologia Waterfall, a partire dall'analisi funzionale, per andare poi alle attività di sviluppo ed effettuare, successivamente altre sessioni di test; l'impatto di questo evento in termini di tempi e costi sarà analizzato nel capitolo successivo. Sono state dunque programmate nuovamente delle sessioni di UAT per verificare insieme al cliente la correttezza del processo ed individuare eventuali bug del sistema. A valle della verifica e dei test effettuati sul sistema, è stata organizzata un'attività di training composta da due incontri durante i quali sono state elencate tutte le caratteristiche del sistema ed è stato mostrato un caso di utilizzo a tutti gli utenti individuati dal cliente. Alla fine di tutte le attività descritte in questo paragrafo è stato possibile procedere con il Go Live, avvenuto con ritardo di una settimana rispetto a quanto schedato inizialmente. Durante l'esecuzione della prima macro fase, si è mantenuto il rapporto con il cliente attraverso dei meeting settimanali della durata di un'ora circa, in cui il team di sviluppo aggiornava il business sullo Stato di Avanzamento dei Lavori (SAL).

3.1.2. Seconda fase del progetto: miglioramenti in Agile

Una volta terminata la fase di realizzazione dell'MVP, si ha una prima versione del prodotto stabile, veloce e utilizzato nell'ambiente di produzione, che può essere utilizzato dagli utenti, i quali hanno la possibilità di fornire feedback sul prodotto

stesso. Da questo momento in poi è possibile procedere con miglioramenti incrementali. È stato scelto questo tipo di approccio, in quanto:

- Gli Sprint brevi e iterativi permettono allo Scrum Team di lavorare alla ricerca del risultato migliore possibile e di apportare modifiche e/o aggiustamenti in ogni Sprint;
- Secondo le logiche dell'Agile, ai meeting partecipano gli utenti giusti creando così un ambiente stimolante, in grado di fornire sempre nuovi spunti;
- Vi è uno sviluppo continuo del prodotto, che migliora in modo incrementale.

All'inizio di questa sotto commessa è stata effettuata una riunione di pianificazione di tutte le cerimonie in cui sono state individuate anche le figure di Product Owner, fornito dal cliente, Scrum Master, facente parte dell'azienda Reply ed è stato definito nuovamente lo Scrum Team. Sono state inoltre definite le modalità di lavoro, ad esempio la durata degli Sprint e la modalità con cui tener traccia delle User Stories. A tal proposito, è stato individuato DevOps, uno strumento che permette la creazione e la gestione del Product Backlog, la creazione delle User Story e la definizione degli sprint. Permette di monitorare, inoltre, l'avanzamento di ogni attività. Di seguito è riportata un'immagine illustrativa dello strumento (*Figura 3.1*).

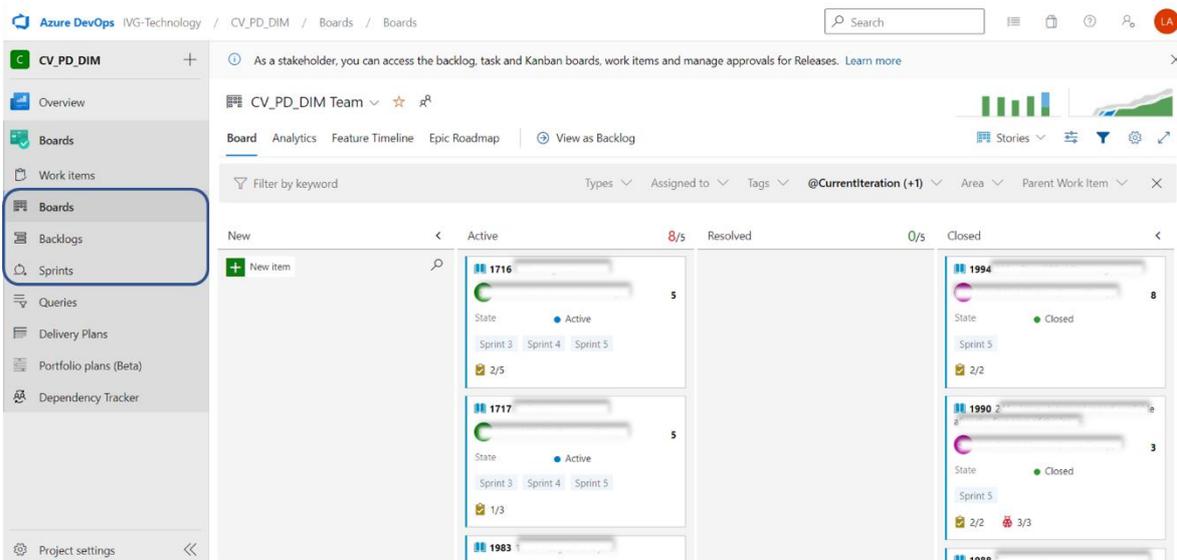


Figura 3.1 DevOps Azure

Questo strumento, inoltre, permette di creare tutti i grafici previsti dall'Agile e descritti nel *paragrafo 1.5.2.3.*, per tenere traccia dello stato di avanzamento del lavoro. Difatti, in base alle necessità è possibile creare una Dashboard in cui sono raffigurati BurnUp Chart, Burn Down Chart, Velocity Chart e altri ancora. All'inizio di ogni Sprint è effettuato lo Sprint Planning, una riunione in cui il Product Owner mostra i requisiti derivanti da feedback ricevuti dagli utenti, o da miglioramenti individuati dal business, che dovrebbero essere sviluppati in quel determinato Sprint. Ad ogni requisito, associa una narrazione sottoforma di User Story e, dopo aver analizzato tutto con lo Scrum Team, si assegna un punteggio in base alla complessità della richiesta. Dopo aver ripetuto questo procedimento per tutte le storie presenti nel Backlog selezionate per quello Sprint, saranno contati gli Story Points raggiunti e, nel caso in cui si dovesse sfiorare il limite stabilito, il Product Owner deciderà quali possono essere lasciate fuori e dunque, rimandate allo Sprint successivo. In questa riunione, si definisce anche un "Basket di riserva" di User Story da cui poter attingere nel caso in cui si riscontrassero delle problematiche durante lo svolgimento di un requisito, in modo che lo Scrum team possa continuare a lavorare. Durante le 2 settimane, sono sviluppati e testati i requisiti richiesti, aggiornando giornalmente il cliente attraverso il Daily Standup Meeting. Alla fine dello Sprint si effettua la riunione di Sprint Review, in cui lo Scrum Team presenta effettivamente al cliente tutti i requisiti che si sono riusciti a sviluppare. Successivamente, si procede con il rilascio del DIM aggiornato nell'ambiente di produzione. A valle del rilascio, viene effettuata la Sprint Retrospective, ovvero la cerimonia in cui il team riflette sull'andamento dello Sprint e identifica eventuali miglioramenti da apportare in futuro. Attualmente sono stati effettuati cinque Sprint.

3.2. Descrizione del processo di Development Issue Management

Il cliente, ha deciso di avviare questo progetto a seguito di alcune necessità. Nello specifico i bisogni principali sono quelli di:

- Disporre di uno strumento in grado di segnalare e gestire in modo semplice le problematiche che emergono durante le fasi di sviluppo di un nuovo prodotto;
- Fornire un supporto rapido attraverso l'implementazione dello strumento in modo da ottimizzare il reporting delle informazioni e garantire una visione complessiva dello stato di gestione delle problematiche;
- Stabilire una metodologia comune per la valutazione dei problemi in modo da facilitare l'identificazione delle priorità di intervento;
- Fornire una fonte unica di input per i risolutori e fornire supporto nella raccolta dei problemi.

A tal proposito, si è deciso di sviluppare la piattaforma DIM, ovvero un nuovo sistema basato su *Microsoft Power Platform* con cui gestire i processi necessari. In questo modo, i moduli di compilazione guidata e i flussi dei processi aziendali dedicati, consentono agli utenti di seguire tutte le stesse procedure. Inoltre lo strumento risulta essere altamente personalizzabile in base alle esigenze aziendali, e dunque può essere configurato per profilare diversi tipi di utenti, dando a ciascuno di essi la corretta visione di ciò che gli interessa. Infine vi è la possibilità di introdurre dashboard, viste o report dedicati e questo permette agli utenti di avere sempre una visione corretta e rapida dello stato delle anomalie. In particolare, nella *Figura 3.2* è riportato un esempio dei grafici riportati nella Dashboard customizzata per il cliente. Nello specifico i grafici raffigurati riportano le Issue attive divise per status in modo da monitorare lo step risolutivo in cui si trovano le Issue. D'altro canto, sulla destra è raffigurato un grafico a torta in cui le Issue sono divise per range di veicolo.

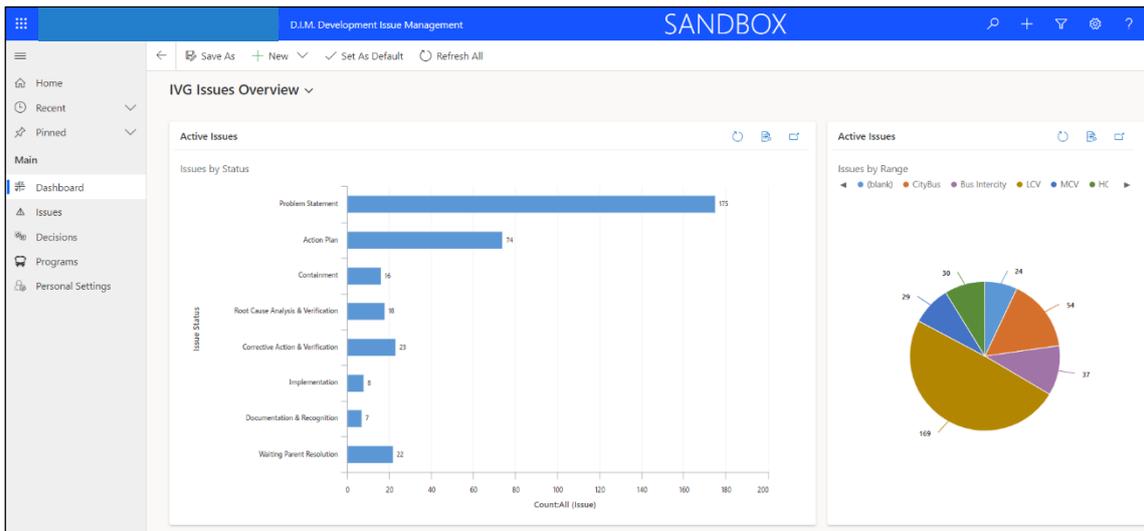


Figura 3.2 Dashboard DIM

Il sistema gestisce le fasi principali del processo di Development Issue Management: nella prima fase un utente, identificato come Issue Originator, registra la Issue nel sistema DIM e un altro utente, identificato come Issue Manager, controlla le informazioni inserite dall'Issue Originator. Può inoltre, selezionare un utente, definito Issue Solution Provider che si occuperà della fase di risoluzione. Successivamente si procede con la definizione dei membri del team; è l'Issue Solution Provider a definire i membri del team e il piano di lavoro per la fase di risoluzione del problema. Si arriva a questo punto alla fase di Problem Solving in cui l'Issue Solution Provider collabora con gli altri membri del team per identificare le cause principali e trovare sia un'azione di contenimento provvisoria, che un'azione correttiva permanente. Alla fine di questo processo, l'Issue Solution Provider inserisce le informazioni finali e chiude il problema. Attualmente, attraverso il DIM sono gestite 12 diverse tipologie di Issues. Di seguito è riportato un'immagine illustrativa del processo gestito con il DIM (Figura 3.3).

QUALITY PROCESSES

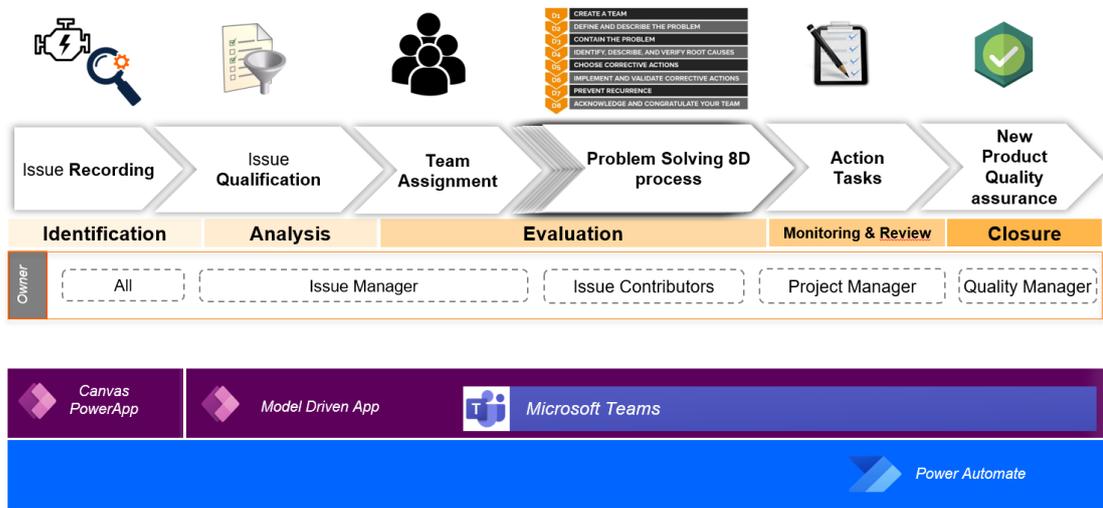


Figura 3.3 Processo DIM

Come si evince dalla figura sopra riportata, il cliente si serve dell'approccio 8D per la risoluzione delle problematiche, ovvero un modello a otto discipline (8D). Si tratta di un approccio tipicamente utilizzato dal reparto della qualità nell'industria automotive. Questa metodologia consente di trovare una soluzione a breve termine e una soluzione permanente ai problemi di prodotto o di processo. Lo scopo di questo approccio è quello di identificare, correggere ed eliminare i problemi ricorrenti. Gli step che caratterizzano questo processo sono ^[13]:

- D1 – Creazione di un team
- D2 – Descrizione del problema
- D3 – Definizione di un piano di contenimento del problema
- D4 – Definizione delle cause che hanno originato il problema
- D5 – Determinare azioni correttive permanenti
- D6 – Implementazione delle azioni correttive permanenti
- D7 – Prevenzione della ricorrenza del problema
- D8 – Congratulazioni al team

Attualmente, nel sistema DIM sono implementati tutti gli step tranne il settimo, che è oggetto dei prossimi Sprint; sarà difatti implementato un modulo per tutta la gestione dei rischi. È importante notare, che all'interno del DIM, i punti D1 e D2 sono invertiti. Nello specifico, avviene prima la descrizione del problema e poi la definizione di un team che sia adatto a risolvere quella specifica problematica. Difatti, come riportato in *Figura 3.4* il primo step gestito attraverso il DIM è il Problem Statement, seguito dalla fase di Action Plan in cui avviene la definizione del Team.



Figura 3.4 DIM steps

Di seguito sono riportate delle immagini illustrative della piattaforma DIM; difatti, nella *Figura 3.5*, è riportato un esempio di una Issue registrata a sistema.

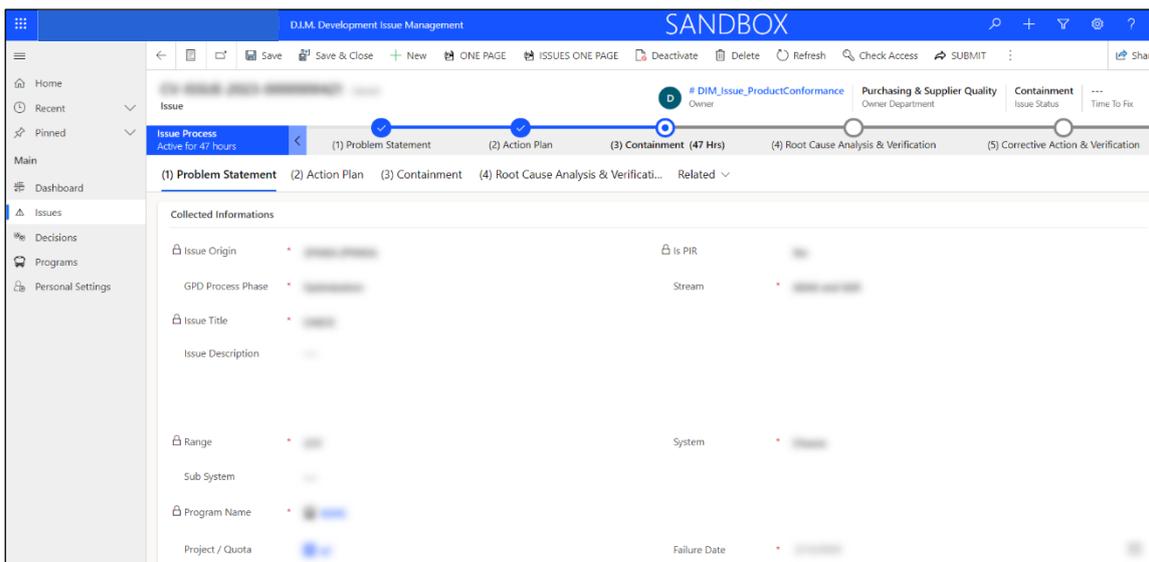


Figura 3.5 DIM Process Phase

3.3. Microsoft Power Platform

Come detto in precedenza, il DIM si basa su Microsoft Power Platform, essendo Microsoft partner di Cluster Reply. È pertanto utile fare un overview su che cos'è una Power Platform e delle sue funzionalità. Include 5 potenti applicazioni per creare applicazioni web e mobile, automazione dei work flow, chatbot, siti Web e BI per l'organizzazione e la gestione dei dati (Figura 3.6).

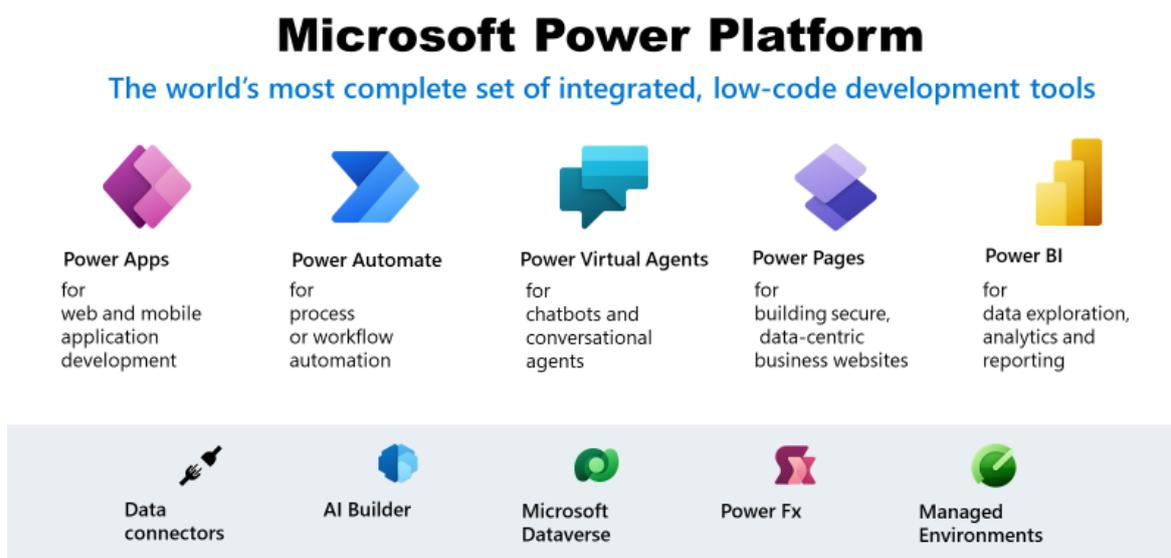


Figura 3.6 Microsoft Power Platform

In generale, le Power Apps consentono di creare rapidamente applicazioni, tramite modelli pre - esistenti messi a disposizione da Microsoft. Difatti, le Power Platform si basano su un approccio *Low Code - Zero Code* e anche per tale motivo risultano essere molto scalabili. Nello specifico, il modulo interesse è quello relativo alle Power Apps, essendo il DIM una **Model Driven App**. La creazione di una Model Driven App si focalizza sull'aggiunta di componenti come moduli, viste, grafici e dashboard, utilizzando uno strumento di progettazione di App. Grazie a questo approccio di sviluppo, vi è la possibilità di collegare tra loro le tabelle, attraverso delle relazioni, in modo da consentire la navigazione tra di esse e garantire che i dati non vengano ripetuti inutilmente. Utilizzando la metodologia *Low Code - No Code*, è possibile creare sia applicazioni semplici che applicazioni molto complesse. Le

Model Driven App sono particolarmente adatte per lo sviluppo di applicazioni relative a processi che si basano su innumerevoli dati rendendo facile per gli utenti spostarsi tra i record correlati. Proprio perché si tratta di applicazioni basate sui dati, è necessario che all'interno di *Microsoft Dataverse* vi sia un Data Model, altrimenti non è possibile creare una Model Driven App. Dal punto di vista della User Experience, tutte le app appartenenti a questa categoria offrono un'esperienza simile, semplice e accessibile a molti utenti anche da diverse tipologie di dispositivi (ad esempio mobile, desktop etc.). Tendenzialmente, nella schermata principale di una Model Driven App si possono inserire diversi tipi di componenti in base alle esigenze del cliente. Risulta essere quindi uno strumento altamente personalizzabile. Questo metodo di sviluppo delle applicazioni presenta alcuni vantaggi notevoli:

- Dopo aver creato il Data Model e le Relazioni esistenti tra le diverse entità, il processo di creazione della piattaforma è relativamente rapido, grazie alla ricchezza dei progettisti no-code incentrati sui componenti;
- Le applicazioni sono estremamente user – friendly grazie alla loro natura responsive per design. Ciò significa che presentano un'interfaccia utente simile su diversi tipi di dispositivi come ad esempio desktop o mobile;
- L'esperienza dell'utente è simile in tutte le Model Driven App. Una volta che l'utente ha imparato ad utilizzare un'applicazione Model-Driven, sarà più semplice adottare applicazioni successive all'interno della stessa organizzazione;
- La migrazione delle applicazioni tra ambienti di sviluppo, test e produzione è relativamente semplice grazie all'utilizzo delle solution, ovvero dei "pacchetti" che contengono tutto quello che è stato sviluppato;
- Questo tipo di applicazioni si integrano con Office 365 e per tale ragione sono familiari a chiunque abbia utilizzato le applicazioni di Dynamics 365. Le app di Dynamics 365, come Dynamics 365 Sales, Dynamics 365 Marketing o

Dynamics 365 Customer Service, sono essenzialmente un insieme di app line-of-business, costruite su Dataverse. L'utilità del modello di Dynamics 365 è stata riconosciuta come potente e, rendendo Dataverse completamente personalizzabile, ha permesso di creare più soluzioni su misura;

- Per quanto concerne l'aspetto della sicurezza, bisogna sottolineare che Dataverse supporta un livello di sicurezza dei dati forte e flessibile che influisce sul grado di interazione degli utenti con i record e le colonne delle tabelle.

I vantaggi sopra elencati trovano riscontro anche da un punto di vista quantitativo. Difatti, come è possibile notare nella *Figura 3.7*, grazie all'utilizzo delle Power Platform si riescono a ridurre i costi degli sviluppatori professionisti di circa il 24%. Inoltre, l'elevato grado di automazione consente una riduzione degli errori fino al 27% circa e una riduzione dello sforzo per la creazione e la manutenzione per un valore che si aggira intorno all'80%. Infine, secondo uno studio svolto da Gartner, il Net Present Value raggiunge il valore di 26 milioni di dollari.

Proven value in real-world business scenarios

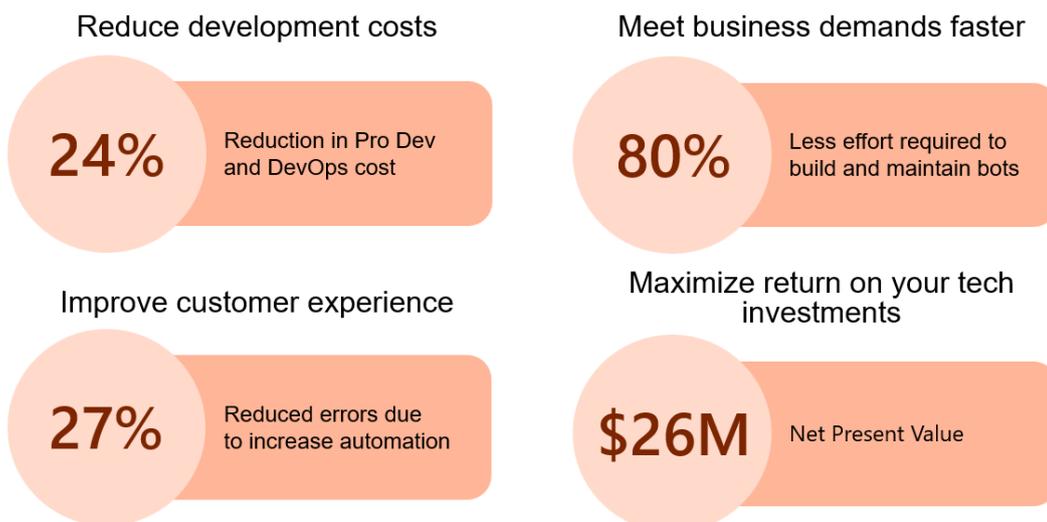


Figura 3.7 Microsoft Power Platform vantaggi

Il mercato inerente alle Low Code Application Platform si posiziona in uno scenario complessivo di tecnologie di sviluppo low-code, che dovrebbe raggiungere \$ 29 miliardi di entrate entro il 2025, con un tasso di crescita annuale composto (CAGR) di oltre il 20%. In particolare, si prevede che il segmento Low Code Application Platform raggiungerà i 14,38 miliardi di dollari nel 2025, con un CAGR del 26,4%^[14].

CAPITOLO 4: Analisi dei risultati

Lo scopo di questo capitolo è quello di fornire un confronto, applicato ad un caso reale, tra i risultati derivanti dalla gestione di una progettualità in modalità Waterfall rispetto alla modalità Agile.

4.1. Analisi dei costi

Il primo aspetto che sarà analizzato è quello relativo ai costi, che risulta essere uno dei punti cardine in una progettualità, sia lato fornitore che lato cliente. In questo caso, saranno analizzati i ricavi generati e i costi sostenuti dal fornitore Cluster Reply. Prima di procedere con l'analisi, bisogna fare un overview sui risultati che ci si prefissa di raggiungere come azienda. Nello specifico, come è stato detto in precedenza, Reply è organizzata in company verticali "autonome" nella gestione delle varie commesse e, ogni company gestisce più progettualità. A livello aziendale è preso in considerazione come indicatore su cui valutare la performance economica, l'indice $\frac{C}{R} = \frac{\text{Costi}}{\text{Ricavi}}$. Per il calcolo di questo indicatore sono presi in considerazione i **costi diretti** allocati alla commessa e i **ricavi** che la stessa genera. In linea teorica l'indice:

- $\frac{C}{R} < 1$ indica che la progettualità sta generando un margine di guadagno per l'azienda;
- $\frac{C}{R} = 1$ indica che la progettualità non sta generando margine per l'azienda;
- $\frac{C}{R} > 1$ indica che la progettualità sta generando perdite per l'azienda, in quanto i ricavi risultano essere minori dei costi diretti associati alla commessa.

Poiché i costi considerati in questo indice sono solo quelli diretti allocati al progetto, Reply chiede alle varie Company di svolgere progettualità che generano un $\frac{C}{R} < 1$.

È importante sottolineare che i costi fissi, come ad esempio i costi delle infrastrutture, dei computer, etc., saranno poi allocati dal gruppo Reply ad un livello più alto. Per tale ragione minore sarà il valore di $\frac{C}{R}$ che si riuscirà ad ottenere, maggiore sarà il margine di contribuzione per l'azienda. A valle di questa panoramica, si può procedere con l'analisi del Conto Economico relativo alle due sotto commesse. Nella *Figura 4.1* è riportato il conto economico relativo alla consegna dell'MVP, realizzato in modalità Waterfall. I valori dei ricavi e dei costi sono puramente indicativi in quanto non possono essere resi noti quelli effettivi. Ad ogni modo, il confronto tra i risultati ottenuti nella prima sotto commessa, e quelli ottenuti nella seconda, **rispecchia quanto accaduto nella realtà**.

	Budget (€)	Ricavi (€)	Costi (€)	Margine (€)	C/R 2022
Lavoro	60000	60000	48600	11400	0,81
T&M					
T&M Rettifiche					
Turn Key	60000 9070 €/mu	60000	48600	11400	0,81
Canone					
R&S					
Lavoro Indiretto	0	0	0	0	
Subappalto	0	0	0	0	
Rivendita					
Varie (Fee)		0	0	0	
Spese	0	0	0	0	
Note Spese (Timesh.)		0	0	0	
Ind. Trasf. (Timesh.)		0	0	0	
Altre	0	0	0	0	
Totale	60000	60000	48600	11400	0,81

Rettifiche		0	0	0	
Totale	60000	60000	48600	11400	0,81

Figura 4.1 Conto Economico – MVP

Come è stato detto in precedenza, il contratto relativo alla prima fase progettuale è un Turn Key a prezzo fisso. Come si evince dal Conto Economico, i ricavi ipotizzati che derivano da questa commessa sono pari alla cifra di 60 k€, mentre i costi del personale che Cluster Reply ha sostenuto sono stati all'incirca 48,6 k€, generando così un margine di circa 11,4 k€ e un valore dell'indice $\frac{C}{R} < 1$. Questa sotto commessa, dunque, si è rivelata essere profittevole per l'azienda. Bisogna però evidenziare un fattore molto importante: qualora dovessero verificarsi ritardi o dovessero esserci giornate lavorative in aggiunta a quelle programmate, essendo un contratto a prezzo fisso, si andrà a bruciare parte del margine del fornitore, non potendo aumentare il prezzo al cliente. Questo è sicuramente un **rischio** che, in fase contrattuale e durante la progettualità stessa, deve essere tenuto in considerazione. Difatti, durante lo svolgimento di questo progetto, tale rischio si è manifestato seppur in maniera lieve. Nello specifico, come è stato riportato nel *paragrafo 3.1.1*, durante le sessioni di User Acceptance Test è stata rilevata una problematica che ha comportato una settimana di ritardo nella consegna del Minimum Viable Product. In base a quanto detto in precedenza, dunque, il margine dell'azienda Cluster Reply avrebbe potuto essere superiore di un valore pari al numero delle risorse impiegate nei giorni non previsti per il valore giornaliero della risorsa stessa per i giorni non inizialmente schedulati. Pertanto, un'altra motivazione a supporto del fatto che Reply richieda il valore dell'indice $\frac{C}{R} < 1$ è proprio la mitigazione di questo rischio, il quale ha una probabilità di accadimento medio – alta se la progettualità è svolta in modalità Waterfall. Nella *Figura 4.2*, sono riportati i costi diretti associati alle varie risorse che hanno lavorato durante la prima sotto commessa. Da tale figura, si evince che le risorse effettivamente allocate alla progettualità risultano essere 6. Ognuna di esse, in base al proprio ruolo e alla propria esperienza risulta essere più o meno coinvolta nel progetto, e risulta avere un costo differente. Inoltre, è possibile notare come, oltre ai costi diretti, sono considerati anche quelli del lavoro indiretto e le note spese, che per questa fase del progetto risultano essere nulli.

AP	GEN	FEB	MAR	APR	MAG	GIU	LUG	AGO	SET	OTT	NOV	DIC	AS	
Lavoro	0	0	0	0	0	0	0	0	0	0	7.987	8.589	0	0
RISORSA 1	0	0	0	0	0	0	0	0	0	1.797	3.521	771	0	0
gg										-9,31	-18,24	-3,99		
€/g										-193	-193	-193		
RISORSA 2	0	0	0	0	0	0	0	0	0	4394,3	8732	0	0	0
gg										-9,31	-18,50			
€/g										-472	-472			
RISORSA 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gg														
€/g														
RISORSA 4	0	0	0	0	0	0	0	0	0	2206,6	4444	1606	0	0
gg										-7,48	-15,06	-5,32		
€/g										-295	-295	-302		
RISORSA 5	0	0	0	0	0	0	0	0	0	2062	4038	1036	0	0
gg										-9,962	-19,51	-5,00		
€/g										-207	-207	-207		
RISORSA 6	0	0	0	0	0	0	0	0	0	1620	2710	1935	0	0
gg										-10,45	-17,48	-12,48		
€/g										-155	-155	-155		
RISORSA 7	0	0	0	0	0	0	0	0	0	1740	2235	3750	0	0
gg										-9,72	-12,49	-20,95		
€/g										-179	-179	-179		
Lavoro Indiretto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Subappalto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rivendita	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Altro	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Spese	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note Spese	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nob 01 - Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nob 06 - Budget Residuo	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Indennità	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nob 01 - Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nob 06 - Budget Residuo	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Altre	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Totale	0	0	0	0	0	0	0	0	0	13.820	25.680	9.098	0	0
Rettifiche	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Totale	0	0	0	0	0	0	0	0	0	13.820	25.680	9.098	0	0

Figura 4.2 Costi diretti - MVP

In *Figura 4.3*, invece, è riportato il Conto Economico relativo alla seconda fase del progetto svolta in modalità Agile, la quale risulta essere ancora in corso.

	Budget (€)	Ricavi (€)	Costi (€)	Margine (€)	C/R 2022
Lavoro	301000	274900	192400	82500	0,70
T&M					
T&M Rettifiche					
Turn Key	301000 9095 €/mu	274900	192400	82500	0,70
Canone					
R&S					
Lavoro Indiretto	0	0	0	0	
Subappalto	0	0	0	0	
Rivendita					
Varie (Fee)		0	0	0	
Spese	0	0	0	0	
Note Spese (Timesh.)		0	0	0	
Ind. Trasf. (Timesh.)		0	0	0	
Altre	0	0	0	0	
Totale	60000	274900	192400	82500	0,70

Rettifiche		0	0	0	
Totale	60000	274900	192400	82500	0,70

Figura 4.3 Conto Economico – Agile

In questo caso, il prezzo della sotto commessa non risulta essere fisso. Per tale motivo, il rischio descritto in precedenza è completamente a carico del cliente. Difatti, se dovessero verificarsi problemi e fossero necessarie ulteriori giornate lavorative o Sprint aggiuntivi rispetto a quelli inizialmente concordati, il prezzo che il committente pagherà, sarà maggiorato di un valore pari alle giornate extra che dovranno essere effettuate. Al momento, i ricavi sono circa 274,9 k€ e i costi sostenuti sono stati di 192,4 k€, generando così un margine di 82,5 k€. Pertanto, anche nella seconda fase progettuale si rileva un indice $\frac{C}{R} < 1$.

Difatti attualmente si osserva l'indice $\frac{C}{R}$ inferiore a quello analizzato nel caso precedente. Pertanto la situazione per quanto concerne i costi risulta essere migliore in questa seconda sotto commessa. Difatti, rispetto all'indice $\frac{C}{R}$ ottenuto nella prima sotto commessa che risultava essere pari a 0,81, in questo caso è stato ottenuto un valore dello stesso indice pari a 0,7. Si rileva dunque, una riduzione di circa il 13,75% dei costi sostenuti nella seconda fase progettuale, svolta in modalità Agile, rispetto a quelli sostenuti nella prima parte progettuale, svolta in modalità Waterfall. Per quanto concerne il margine, invece, si può notare un aumento percentuale tra le due fasi di circa il 57,9% in quanto nella prima fase progettuale il margine risultava essere di circa il 19% dei ricavi, mentre nella seconda risultava essere del 30% circa. Questi risultati sono dovuti molto probabilmente al fatto che il primo contratto, per la consegna dell'MVP fosse un Turn Key a prezzo fisso e, a seguito dell'inconveniente verificatosi durante lo svolgimento degli User Acceptance Test della fase di MVP, i costi sostenuti dall'azienda Reply sono aumentati a sfavore del margine il quale è, ovviamente, diminuito.

Nella *Figura 4.4*, sono riportati i costi diretti delle risorse associati a questa fase progettuale svolta in modalità Agile.

	AP	GEN	FEB	MAR	APR	MAG	GIU	LUG	AGO	SET	OTT	NOV	DIC	AS
Lavoro	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 1	3909	3872	4042	0	0	0	0	0	0	0	0	0	0	0
gg	-8,28	-8,20	-8,56											
€/g	-472	-472	-472											
RISORSA 2	2436	2834	3280	0	0	0	0	0	0	0	0	0	0	0
gg	-12,62	-14,68	-16,99											
€/g	-193	-193	-193											
RISORSA 3	5581	4754	4793	0	0	0	0	0	0	0	0	0	0	0
gg	-18,92	-16,12	-16,25											
€/g	-295	-295	-295											
Nob 06 - Budget Residuo	0	0	0	15.250	15.250	15.250	15.250	15.250	15.250	15.250	15.250	9.751	0	0
RISORSA 4	1346	0	0	0	0	0	0	0	0	0	0	0	0	0
gg	-6,5													
€/g	-207													
RISORSA 5	3574	3395	3217	0	0	0	0	0	0	0	0	0	0	0
gg	-19,97	-18,97	-17,97											
€/g	-179	-179	-179											
Nob 01 - Buffer	0	0	0	0	0	0	0	0	0	0	0	0	10.750	0
RISORSA 6	0	0	2866	0	0	0	0	0	0	0	0	0	0	0
gg			-6,07											
€/g			-472											
RISORSA 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gg														
€/g														
Lavoro Indiretto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Subappalto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rivendita	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Altro	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Spese	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note Spese	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nob 01 - Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nob 06 - Budget Residuo	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Indennità	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RISORSA 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nob 01 - Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nob 06 - Budget Residuo	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Altre	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Totale	16846	14855	18198	15250	15250	15250	15250	15250	15250	15250	15250	9751	10750	0
Rettifiche	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Totale	16846	14855	18198	15250	15250	15250	15250	15250	15250	15250	15250	9751	10750	0

Figura 4.4 Costi diretti - Agile

4.2. Analisi dei tempi

Per quanto riguarda la gestione delle tempistiche bisogna sottolineare che nella prima fase del progetto, inerente alla consegna del Minimum Viable Product, è stata effettuata una schedulazione di dettaglio sin dalle prime fasi del progetto. Inizialmente, in fase di offerta, è stato fornito al cliente un piano di alto livello, che aveva il fine di riportare le scadenze e degli eventi più importanti che si sarebbero verificati durante la fase di esecuzione del progetto. Una volta approvato dal cliente stesso, il team di progetto ha provveduto ad effettuare una schedulazione dettagliata dei vari eventi, fissando così tutte le scadenze progettuali. In questo Gantt, effettuato con Microsoft Project e riportato in *Figura 4.3*, è possibile notare come sono state considerate sei macro-attività da svolgere.

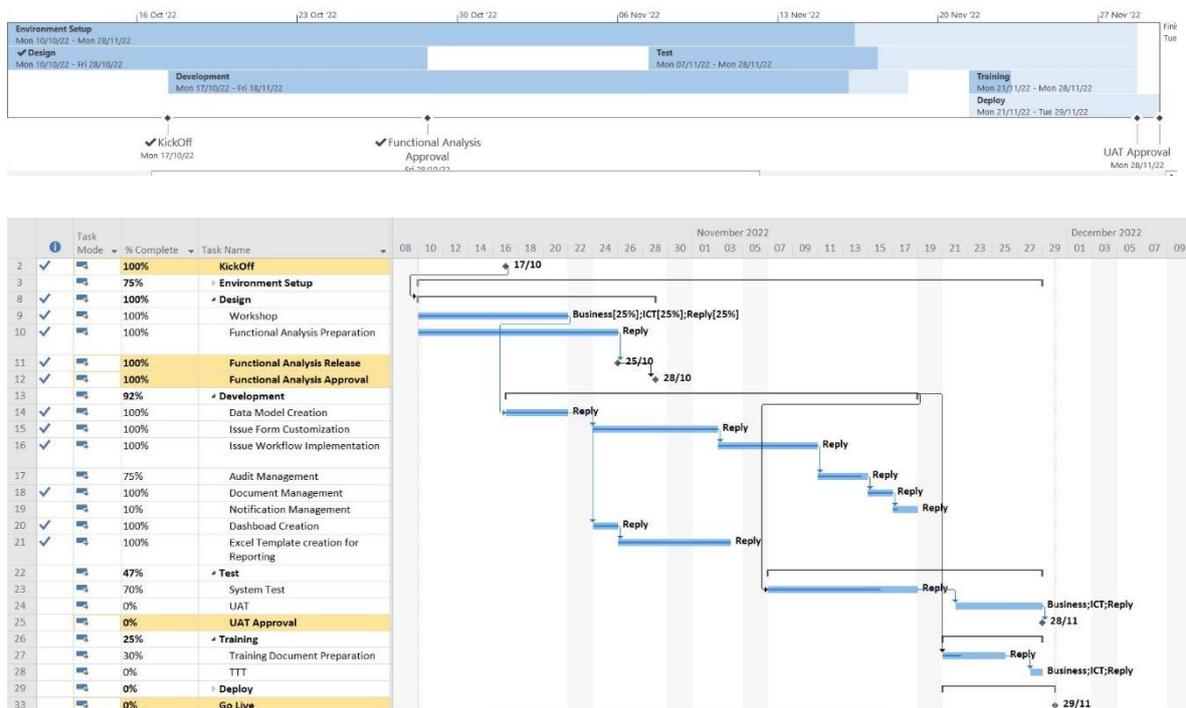


Figura 4.5 Gantt di progetto

Le macro-attività previste sono:

- Environment Setup: questo primo step è effettuato all'inizio di ogni nuova progettualità e permette agli sviluppatori la configurazione dell'ambiente in cui poi si svolgeranno le attività di sviluppo. La durata prevista di questa fase risulta essere di una settimana per quanto riguarda la configurazione dell'ambiente di sviluppo, effettuata prima dell'inizio del progetto, e poi di altre due settimane rispettivamente per gli ambienti di QA e per quello di produzione.
- Design: in questo secondo step vi è l'inizio delle attività di design, in cui la parte funzionale del team redige l'Analisi Funzionale, al termine della quale sarà necessaria l'approvazione del cliente. Per questa fase progettuale sono state considerate tre settimane di lavoro.
- Development: come si può notare dal Gantt di progetto le attività di sviluppo sono state svolte in parallelo alla fase precedente soltanto per due settimane, in quanto, essendoci la necessità di avere un MVP in tempi relativamente brevi, si è ritenuto opportuna questa parallelizzazione in modo da terminare la prima fase in tempi più brevi. La durata schedulata per questa fase, che risulta essere la più importante del progetto è di cinque settimane.
- UAT: al termine della fase di development, sono stati svolti gli User Acceptance Test con i key user identificati dal cliente, per testare il corretto funzionamento del DIM. Per questa attività è stata inizialmente prevista una sola settimana.
- Training: in fase di offerta è stata prevista la formazione degli utenti per l'utilizzo della piattaforma. In particolare, sono state concordate delle sessioni di "Train the trainer" in cui il team di progetto aveva l'obiettivo di formare dei key user i quali, successivamente avrebbero dovuto estendere la formazione a tutti i colleghi. A supporto di questa attività è stato redatto un manuale in formato Power Point, contenente tutte le funzionalità supportate dal DIM.

- Deploy: in questa fase la solution precedentemente sviluppata, è riportata dapprima negli ambienti di QA, per verificarne il corretto funzionamento prima del Go Live, in cui la solution viene riportata nell'ambiente di produzione.

Le tempistiche sopra elencate sono quelle inizialmente schedulate, prima della fase di sviluppo. Durante l'esecuzione del progetto però, sono state leggermente modificate durante le fasi finali del progetto, proprio perché, come spiegato nei paragrafi precedenti, si è manifestato il rischio di rilevare “non conformità” durante lo svolgimento degli UAT. L'impatto di tale evento a livello delle tempistiche è stato il ritardo di una settimana rispetto a quanto inizialmente schedulato.

Al contrario, nella seconda sotto commessa del progetto non è stato sviluppato nessun Gantt proprio per la natura ciclica dell'Agile, ma si è andati a valutare in ogni sprint, il peso delle varie User Stories richieste. Per definire gli **story points** relativi alle user stories ci sono diverse tecniche che possono essere utilizzate. Difatti, devono essere tenuti in considerazione diversi aspetti per assegnare il giusto punteggio a una storia. Principalmente si devono tenere in considerazione tre aspetti fondamentali:

1. La quantità di lavoro da effettuare per concludere quella user story;
2. La complessità del lavoro richiesto;
3. Qualsiasi rischio o incertezza legata al lavoro stesso.

Nello specifico, durante questa progettualità, avendo sviluppato il Minimum Viable Product prima dello svolgimento della fase Agile, il team era già consapevole dello sforzo necessario per sviluppare le nuove features richieste dal cliente. Pertanto, per stimare le user stories presenti nel Backlog del primo Sprint ci si è basati soprattutto sull'**esperienza** legata alla progettualità, essendo le features richieste note al team in termini di effort necessario per svilupparle. L'esperienza e le statistiche precedenti da cui imparare, sono sicuramente uno dei modi più efficaci per stimare

le storie nella maniera più corretta possibile. Per quanto concerne, invece, le User Stories del tutto nuove, per cui il team non aveva un'esperienza simile in termini di quantificazione del lavoro richiesto, si è deciso di utilizzare la nota **tecnica di gamification**, anche conosciuta come **Planning Poker**. Si tratta di una tecnica che pone le sue basi sulla sequenza di Fibonacci; pertanto, come punteggi possibili da assegnare alle varie storie sono presi in considerazione solo quelli presenti nella sequenza (1,2,3,5,8 etc.), e non tutti i numeri in sequenza lineare crescente (1,2,3,4 etc.). Questo perché, essendo i numeri della serie di Fibonacci distanziati, sarà molto più facile e veloce scegliere se assegnare a una storia il punteggio 5 anziché 8, rispetto alla scelta che ci sarebbe da fare tra un 5 e un 6. In ogni caso, quando il team si è trovato di fronte a nuove funzionalità da stimare si procedeva in questo modo: durante le riunioni di Sprint Planning, dopo aver discusso e presentato la user story, ogni membro del team, assegnava un punteggio in maniera anonima a quella storia. Qualora il valore assegnato fosse stato uguale a quello degli altri, allora sarebbe stato automaticamente assegnato quel punteggio; altrimenti ogni membro del team era chiamato a spiegare il motivo del punteggio assegnato, per poi ripetere di nuovo tutto il processo finché non si trovava un accordo. Gli story point si rivelano fondamentali nella comprensione di cosa si riesce a sviluppare in un determinato sprint perché, grazie a una corretta stima, si riesce a determinare la **velocity** del team. Difatti, man mano che si avanza con il numero Sprint, l'esperienza del team aumenta e, per stimare la capacity dei nuovi Sprint, ci si basa sempre di più su quello che si è riusciti a sviluppare negli sprint precedenti. In questo caso per valutare i tempi, ci si può ricondurre ai grafici previsti dalla teoria, e in particolare al Velocity Chart, BurnUp Chart, Burn Down Chart e Cumulative Flow Diagram raffigurati rispettivamente nelle *Figure 4.6, 4.7 e 4.8*. È importante specificare che in ogni Sprint, sono presi in considerazione e successivamente sviluppati circa 40 story point.

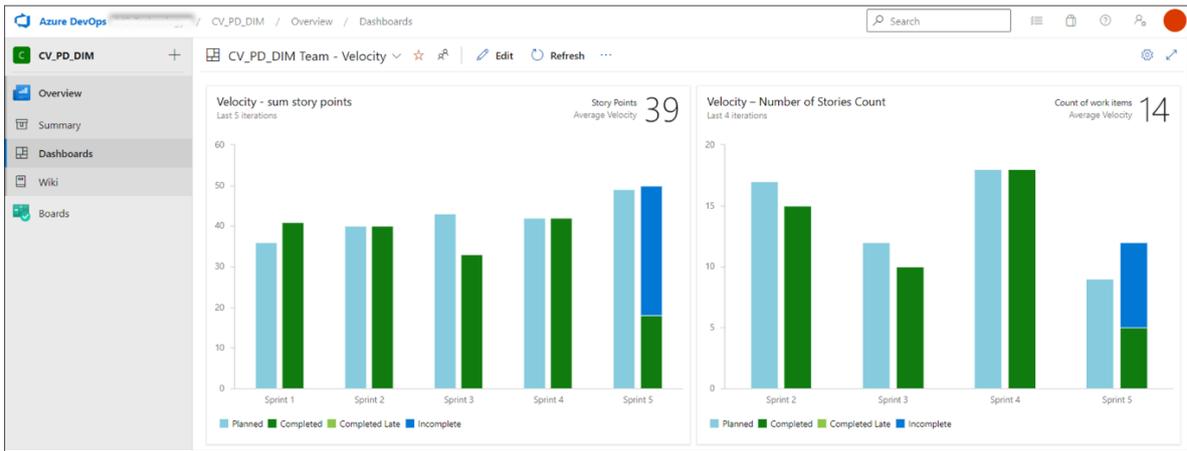


Figura 4.6 Velocity chart

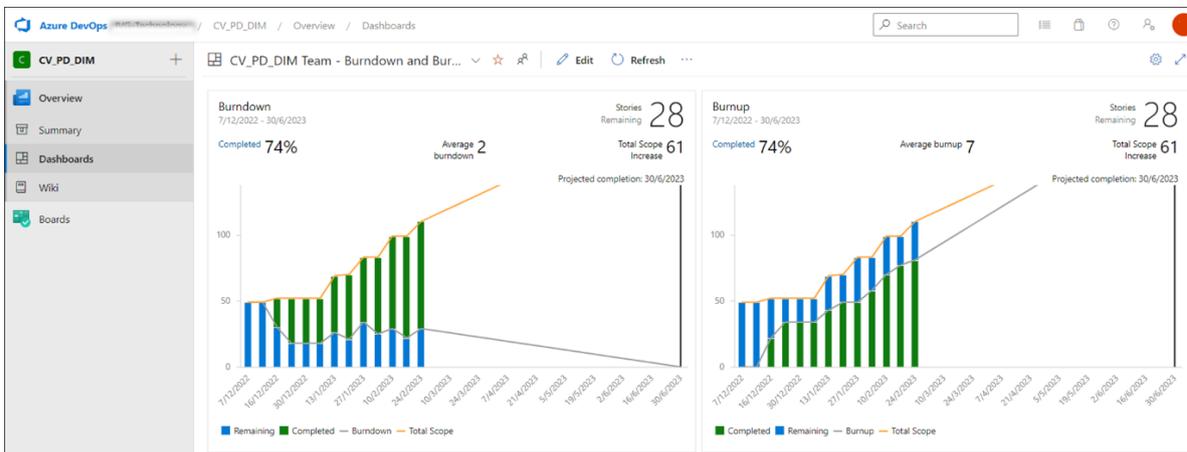


Figura 4.7 Burdndown chart e Burnup chart

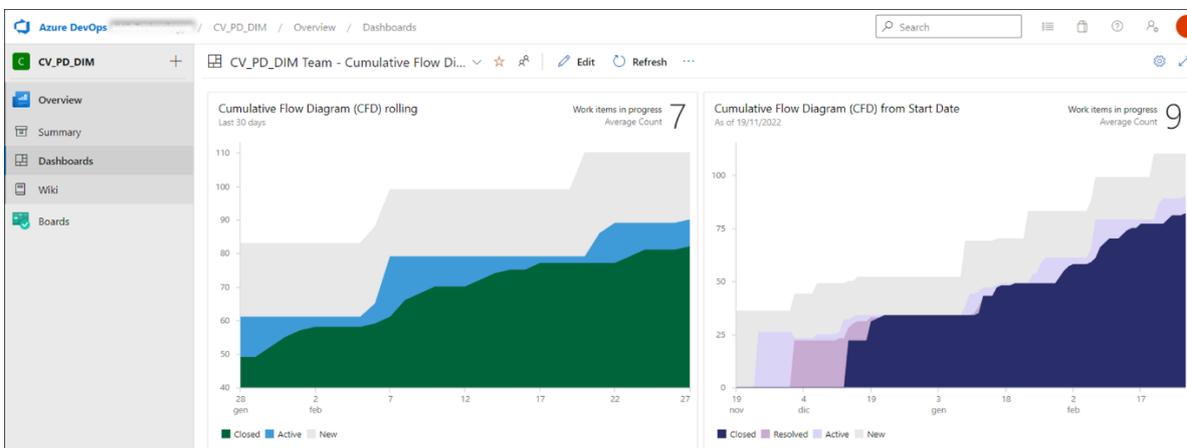


Figura 4.8 Cumulative flow diagram

Dal velocity chart basato sugli story points, si evince in primo luogo come durante l'esecuzione dello Sprint 1, sono stati completati un **maggior numero** di story points rispetto a quelli che erano stati inizialmente pianificati. Questo perché il team di sviluppo, mentre era in attesa di informazioni chiave da parte del business del cliente, ha deciso di non fermarsi e dunque ha deciso di implementare delle features che erano state messe nel "basket" di storie da cui attingere in caso di problemi, di cui si è discusso nel *Paragrafo 3.1.2*. Un altro evento interessante si è verificato nel terzo sprint. In particolare, si vede come gli story points conclusi al termine dello Sprint sono risultati essere **inferiori** rispetto a quelli inizialmente pianificati. Questo perché le storie a cui il Product Owner aveva dato una priorità più alta per quel determinato Sprint, erano principalmente analisi di possibili integrazioni con altri sistemi. Le user story, però, risultavano essere poco chiare e dettagliate, e pertanto è stato assegnato un punteggio troppo basso rispetto all'effort effettivamente richiesto. Inoltre, non essendo troppo specifiche, erano necessarie delle informazioni sia dal business del cliente, sia dagli altri sistemi per cui doveva essere analizzata l'integrazione, che purtroppo però non sono arrivate in tempo per poter concludere le storie nei tempi richiesti e concordati. Dai grafici di BurnUp e BurnDown, invece, si evince come il Product Backlog sia in continua evoluzione, dato che in entrambi i grafici il Total Scope aumenta. Inoltre, risultano essere perfettamente in linea con quanto descritto da un punto di vista teorico nel *Paragrafo 1.5.2.3*. In particolare, il trend del BurnDown chart risulta essere decrescente, in quanto indica il lavoro che manca per raggiungere l'obiettivo. Al contrario, il trend del BurnUp chart risulta essere crescente, poiché tale linea sta ad indicare il lavoro effettivamente svolto.

4.3. Analisi della qualità

Un altro aspetto da tenere sicuramente in considerazione per confrontare le due modalità, è quello della qualità. L'azienda Reply, ha due accezioni del termine qualità. In primo luogo, è considerata da un punto di vista di delivery e dunque, si pone molta attenzione al concetto di qualità del software offerto. Quest'ultima è monitorata attraverso il numero di bug rilevati dal cliente dopo la fase di Go Live. È interessante notare come, in questa progettualità non sono ancora stati rilevati bug, probabilmente perché il progetto è ancora in corso e, con cadenza di due settimane, viene migliorato con sviluppi incrementali. Difatti, durante le iterazioni Agile, qualora dovesse essere rilevato un problema o una funzionalità non del tutto corretta, durante l'iterazione successiva sarà modificato in base ai feedback ricevuti portando dunque un'evoluzione continua alle funzionalità del software. Per tale ragione, durante lo svolgimento della seconda sotto commessa in modalità Agile, il risultato degli sprint, e dunque le funzionalità del DIM, aderisce molto di più alle esigenze del cliente, il quale risulta essere molto più soddisfatto anche da un punto qualitativo del prodotto. Oltre a questo aspetto, l'azienda Reply richiede una gestione della qualità da un punto di vista di organizzazione e documentazione interna. Nello specifico, deve essere compilato un documento, denominato *Scheda Piano Commessa*, in cui si tiene traccia di diversi aspetti legati a quella progettualità, come ad esempio le approvazioni, gli output, gli input e tutti gli eventi salienti. Il documento è strutturato in diverse sezioni, in particolare:

- Riferimenti Commessa: in questa sezione sono indicati i riferimenti contrattuali, l'organizzazione della commessa in termini di comitato guida. Difatti, sono individuate due figure, una che rappresenti il committente e una che rappresenti il fornitore le quali compongono il **comitato guida**, e ne sono indicate le responsabilità di progetto. Infine, sono indicate le responsabilità progettuali sia lato committente che lato cliente.

- General Data Protection Regulation (GDPR): si tratta di un regolamento complesso attraverso il quale si cerca di regolare la raccolta, l'utilizzo e la gestione dei dati personali da parte di aziende che offrono beni e servizi a persone residenti nell'UE.
- Marketing: in questa sezione sono indicate tutte le azioni effettuate per far conoscere il progetto al resto del gruppo aziendale. In particolare, esiste una piattaforma interna a Reply in cui, ogni qual volta nasce un nuovo progetto si pubblicano informazioni generali sul progetto stesso e sul cliente.
- NDA: questa parte del documento, serve ad indicare se l'azienda cliente e il gruppo Reply sono soggetti a Non Disclosure Agreement, ovvero l'accordo di riservatezza mediante il quale, le parti si impegnano a non divulgare informazioni e dati sensibili relative al progetto.
- Project Journaling: si tratta di una sezione in cui si tiene traccia di tutte le milestone principali del progetto. In particolare si riporta la data, l'evento, gli item trattati ed eventuali allegati. Un esempio potrebbe essere la riunione di kick – off, inserendo come allegato la presentazione utilizzata per tale meeting; o ancora un altro esempio potrebbe essere la consegna dell'analisi funzionale da parte del team, allegando la versione definitiva della stessa e così via per le altre attività.
- Project Scheduling: in questa sezione sono riportate le diverse fasi del progetto, definendo gli input e gli output previsti. Si tiene traccia, altresì, delle date previste e di quelle effettive in cui si verificano gli eventi, e dei responsabili lato fornitore e lato cliente.
- Riesami di avanzamento progetto: si tratta di una sezione in cui si riportano gli eventi ciclici, che si ripetono con una determinata cadenza. In particolare, nel caso specifico di questa progettualità sono stati riportati i SAL effettuati settimanalmente con il cliente. In questo caso, si tiene traccia della data prevista, di quella effettiva, dei partecipanti all'evento, delle attività

riesaminate, dell'esito, di eventuali criticità evidenziate e delle azioni intraprese.

- Validazione finale del progetto post Go Live – periodo di roll out: in questa sezione è indicata la data di Go Live, quindi quella relativa al rilascio in produzione, e si indica come sarà gestito il periodo successivo a questa fase. Sarà, inoltre, indicata una statistica dei problemi rilevati nel software dopo il rilascio.

Questo documento è stato redatto sia per la prima sotto commessa che per la seconda, svolta in modalità Agile. La differenza risiede nella sezione del Project Scheduling in quanto, per la parte Waterfall vi era un piano ben definito e la suddivisione tra le diverse fasi che ha permesso la compilazione del documento a priori indicando le date previste e i responsabili; per quanto concerne la seconda sotto commessa, invece, non sono state indicate le diverse fasi poiché non previste, ma sono stati indicati gli Sprint effettuati.

Quando si parla di qualità è importante, inoltre, considerare la risposta del cliente. Difatti, come KPI è stato preso in considerazione il numero di nuovi utenti del DIM per ogni settimana, da quando è stato rilasciato l'MVP, ad oggi. Come si può notare nel grafico a barre riportato nella *Figura 4.9*, il numero di utenti è crescente.

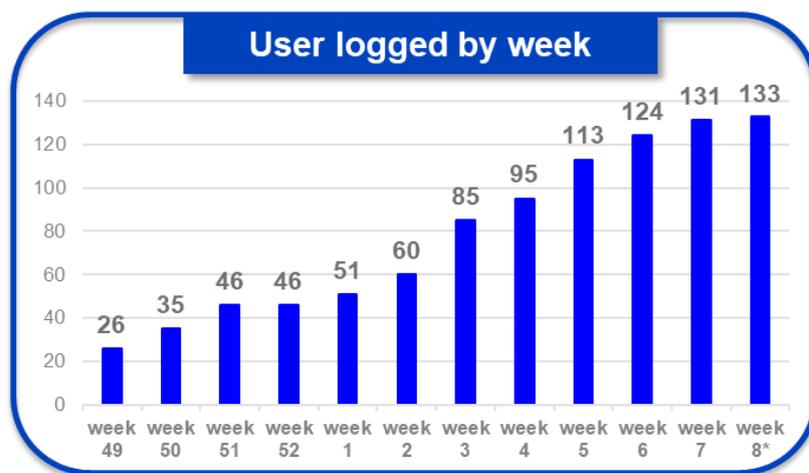


Figura 4.9 KPI - nuovi utenti per settimana

Difatti, dopo il rilascio del Minimum Viable Product, gli utenti che effettivamente utilizzavano la piattaforma DIM erano solamente 26. Questo perché le funzionalità inizialmente previste, e dunque quelle presenti sulla piattaforma risultavano essere limitate. Inoltre, non si è riscontrato un coinvolgimento degli user in questa prima fase e, per tale ragione, la risposta degli utenti non è stata particolarmente positiva. Si può notare però come con il tempo, e dunque con l'avanzamento degli Sprint e lo sviluppo di nuove funzionalità proposte dagli user, il numero di utenti che utilizza il sistema, è in continuo aumento. Si può notare come, durante la settimana 8, solo il 27% degli utenti, ovvero 133 utenti su 491 abilitati, hanno effettuato almeno il primo accesso e hanno effettivamente iniziato ad utilizzare la piattaforma. È dunque evidente come durante la sotto commessa svolta in modalità Agile, il numero di user ha un trend crescente. In questa fase, inoltre, si è riscontrato un maggior coinvolgimento del business del cliente, il quale può riscontrare in maniera pratica lo sviluppo delle funzionalità richieste dagli utenti stessi dopo poche settimane nella piattaforma DIM e riesce ad intervenire in tempi brevi qualora dovesse essere necessario effettuare dei cambiamenti. Queste motivazioni, dunque, comportano una maggiore **user adoption** nella seconda fase del progetto.

4.4 Analisi delle risorse

Come detto in precedenza questa progettualità è partita ad ottobre ed è ancora in corso. Fino a questo momento, il team che ha lavorato durante la fase svolta in Waterfall è lo stesso che sta lavorando sulla seconda sotto commessa in Agile. Pertanto, in questo caso specifico non si evidenziano particolari differenze da un punto di vista del **numero di risorse**; questo potrebbe essere dovuto al fatto che il progetto risulta essere ancora in corso e, finora, non si sono riscontrate problematiche particolarmente importanti. Andando però ad analizzare questo fattore in linea teorica, si riscontra che in Agile si predilige un team meno numeroso

e più stabile rispetto alla modalità Waterfall, in quanto in quest'ultima modalità il team ha il compito di assorbire i picchi di imprevisti e di anomalie. Difatti, se la data di Go Live risulta essere fissa e non può essere posticipata, in modalità Waterfall si deve inevitabilmente aumentare la capacity. Potenzialmente nella modalità tradizionale il team dovrebbe essere più numeroso anche perché, da un punto di vista consulenziale, bisogna concludere tutte le attività previste e concordate con il cliente. Inoltre, nel momento in cui dovessero esserci delle attività non previste e la data di Go Live non può essere posticipata, bisognerà parallelizzare le attività allocando, dunque, maggiori risorse al team. In Agile invece, non c'è questa possibilità, perché inizialmente si dimensiona il team definendone i componenti e successivamente, durante le prime iterazioni, si definisce il numero di story points che quel team riesce a sviluppare. In questo modo, durante gli Sprint Planning si valutano solamente gli story points che si riescono a sviluppare e, tutte le storie extra saranno valutate negli Sprint successivi. Fino a questo momento, l'andamento dei vari Sprint, può essere riassunto come riportato nella *Tabella 2*.

	Points Planned	Points Completed
Sprint 1	35	41
Sprint 2	40	40
Sprint 3	43	33
Sprint 4	42	42
Sprint 5	45	39

Tabella 2 Andamento degli Sprint

Nella colonna Points Planned sono riportati gli story point che sono stati concordati con il cliente in fase di Sprint Planning. Nella seconda colonna invece, denominata Points Completed, sono riportati gli story point effettivamente conclusi alla fine di ogni Sprint. Nel *Paragrafo 4.2* sono riportate le motivazioni delle principali discrepanze tra gli story points previsti, e quelli effettivamente completati. In questo

paragrafo però, è importante analizzare gli incrementi principali che il team ha sviluppato durante questi Sprint e che hanno portato ad una maggiore User Adoption. In particolare, le stories più importanti possono essere così riassunte:

- Sono state introdotte ulteriori notifiche che potessero aiutare gli user nell'avanzamento del processo. In particolare, nel caso in cui una Issue dovesse essere riassegnata ad un determinato Manager, esso riceverà una notifica con le motivazioni della riassegnazione;
- È stata migliorata la reportistica che può essere scaricata direttamente dal DIM. Nello specifico, è stato implementato un template Excel, in grado di riportare e classificare le Issue presenti sulla piattaforma che risultano essere ancora attive, in base ad alcuni parametri definiti dal cliente. È stata, inoltre, fornita la possibilità di scaricare un Power Point contenente un sommario con le informazioni di particolari tipologie di Issues;
- Sono state introdotte nuove origini delle Issue e, dunque, possono esserne gestite e registrate più tipologie rispetto a quelle presenti nell'MVP;
- Sono stati introdotti ulteriori ruoli. Per ogni tipologia di problematica gestita attraverso la piattaforma, è identificata una figura che ricopre un determinato ruolo e che ha determinate responsabilità. Avendo implementato diverse tipologie di Issue, sono state introdotte a sistema ulteriori figure manageriali;
- È stato migliorato il processo che permette di allegare i file ad una determinata Issue. Difatti è stata implementata la funzionalità del Drag and Drop per migliorare l'esperienza degli utenti;
- È stata introdotta la possibilità di consultare la guida del Development Issue Management all'interno della piattaforma stessa in modo che gli user possano verificarne tutte le funzionalità.

Ad oggi il team deve ancora sviluppare innumerevoli incrementi; difatti, nello Sprint Backlog sono attualmente presenti all'incirca 350 story points. Ciò implica che, se venisse mantenuta la velocity che il team attualmente sta sostenendo, a parità di team e di condizioni esterne, saranno necessari orientativamente 9 Sprint da 40 story points ciascuno. Chiaramente, come è previsto dalla teoria dell'Agile stessa, il Product Backlog è in continua evoluzione e, pertanto, potrebbero essere richieste molte più funzionalità di quelle attualmente previste, incrementando così il numero di Sprint necessari.

CAPITOLO 5: CONCLUSIONI ED EVOLUZIONI FUTURE

Dopo un mese e mezzo svolto in modalità Waterfall e dopo aver effettuato 5 Sprint in modalità Agile, il cliente ha iniziato ad apprezzare la seconda modalità progettuale per diverse motivazioni. In particolare, ha notato un coinvolgimento degli user che utilizzano la piattaforma DIM migliorando così il KPI che indica la **User Adoption**. Difatti, gli utenti sentendosi più coinvolti, utilizzano con una frequenza sempre maggiore il tool creato e mostrano interesse nel proseguo degli sviluppi, presentando al business tutti i miglioramenti e le modifiche che potrebbero essere apportati. Inoltre, anche a livello di **gestione dei costi** relativi al progetto, il cliente preferisce la modalità Agile in quanto può monitorare l'andamento del lavoro ogni due settimane, concentrandosi sui nuovi sviluppi, riuscendo così a quantificare e comprendere meglio il valore di quanto offerto dall'azienda Reply. Infine, anche per quanto riguarda il fornitore questa modalità risulta essere migliore sia per quanto concerne la riduzione dei costi che la stessa deve sostenere, sia per quanto riguarda la **qualità del software** intesa come soddisfazione del cliente. Per tali ragioni, questa progettualità continuerà ad essere gestita in modalità Agile. Difatti, si stanno già iniziando ad inserire nel Product Backlog una serie di User Stories che saranno analizzate nel corso dei prossimi Sprint. Difatti, per rendere la piattaforma DIM sempre più completa si stanno analizzando due importanti sviluppi futuri:

- Chat GPT: si tratta di un chatbot basato su Intelligenza Artificiale il quale utilizza una tecnica di apprendimento di deep learning, nota come transformer. Difatti, risulta essere in grado di garantire un livello di conversazione molto avanzato e dettagliato sulle tematiche più disparate. Nel contesto di questa progettualità, ChatGPT raccoglierà informazioni sulla Issue specifica registrata a sistema sulla quale si vogliono avere informazioni e risponderà alle domande poste dagli utenti nella lingua preferita. Dopo

l'interazione l'utente potrà indicare se la risposta sia stata utile o non sufficiente. L'interazione rimane visibile a tutti gli utenti all'interno della issue e può essere riaperta in qualsiasi momento. Si stanno pertanto analizzando le ulteriori possibili applicazioni che tale strumento potrebbe avere sulla piattaforma DIM.

- Integrazioni esterne: attualmente il Development Issue Management risulta essere chiuso e senza integrazioni. Si sta dunque pensando ad un nuovo ambiente complessivo, costruito secondo il paradigma IaaS/PaaS, come richiesto dal cliente, in modo che non sarà necessario mantenere l'infrastruttura del sistema. Sulla base delle informazioni disponibili e dei requisiti del cliente, l'architettura proposta sarà quella riportata in *Figura 5.1*.

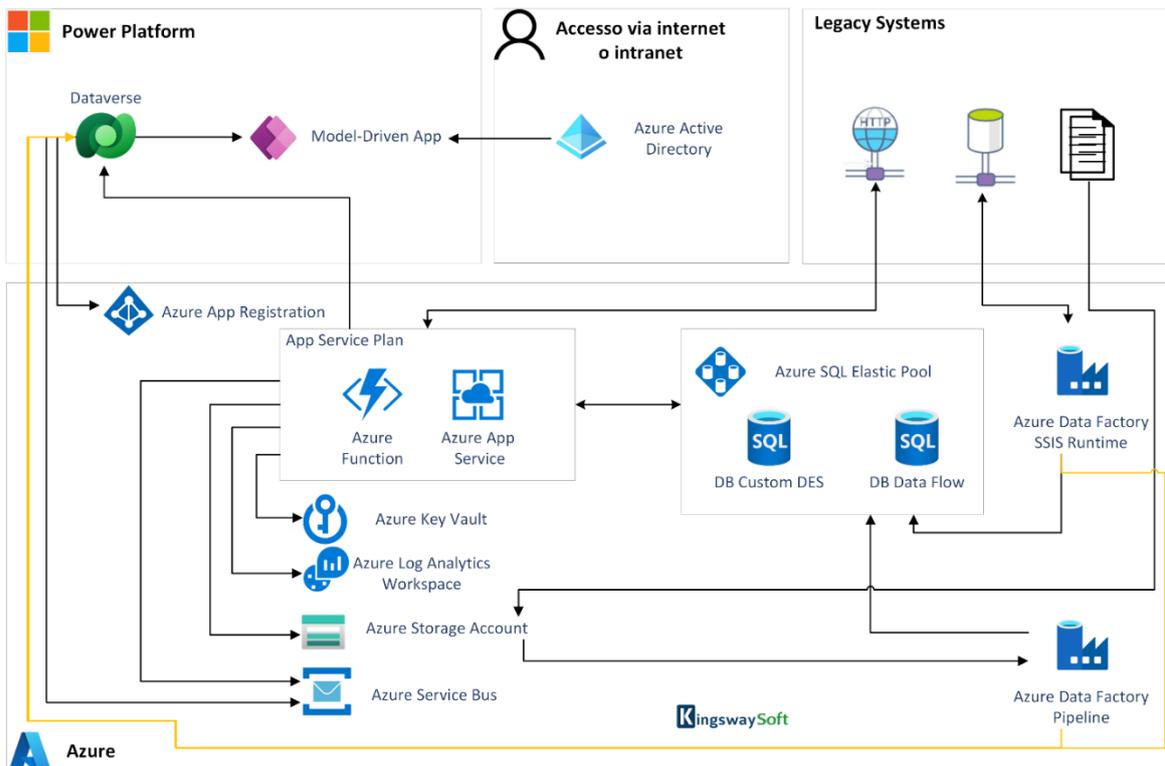


Figura 5.1 Solution Design - confidential

Per quanto concerne la gestione in generale delle future progettualità, seguendo il Cynefin Framework si può delineare a priori una scelta sulla modalità progettuale più indicata per gestire i progetti. Difatti, se ci si trova nel dominio del semplice o del complicato, in cui l'approccio risulta essere di tipo **predittivo** poiché vi è la presenza di relazioni *causa – effetto* in qualche misura note, è preferibile utilizzare l'approccio Waterfall. Al contrario, se ci si trova nel dominio del complesso o del caotico, in cui l'approccio risulta essere di tipo **adattivo**, la responsabilità è condivisa e le decisioni sono distribuite tra gli attori, è preferibile utilizzare l'approccio Agile. Oltre a questo fattore però bisogna porre particolare attenzione sulla predisposizione del cliente ad adottare una metodologia progettuale piuttosto che un'altra. Soprattutto per quanto concerne la modalità Agile, è necessario infatti che il cliente ne conosca e ne abbracci i principi, soprattutto perché dovrà essere identificata la figura del Product Owner dal cliente stesso. Infine, sono importanti anche altri fattori come, ad esempio, le aspettative che il cliente ha sul progetto stesso. Ad esempio, se esso avrà un budget fisso, il tempo fisso e le idee chiare su cosa si aspetta, allora è preferibile andare in Waterfall. Altrimenti se il cliente ha un business labile, potrebbero cambiare le condizioni di mercato in quanto si trova di fronte a tecnologie disruptive e predilige sviluppi incrementali e veloci, allora è preferibile la modalità Agile. Per quanto concerne lo specifico contesto aziendale, bisogna specificare che l'azienda Reply preferirebbe lavorare in modalità Agile in quanto, da un punto di vista consulenziale tale modalità aiuterebbe a ridurre l'incertezza. Purtroppo però, questa decisione non dipende soltanto dal gruppo Reply, ma si deve tenere in considerazione il fattore della **cultura aziendale** di cui si è discusso nei precedenti paragrafi. Difatti, non si riesce ad applicare Agile puro in quanto il cliente ha la necessità di un periodo di rodaggio per comprendere il funzionamento del framework Scrum. Inoltre, il cliente sembra essere legato alla metodologia Waterfall poiché l'impostazione di tali aziende prevede l'allocazione di un determinato budget al progetto. Pertanto, se tale progettualità è svolta in modalità Waterfall, il cliente riesce a perimetrare sin dall'inizio cosa sarà sviluppato

e a quale prezzo, riducendo così dal suo punto di vista l'incertezza legata alla metodologia Agile per la quale non si può prevedere cosa si riuscirà a sviluppare con il budget a disposizione. Difatti, il principale contro evidenziato dal cliente per quanto concerne la metodologia Agile, è proprio quello di non sapere sin dall'inizio dove si arriverà al termine del progetto. Per tutte le ragioni elencate, Reply sta cercando di promuovere anche nelle altre progettualità, la modalità utilizzata per il progetto in esame; ovvero propone una prima fase progettuale svolta in modalità Waterfall in modo che il cliente possa avere un MVP in tempi relativamente brevi e riesca a capire la direzione del progetto e successivamente una seconda fase progettuale svolta in modalità Agile per apportare incrementi in maniera iterativa. Nella *Tabella 3*, sono riportate le principali differenze tra le due modalità progettuali in termini di processi organizzativi, ruoli e processi informativi, ponendo particolare attenzione su quanto si è verificato nel progetto oggetto di tesi.

	WATERFALL	AGILE
Suddivisione Progetto	<p>Fasi distinte</p> <p>Ogni fase è stata eseguita dopo la fine della precedente e solamente una volta</p>	<p>Sprint</p> <p>Attualmente sono stati svolti 5 sprint, e in ogni sprint sono state eseguite le fasi di sviluppo test etc.</p>
Sviluppo Progetto	<p>Sequenziale</p> <p>Le principali fasi in sequenza sono state: Approvazione Analisi Funzionale, Sviluppo, Test, rilascio in produzione</p>	<p>Incrementale</p> <p>Durante ogni Sprint Planning il cliente spiega cosa si aspetta dai nuovi sviluppi e il team lo implementa</p>
Struttura	<p>Rigida</p> <p>Sin dalle fasi iniziali erano ben chiari tutti i requisiti</p>	<p>Flessibile</p> <p>Di volta in volta veniva popolato il Product Backlog per i nuovi sviluppi</p>
Ruoli	<p>Project Manager</p> <p>Responsabile degli obiettivi di progetto</p>	<p>Scrum Master</p> <p>Coach che detta i tempi e risolve gli impedimenti per il development team</p>

Gestione Team	Development Team Il Project Manager assegnava loro i task	Development Team Ogni mattina il team si auto organizza per la gestione dei task
Struttura team	Team più numeroso È possibile aumentare la numerosità del team nel caso di crashing di alcune attività.	Team meno numeroso È preferibile che il team sia stabile e non vi è la necessità di fare crashing delle attività in quanto si prende in considerazione solo quello che si riesce a sviluppare
Rapporti con il cliente	SAL settimanale Una volta a settimana si parlava con il business del cliente	Cerimonie Agile Una volta al giorno si svolge il Daily Standup Meeting in cui partecipa anche il cliente
Feedback del cliente	Nella fase finale del progetto È solo nell'ultima fase che il cliente ha potuto vedere il risultato del progetto	Durante l'intero Sprint Attraverso la figura del Product Owner il cliente vede gli sviluppi man mano che sono effettuati

Tabella 3 Differenze Waterfall e Agile

BIBLIOGRAFIA

- [2] Archibald 2003. R.D. Archibald, Project management: la gestione di progetti e programmi complessi, Franco Angeli, Milano 2003
- [3] Clegg S., Killen C. P., Biesenthald C., Sankaranc S., International Journey of Project Management, pp. 762-772, Practices, projects and portfolios: Current research trends and new directions
- [4] Larman C., Basili V.R., 2003, Computer 36 (6) pp. 47-56, Iterative_and_incremental_developments_a_brief_history
- [5] Alberto De Marco, Project management for facility constructions 2nd Edition, Springer, 2018

SITOGRAFIA

- [1] <https://www.tableau.com/it-it/learn/articles/smart-goals-criteria>
[Crea obiettivi SMART più efficaci - Suggerimenti con esempi • Asana](#)
- <https://www.researchgate.net/publication/313768756> [The Waterfall Model and the Agile Methodologies A comparison by project characteristics](#)
[Iterative and incremental developments. a brief history | IEEE Journals & Magazine | IEEE Xplore \(polito.it\)](#)
- [Scrum: cos'è, come funziona e com'è strutturato - Agile Way](#)
- [6] [Il team Scrum e i ruoli - Agile Way](#)
- [7] [Sprint Planning meeting - Agile Way](#)
- [8] [Il Daily Scrum Meeting - Agile Way](#)
- [9] [Retrospective meeting - Inspect and Adapt! - Agile Way](#)
- [Agile VS Waterfall: due metodologie a confronto \(appvizer.it\) \(???\)](#)
- [10] [Cynefin: un framework per aiutare il processo decisionale negli ambienti complessi - Project Management Europa](#)
- [11] https://www.reply.com/contents/Relazione_Trimestrale_Q3_2022-1.pdf
- [12] [Reply named a Leader in the May 2021 Gartner Magic Quadrant for CRM and Customer Experience Implementation Services | Reply](#)
- [13] [Metodo 8D Problem Solving: Consulenza automotive \(reconsultsrl.it\)](#)
- [14] [Ristampa Gartner](#)