



**Politecnico
di Torino**

POLITECNICO DI TORINO

DIPARTIMENTO DI INGEGNERIA GESTIONALE

Corso di Laurea in Ingegneria Gestionale 'industria 4.0'

ANNO ACCADEMICO:2024/2025

Tesi di Laurea Magistrale - Novembre 2024

**ANALISI DEI PRINCIPALI FRAMEWORK AGILE APPLICATI
A PROGETTI IT DI UNA AZIENDA CHE OPERA
NEL SETTORE TURISTICO**

Relatore: Mangano Giulio

Candidato: Sartore Lorenzo

Indice

1	Metodi di gestione dei progetti	9
1.1	Definizione di gestione di un progetto	9
1.2	Metodi di gestione di progetti	10
1.2.1	Metodologia tradizionale: 'Waterfall'	10
1.2.2	Metodologia: 'Agile'	13
1.2.3	Team Agile	17
1.2.4	Metodologie ibride	20
2	Maggiori framework Agile: Scrum e Kanban	22
2.1	Framework Scrum	22
2.1.1	Origini della metodologia	23
2.2	Principali strumenti Scrum	25
2.2.1	Sprint	25
2.2.2	Product Backlog	26
2.2.3	Monitorare l'avanzamento in uno Sprint	29
2.3	Principali cerimonie Scrum	32
2.3.1	Sprint Planning	32
2.3.2	Daily Scrum	33
2.3.3	Sprint Review	34
2.3.4	Sprint Retrospectives	34
2.4	Framework Kanban	35

2.4.1	Origini della metodologia	36
2.4.2	Strumenti e cerimonie	37
2.5	Confronto tra i diversi framework	38
2.6	Framework Hybrid	39
3	Framework agili applicati a progetti IT di una società di viaggi (Alpitour s.p.a.)	40
3.1	Alpitour s.p.a.: Introduzione all'azienda	40
3.1.1	Organizzazione dell'area Information Technology	41
3.2	Presenza e impatto dell'Agile da un punto di vista interno	42
3.2.1	Raccolta dei dati	43
3.2.2	Descrizione del campione	44
3.2.3	Test non parametrico di Kruskal-Wallis	48
3.3	Ciclo-vita dei progetti	52
3.4	Valutazione dell'andamento di progetti agili tramite KPI	53
3.4.1	Key Performance Indicator	53
3.4.2	Accuratezza dell'effort previsto	53
3.4.3	Lead Time	55
3.4.4	Cycle Time	56
3.4.5	Numero di PBI svolti nella durata di uno Sprint	57
3.5	Valutazione di progetti aziendali, mediante l'utilizzo di KPI	58
3.5.1	WebCola	58
3.5.2	Turisanda (Eventi)	63
3.5.3	Nuovo Flusso	69
3.5.4	Via Milano Parking	73
3.5.5	Confronto dei KPI raccolti	78

Elenco delle figure

1.1	Immagine dello schema di metodologia 'Waterfall'	11
1.2	Immagine dello schema di metodologia 'Agile'	14
1.3	Immagine dello schema di una struttura tipica di Team 'Agile'	18
1.4	Immagine della formazione delle tecniche Ibride	21
2.1	Grafico del numero di documenti relativi al framework Scrum pubblicati per anno	24
2.2	Numero dei documenti per tipo	24
2.3	Rappresentazione dei cicli Sprint consecutivi	25
2.4	Rappresentazione della gerarchia degli item di progetto	28
2.5	Rappresentazione di un grafico Burndown	29
2.6	Rappresentazione di un grafico Velocity	30
2.7	Rappresentazione di un CFD	31
2.8	Rappresentazione schematica di 'tips' per lo svolgimento di uno Sprint Planning	33
2.9	Rappresentazione del numero di documenti relativi a Kanban pubblicati per anno	36
2.10	Rappresentazione del numero di documenti relativi a Kanban pubblicati per tipo	37
3.1	Principali Brand di Alpitour	40
3.2	Organizzazione teams Area IT Alpitour	42
3.3	Grafico a torta dell'età dei rispondenti	45
3.4	Grafico a torta del sesso dei rispondenti	45

3.5	Grafico a torta del titolo di studio dei rispondenti	46
3.6	Grafico a torta degli anni di esperienza dei rispondenti	46
3.7	Grafico a barre dei ruoli in azienda dei rispondenti	47
3.8	Risultati del Test di Kruskal-Wallis	49
3.9	Risultati del Test di Kruskal-Wallis	50
3.10	Grafico che rappresenta il lead time	55
3.11	Grafico che rappresenta il Cycle Time	56
3.12	Logo WebCola	58
3.13	Grafico a torta dei Work item per tipo relativi a WebCola	59
3.14	Effort e Completed work	60
3.15	Medie di Effort e Completed work per Work item type	60
3.16	Lead Time per Work Item Type	61
3.17	Cycle Time per Work Item Type	62
3.18	Logo Turisanda	63
3.19	Work Items Turisanda	64
3.20	Burndown Chart Turisanda	64
3.21	Effort e Completed work a confronto	65
3.22	Effort e Completed work a confronto per Work item	66
3.23	Lead Time Turisanda	67
3.24	Cycle Time Turisanda	68
3.25	Grafico a torta relativo ai Work Items di Nuovo Flusso	69
3.26	Effort e Completed work a confronto	70
3.27	Effort e Completed work di task e bug a confronto	71
3.28	Grafico a barre del Lead Time per Work Item di Nuovo Flusso	71
3.29	Grafico a barre del Cycle Time per Work Item di Nuovo Flusso	72
3.30	Logo Via Milano Parking	73
3.31	Grafico a torta dei work item di Via Milano Parking	74

3.32 Grafico di Effort e Completed Work di Via Milano Parking 75

3.33 Grafico a barre sul Lead Time degli item di Via Milano Parking . . . 76

3.34 Grafico a barre sul Cycle Time degli item di Via Milano Parking . . . 77

3.35 Tabella contenente i KPI trovati 78

Abstract

Il presente lavoro di tesi esplora la metodologia Agile di gestione dei progetti, ovvero una serie di tecniche nate in risposta alla poca flessibilità e difficoltà di adattamento dei metodi di gestione già esistenti.

Attraverso lo studio della letteratura riferita all'argomento, è stato possibile portare all'interno della prima parte di questo testo le caratteristiche che contraddistinguono questo metodo e alcune sue sfaccettature, mettendo in evidenza i benefici e i limiti, fornendo un confronto approfondito con i metodi tradizionali di gestione dei progetti, i cosiddetti metodi 'a cascata'.

Nella parte successiva sono poi presentate e analizzate tre varianti: Scrum, Kanban e Hybrid, che corrispondono ai principali framework utilizzati quando si tratta di applicare l'Agile alla gestione dei progetti. Per ogni variante, sono riportate non solo le caratteristiche, ma sono anche discussi i ruoli e le responsabilità dei membri del team Agile e l'importanza della comunicazione tra le diverse funzioni aziendali.

Inoltre, nell'ultima sezione è riportato il caso dell'area Information Technology dell'azienda Alpitour S.p.a., la quale si è affacciata all'utilizzo di queste metodologie Agili nel corso degli ultimi anni.

Tramite un questionario, sono stati raccolti dati da molti dei dipendenti di quest'area, attraverso domande utili sia per capire le caratteristiche del rispondente, sia per raccogliere informazioni riguardanti le opinioni nei confronti dei metodi e dei risultati degli stessi.

Proseguendo verso la conclusione, è presente un'analisi svolta attraverso l'utilizzo di alcuni Key Performance Indicators (KPI), utile per monitorare le performances e lo stato di avanzamento dei progetti aziendali gestiti mediante queste tecniche innovative, con il fine di fornire una valutazione oggettiva sull'efficacia dell'implementazione dei metodi e di individuare eventuali aree di miglioramento.

Introduzione

La gestione Agile dei progetti è una realtà che sta prendendo piede in un numero di aziende via via crescente, poichè essa permette di procedere con lo stato di avanzamento dei progetti, di sviluppare e rilasciare funzionalità tramite un processo iterativo di sviluppo continuo, formato da cicli di lavoro relativamente brevi.

Generalmente, seguendo queste tecniche, si fornisce maggiore autonomia ai singoli team, i quali hanno la libertà di organizzare e riadattare il lavoro secondo le proprie necessità e disponibilità.

Inoltre, rendendo lo specifico team più autonomo, e dunque fornendogli più 'potere', si notano anche dei benefici in termini di motivazione del singolo individuo, migliorandone quindi la resa.

Il fatto di gestire il lavoro in maniera 'agile' non impatta soltanto sulle condizioni degli individui costituenti la squadra, ma ha anche un impatto direttamente sui risultati del progetto, potenziando la qualità di prodotti e servizi, superando anche i vincoli di budget e soprattutto di tempo.

L'elaborato, in seguito ad una prima parte esplicativa dove si affrontano nel dettaglio i principali framework (con annessi vantaggi e svantaggi di ognuno di essi), verifica la veridicità di quanto detto precedentemente, utilizzando dati provenienti direttamente sia dalle opinioni personali dei dipendenti che lavorano a stretto contatto con il metodo in questione sia dai progetti appartenenti all'area IT di Alpitour S.p.a., ovvero una grande società italiana di Tour Operating.

1 Metodi di gestione dei progetti

1.1 Definizione di gestione di un progetto

Per introdurre il concetto di gestione di un progetto, è necessario prima chiarire che cosa si intenda con il termine stesso ed essere in grado di individuarne le caratteristiche principali. Possiamo dare diverse definizioni di 'progetto'. A livello più generale è definito come:

Definizione 1.1.1. *Impegno temporaneo per raggiungere uno o più obiettivi definiti. [3]*

Altre definizioni più specifiche invece possono essere:

Definizione 1.1.2. *Un progetto è un complesso coordinato di azioni finalizzate ad ottenere risultati determinati, nel rispetto dei limiti stabiliti di tempo e risorse. [2]*

Definizione che introduce già dei vincoli in termini di tempo speso e risorse impiegate. O ancora, aggiungendo un ulteriore concetto:

Definizione 1.1.3. *Un progetto è un'impresa unica e di durata limitata, con lo scopo di conseguire obiettivi pianificati, che possono definirsi in termini di prodotti ('output'), risultati finali ('outcomes') o benefici. Un progetto si ritiene di norma un successo se raggiunge gli obiettivi in linea con i rispettivi criteri di accettazione, entro un arco temporale ed un budget concordati. [4]*

In quest'ultima definizione, sebbene non sia esplicitato, si fa riferimento al valore aggiunto che un progetto deve essere in grado di portare all'azienda, concetto fondamentale, che da ragione allo stesso di esistere e ci ricollega con il suo senso di 'gestione'.

Il sistema di gestione ha come fine ultimo i risultati. Se volessimo fornire una vera e propria definizione questa potrebbe essere :

Definizione 1.1.4. *gestione sistemica di un'impresa complessa, unica e di durata determinata, rivolta al raggiungimento di un obiettivo chiaro e predefinito mediante un processo continuo di pianificazione e controllo di risorse differenziate e con vincoli interdipendenti di tempi-costi-qualità. [2]*

1.2 Metodi di gestione di progetti

Dopo aver definito i concetti di 'progetto' e di 'gestione di progetto' è opportuno ora, analizzare quali siano i metodi di gestione più utilizzati dalle aziende, focalizzando l'attenzione sulle tecniche odierne così come su quelle del passato.

1.2.1 Metodologia tradizionale: 'Waterfall'

Quando si parla di metodologia tradizionale, si fa riferimento al cosiddetto modello 'a cascata' (anche noto come Linear Sequential Life Cycle Model). Sia che si stia parlando di creazione o di modifica di un software o che si parli della costruzione di un edificio o di un ponte, questo resta il modello più classico per la gestione di progetto. Esso si contraddistingue per alcune sue caratteristiche, come ad esempio la sua struttura, essendo composto da una serie di fasi ben definite ed in sequenza. Un altro tratto distintivo è costituito dal processo seguito durante la gestione. Infatti, ogni sua fase deve essere non solo completata, ma anche verificata prima che si possa procedere con lo svolgimento della fase successiva.

Struttura del modello

Si analizza ora la struttura tipica di questo tipo di modelli, focalizzandosi su progetti informatici. Le principali fasi del modello 'waterfall' sono:

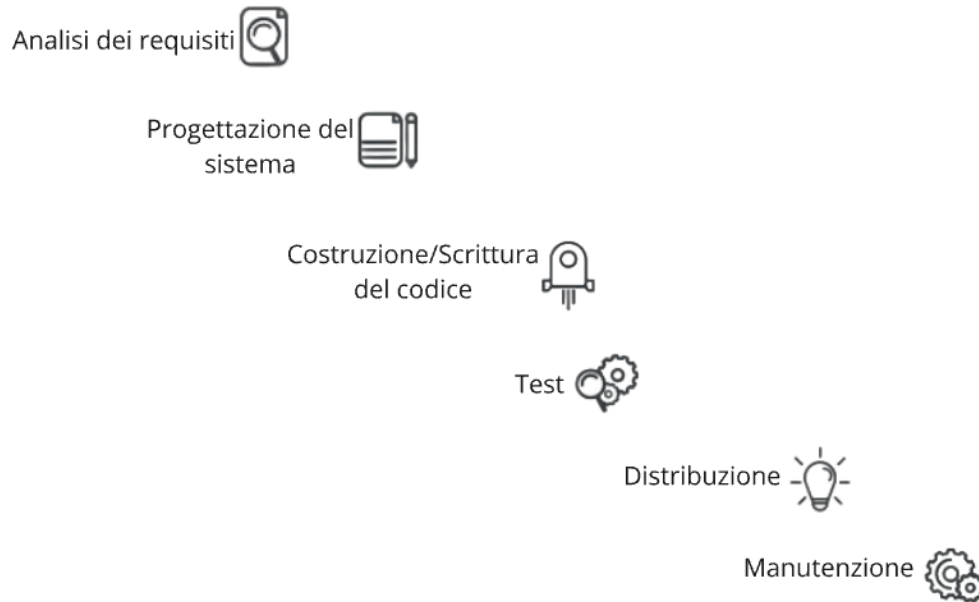


Figura 1.1: Immagine dello schema di metodologia 'Waterfall'

1. **Analisi dei requisiti:** In questa prima fase, si raccolgono, documentano e descrivono tutti i requisiti necessari al successo del progetto. Qui entrambe le parti lavorano in modo coordinato (stakeholder e team di progetto) per definire in modo chiaro e senza equivoci, non solo le specifiche tecniche del progetto, ma anche le funzionalità e gli obiettivi.
2. **Progettazione del sistema:** Dopo averne definito e compreso i requisiti, il team incaricato sviluppa un progetto dettagliato dell'intero sistema, sviluppandone l'intera architettura e la totalità delle risorse digitali (come hardware, software, interfacce, database, ecc)
3. **Costruzione/Sviluppo del codice:** Terminata la progettazione, si passa alla fase di implementazione di quanto fatto negli step precedenti. Se si trattasse di un progetto nell'ambito delle construction, questo step consisterebbe nella costruzione vera e propria, mentre se l'ambito del progetto fosse l'area IT, generalmente questa fase corrisponderebbe alla scrittura del codice.
4. **Test di verifica:** Successiva a quella di sviluppo, è la fase in cui si controlla che il lavoro sia stato svolto nel modo corretto, e che dunque oltre a funzionare

correttamente, soddisfi tutti i requisiti di progetto stilati nella fase alta della 'cascata'. In questo momento, possono essere fatti diversi test: unitari, di integrazione, di sistema e di accettazione.

5. **Distribuzione/Rilascio:** Specialmente in ambiente informatico, se la verifica ha un esito positivo, il sistema progettato può essere implementato nell'ambiente di produzione. Ciò significa che, se si sta parlando di un software, questo venga distribuito, se invece il progetto in questione è riferito ad un hardware, che questo venga installato.
6. **Manutenzione:** L'ultima fase della metodologia tradizionale è proprio la manutenzione. In essa, vengono apportati aggiornamenti e miglioramenti per fare in modo che il prodotto continui a rispondere correttamente alle esigenze mutevoli degli utenti. Inoltre viene risolto qualsiasi tipo di 'bug' si crei, dovuto ad errori nelle fasi precedenti o a malfunzionamenti.

Qui di seguito è riportata una breve analisi dei punti di forza e delle debolezze di questo metodo.

Vantaggi

- **Semplice da comprendere:** generalmente non lascia spazio ad equivoci, essendo l'iter di sviluppo lineare ed intuitivo, ed essendo gli obiettivi ben definiti e difficilmente modificabili. Per un project manager alle prime armi può essere utile cominciare da questo tipo di progetti; [7]
- **Semplice da monitorare:** le pietre miliari sono chiare, così come la documentazione, che deve essere completa in ogni fase del progetto. Questi due elementi combinati fanno sì che sia molto semplice tenere traccia dell'avanzamento e individuare eventuali colli di bottiglia. Un'altra considerazione a favore è che questo crea una buona base per eventuali nuovi arrivati nel team di progetto. [7]
- **Semplicità di gestione:** essendo il flusso di progetto intuitivo e 'naturale', i project manager non incontrano particolari difficoltà nel seguire il processo nel corso del suo sviluppo. Rimanere allineati sull'avanzamento non appare come un ostacolo, dato che i requisiti vengono stabiliti nella prima fase e restano invariati lungo tutta la durata del progetto.

Svantaggi

- **Gestione del rischio inefficace:** non è possibile effettuare un'analisi del rischio iterativa ed incrementale, bensì ci si accorge di eventuali rischi, solamente in una fase avanzata del progetto e dover apportare una modifica per garantire un certo livello di qualità, significa dover ripercorrere tutte le fasi dall'inizio, causando un'ingente spesa in termini di tempo e di costi. [7]

- **Rigidità:** non si può trascurare questo vincolo, in quanto rappresenta uno dei principali limiti del metodo. Questa si nota, sia nel momento in cui è necessaria una modifica, sia per quanto riguarda l'aspetto dell'adattabilità. Infatti, questo modello è fortemente sconsigliato nei progetti in cui è plausibile un'evoluzione rapida nel corso del tempo. [7]
- **Stakeholder poco coinvolti:** Essi vengono coinvolti molto nella fase iniziale, ovvero quella di raccolta e descrizione dei requisiti, però poi non sono resi particolarmente partecipi fino alle fasi finali. Questo lungo lasso di tempo, caratterizzato da pochi allineamenti, può essere dannoso in quanto esiste il rischio di allontanarsi da ciò che erano le volontà degli stakeholder al momento d'inizio del progetto.

1.2.2 Metodologia: 'Agile'

La metodologia Agile è un insieme di tecniche volte principalmente alla gestione dei progetti Software, i quali hanno necessità di essere gestiti in maniera flessibile per poter rispondere tempestivamente ai rapidi cambiamenti di questa categoria di progetti. Gestire progetti Software con il metodo tradizionale potrebbe risultare difficile e particolarmente oneroso. Questo metodo è più recente di quello tradizionale, tant'è che le sue origini possono essere attribuite a pratiche svolte nei primi anni Novanta, anche se si riconosce in un vero e proprio punto di svolta la conferenza tenuta nel Febbraio 2001 a Snowbird, Utah, in cui un gruppo di esperti del settore e di queste metodologie, si riunì per discutere dei personali approcci nei confronti di questa tecnica riuscendo ad elaborare il "Manifesto Agile".

Struttura del modello

Il modello in questione si discosta completamente dalla struttura lineare della tradizione, e assume una forma ciclica. Come si può notare nell'immagine sottostante, le fasi percorse sono praticamente analoghe, con la sostanziale differenza che vengono svolte ripetutamente in periodi di tempo relativamente brevi e applicate con un focus su parti minori del progetto. Non è necessaria una nuova spiegazione per nessuna di queste fasi, è sufficiente mettere in chiaro la sostanziale differenza rappresentata dalla ciclicità.

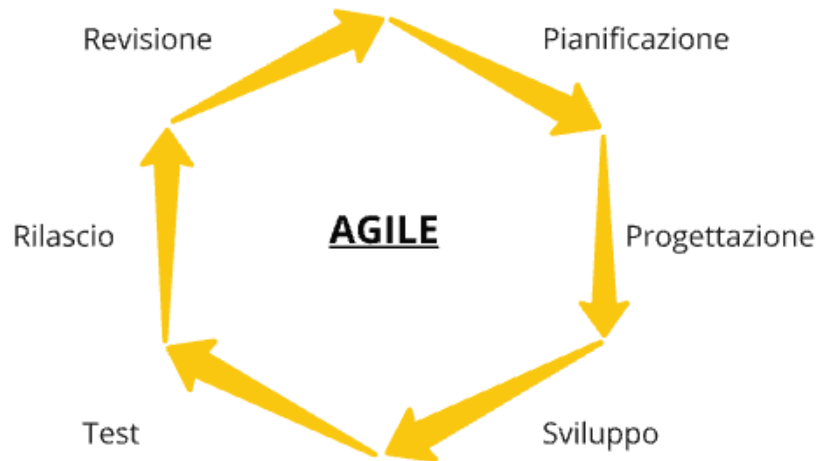


Figura 1.2: Immagine dello schema di metodologia 'Agile'

Manifesto Agile

Questo testo racchiude quelli che sono i principi costituenti la filosofia della metodologia stessa. Scritto da un gruppo di diciassette esperti, è un'opera che ancora oggi resta tra i maggiori testi riguardanti l'Agile.

Si basa su quattro principi fondamentali:[13]

1. Individui e interazioni piuttosto che processi e strumenti:

Concentrandosi su individui ed interazioni ci si trova in presenza di team adattivi e reattivi, in grado di risolvere ogni problema scegliendo lo strumento (o il processo) più adeguato. Questo permette ai membri del team di esprimere liberamente il proprio valore.

2. Software funzionante piuttosto che documentazione completa:

Secondo i metodi tradizionali, la gestione di un progetto prevede la creazione di una documentazione dettagliata. Ciò comporta un grande ostacolo da superare in termini di tempi e produttività. Con il metodo Agile ci si è accorti che il valore primario fornito al cliente è dato da un software funzionante, in quanto dimostra che il prodotto è utilizzabile e soddisfa i requisiti per cui è stato progettato, evitando problemi di staticità che si verrebbero a creare dovendo far fede ad un documento troppo dettagliato.

3. Collaborazione con il cliente piuttosto che negoziazione del contratto:

Questo principio consiste nel cambiamento in termini di contratto. Non è opportuno utilizzare risorse in termini di tempo e di denaro per negoziare un contratto dettagliato, bensì è preferibile avviare un rapporto costituito da

continue interazioni in modo da rimanere allineati sulle richieste del cliente nel corso dell'intero progetto.

4. **Rispondere al cambiamento piuttosto che seguire un piano:**

Nell'approccio tradizionale, apportare delle modifiche in corso d'opera, discostandosi dal progetto iniziale, comporta dei costi non irrilevanti. Secondo i metodi innovativi invece questi sarebbero più contenuti. I cambiamenti in itinere sono accettati, perché si tratta di feedback che garantiscono la qualità del prodotto finale, massimizzandone il valore aggiunto.

Questi racchiudono l'essenza di quello che è l'approccio da mantenere se si vuol gestire un progetto software in maniera dinamica, adattiva e quindi efficace. Ciò che il Manifesto propone, inoltre, è supportato da dodici principi fondamentali: [13]

- **La nostra massima priorità è soddisfare il cliente rilasciando software di valore, fin da subito e in maniera continua:**

E' necessario soddisfare il cliente tramite delle consegne ricorrenti, generalmente con frequenza costante, poiché chi usa i software è abituato ad ottenere aggiornamenti continui ed istantanei.

- **Accogliamo i cambiamenti nei requisiti, anche a stadi avanzati dello sviluppo. I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente:**

Nel mercato odierno il cambiamento è una costante. Sarebbe troppo difficile seguire i mutamenti dopo aver redatto dei documenti dettagliati all'inizio del progetto (così come prevedono le tecniche di gestione tradizionali).

- **Consegnamo frequentemente un software funzionante, con cadenza variabile da un paio di settimane a un paio di mesi, preferendo i periodi brevi:**

Sebbene non sia necessaria una documentazione iniziale sulla quale fare affidamento, è necessario avere dei documenti almeno come riscontro per la pianificazione dello sviluppo del software. Rilasciare un software funzionante frequentemente migliora la qualità del prodotto finito.

- **Committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto:**

Il cliente, gli sviluppatori ed i dirigenti lavorano a stretto contatto con allineamenti quasi giornalieri per riuscire a completare il progetto nel migliore dei modi. Come già detto precedentemente, la collaborazione è un tassello fondamentale e non può essere omessa. Si aspira ad un sistema di gestione fortemente integrato con il processo, per poter individuare eventuali modifiche di miglioramento.

- **Fondiamo i progetti su individui motivati. Diamo loro l'ambiente e il supporto di cui hanno bisogno e confidiamo nella loro capacità di portare il lavoro a termine:**

Lasciare ai team una certa libertà, anche non fornendo loro una documentazione eccessivamente rigida e dettagliata a cui far fede, rende gli individui maggiormente motivati ed incentivati dalla sfida. La motivazione individuale è appunto molto importante e deve essere mantenuta ad un livello elevato ed il contesto lavorativo è molto importante per far sì che quanto accada.

- **Una conversazione faccia a faccia è il modo più efficiente e più efficace per comunicare con il team e per le comunicazioni interne:**

Il cosiddetto 'confronto diretto' è il miglior modo per effettuare delle comunicazioni, sia che queste siano interne che esterne al team. Inoltre, in questo modo, si migliorano anche i rapporti interpersonali creando un ambiente di lavoro più piacevole.

- **Il software funzionante è il principale metro di misura di progresso:**

La misura più importante per valutare lo stato di successo di un progetto non è più il rispetto dei tempi di consegna, ma essendo i rilasci di software molto frequenti, il nuovo indicatore di performances è proprio la qualità di quest'ultimo.

- **I processi agili promuovono uno sviluppo sostenibile. Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante:**

La gestione Agile è un processo iterativo continuo, per questo motivo è necessario che tutti gli attori coinvolti si tengano allineati a vicenda per rimanere al corrente degli sviluppi pertinenti.

- **La continua attenzione all'eccellenza tecnica e alla buona progettazione esaltano l'agilità:**

Il fatto che vengano apportate modifiche continue, non solo in fase di sviluppo ma anche a progetto terminato fa capire come si cerchi continuamente una sorta di perfezione tecnica e di qualità assoluta.

- **La semplicità: l'arte di massimizzare la quantità di lavoro non svolto:**

Diventa rigorosa l'attenzione alle funzionalità che si ha intenzione di sviluppare nel corso del progetto. Dopo aver individuato quelle necessarie, è molto importante effettuare una 'pulizia' ed eliminare quelle che sono considerate superflue, per ridurre il carico di lavoro e per potersi concentrare solo su quelle primarie, ottenendo così una maggiore qualità.

- **Le architetture, la progettazione ed i requisiti migliori emergono da team che si auto-organizzano:**

La comunicazione interna è molto importante anche in sede di organizzazione. Infatti i team che sono in grado di auto-gestirsi in generale sono in grado di produrre una qualità superiore. Sono molto frequenti ed incentivate delle sessioni di Brainstorming per agevolare lo scambio di idee e di opinioni.

- **Ad intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza:**

Questa fase è chiamata Retrospettiva, qui i membri del team hanno una finestra temporale per riflettere su ciò che ha funzionato e ciò che invece non è andato per il verso giusto nel corso delle recenti iterazioni. L'esperienza passata è sempre un buon punto di partenza per migliorarsi e cercare di essere più efficienti quando si dovranno affrontare progetti futuri.

1.2.3 Team Agile

La struttura del team Agile è formata perché possa far fede ai fondamenti della metodologia stessa, cioè è progettata per essere flessibile e collaborativa, riuscendo a raggiungere un'efficacia particolarmente elevata.

Nella gestione di un progetto Agile si distinguono alcune figure principali e fondamentali.

il Product Owner è la figura più influente sulle decisioni di progetto ed ha alcune delle responsabilità tra le più importanti, tra cui la definizione e la prioritizzazione dei requisiti e delle funzionalità che devono essere sviluppate per la realizzazione di un prodotto. Si occupa di rappresentare gli stakeholder e quindi i loro interessi, operando come una sorta di 'controllore' nei confronti del team e assicurandosi che esso spenda la propria capacità lavorando sulle funzionalità di priorità maggiore. Ha come obiettivo principale quello di assicurarsi che il valore aggiunto del prodotto sia massimizzato, affiancato da una figura che lo assista per quanto riguarda l'organizzazione degli individui costituenti il team e del loro carico di lavoro, **Lo Scrum Master**. Esso ha un ruolo chiave nelle varie cerimonie che il team deve affrontare durante il corso del progetto, con la responsabilità di facilitare il processo e rimuovere eventuali ostacoli, assicurandosi che nessun elemento costituente la filosofia da seguire venga tralasciato. Il suo nome deriva da una delle principali strutture di Agile applicate al giorno d'oggi in contesti aziendali, l'omonimo framework Scrum. L'ultimo elemento è **il team di sviluppo** vero e proprio che si occupa della parte maggiormente operativa, cioè di sviluppare, testare e rilasciare i costanti incrementi del prodotto. Tutte queste figure sono necessarie al corretto svolgimento delle attività, e l'assenza di una di esse renderebbe estremamente difficoltoso riuscire a creare il maggior valore possibile.

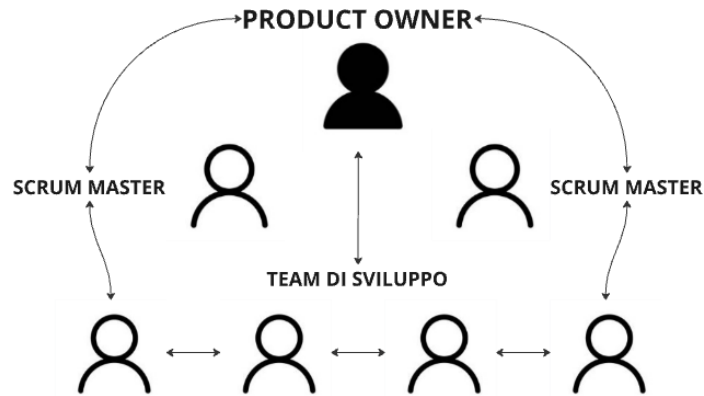


Figura 1.3: Immagine dello schema di una struttura tipica di Team 'Agile'

Vantaggi

- **Garanzia della qualità:** i continui cicli, garantiscono la possibilità di avere numerosi feedback e quindi di rispettare i requisiti di progetto soddisfacendo le esigenze del cliente; [7]
- **Flessibilità:** la natura della metodologia è perfettamente complementare con i progetti in continuo mutamento, assicurando un'ottima adattabilità in ogni situazione;
- **Collaborazione e comunicazione:** il lavoro di squadra favorisce, non solo la buona riuscita del progetto, ma contribuisce anche a creare un ambiente di lavoro ottimale e stimolante per la motivazione dei singoli individui.[7]
- **Riduzione dei rischi:** se fosse necessario effettuare delle modifiche in corso d'opera, sarebbe possibile senza causare una gran perdita, sia che si parli di costi, di tempi o di qualità del prodotto.
- **Riduzione dei tempi di sviluppo:** grazie al fatto che il procedimento sia composto da una lunga serie di piccole iterazioni, queste possono essere focalizzate dapprima sulle principali funzionalità del progetto riducendo il time to market, per poi scendere nel dettaglio dei requisiti meno importanti ma che comunque forniscono valore aggiunto al prodotto finito.
- **Maggior coinvolgimento degli stakeholder:** essendo, il processo, composto da frequenti interazioni, il team di sviluppo riesce a comprendere meglio le esigenze degli stakeholder tramite feedback continui, i quali vedono ogni stato del progetto in avanzamento. Si può dire che, quindi, ne sia sensibilmente migliorata la trasparenza.

Svantaggi

- **Dipendenza dal team:** il successo è fortemente dipendente dai membri della squadra. Se quest'ultima non è sufficientemente esperta o disciplinata, c'è la possibilità di incappare in situazioni di inefficienza e ottenere un output di qualità inferiore a quella ricercata. Inoltre, è necessario che ci siano svariate ed equilibrate competenze tra le persone che devono collaborare. [8]
- **Scarsa documentazione:** la capacità dell'Agile di essere molto adattivo e pensato per situazioni ad-hoc è sicuramente un benefit, però, è opportuno fare attenzione ad entrambi i lati della medaglia. La flessibilità prevede comunicazione 'faccia a faccia' tra gli individui e sconsiglia una documentazione dettagliata. Questo può causare problemi di diversa origine: diventa molto complicato inserire nuovi membri all'interno del team ed è, di conseguenza, una vera e propria sfida mantenere un prodotto a lungo nel tempo [11]
- **Difficoltà di gestione di tempi e costi:** La natura stessa del metodo rende difficile gestire le previsioni di costi e tempi. Quando si inizia un nuovo progetto, si cerca di sapere quanto le persone impiegate saranno impegnate dallo svolgimento, però a priori, è molto difficile capire l'effort necessario totale, essendo l'attenzione dei membri improntata sul futuro imminente e non su quello a lungo termine. Lo stesso discorso può essere applicato ai costi, i quali sono poco molto variabili anche in funzione degli eventuali cambiamenti o implementazioni da effettuare in corso d'opera. Per ridurre questo rischio, è opportuno scrivere i requisiti in maniera completa ed occuparsi prevalentemente di quelli ad alta priorità che corrispondono alle caratteristiche principali. [8]
- **Necessaria cultura aziendale:** nelle realtà in cui l'Agile non è presente, non è semplice e rapida la sua integrazione, poiché è necessario un cambio di mentalità da parte dei dipendenti interessati. Questo porterà via molto tempo e denaro speso in corsi di formazione all'azienda, e per riuscire ad essere efficaci, c'è bisogno di un periodo adeguato. Spesso, inoltre, alcuni individui legati alla tradizione perché indottrinati secondo quei principi, opporranno una certa 'resistenza', talvolta involontaria.

1.2.4 Metodologie ibride

La realtà dei fatti è diversa. Si è detto precedentemente che la diffusione della gestione dei progetti secondo una filosofia Agile sia relativamente recente. Questo implica che le aziende già esistenti abbiano dovuto affrontare un percorso di implementazione delle tecniche per poter essere capaci di adoperarle.

Il percorso non è né semplice, né breve e prevede l'impiego di non poche risorse. E' risaputo che le aziende non siano tutte uguali, e che non seguano gli stessi metodi o principi, per questo anche il percorso di transizione dei metodi di gestione è adattato da ogni impresa a se stessa nel modo più opportuno, con l'obiettivo di trarne il maggior valore aggiunto.

Proprio grazie alla flessibilità delle tecniche più recenti, sono nate delle forme di gestione Ibrida, in cui ogni realtà lavorativa ha assimilato solo alcuni dei principi cardine dell'Agile ed ha creato dei metodi separati.

I metodi di gestione Ibrida, quindi, uniscono framework come Scrum o Kanban con tecniche di gestione tradizionali e sono caratterizzati da benefici derivanti da entrambe le filosofie, quali: [9]

- **Processi su misura:** l'adattabilità alle specifiche di uno specifico progetto è notevole. Inoltre, la gestione è modificabile secondo i vincoli strutturali aziendali, riuscendo a trovare una struttura ottima;
- **Comunicazione e collaborazione:** La comunicazione molto frequente tra i membri del team unita ad uno scheduling dettagliato migliora la qualità delle informazioni scambiate tra i vari attori;
- **Controllo e prevedibilità:** L'utilizzo di una pianificazione iniziale di alto livello (derivante dalle tecniche tradizionali), permette di avere un quadro generale del progetto nel tempo, migliorando la scarsa capacità di identificare eventuali rischi del metodo Agile. Ciò permette all'azienda di mitigare, evitare o trasferire il rischio in tempo riducendo l'impatto del danno.

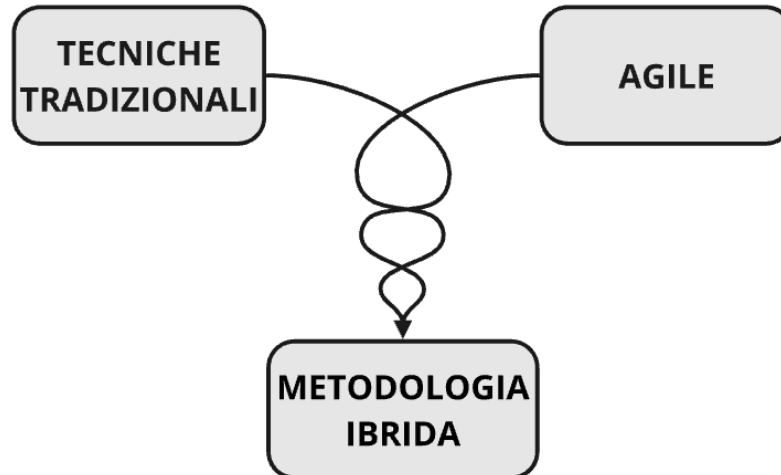


Figura 1.4: Immagine della formazione delle tecniche Ibride

Considerazioni sulle diverse metodologie

Chi si trova di fronte ad un progetto da gestire deve quindi essere in grado di capire, a seconda dei relativi fattori specifici e della organizzazione, quali criteri seguire come linee guida per un operato ottimale. La presenza di tecniche tradizionali è influenzata dal settore di appartenenza del progetto, nel mondo del Construction Project Management, per esempio, è più usuale utilizzare documentazione e pianificazione dettagliata guardando al progetto nella sua interezza.

Se si fa riferimento a progetti di sviluppo software, invece, sarebbe anche controproducente utilizzarle per via dell'incertezza e facilità di cambiamento dei requisiti del settore.

Nel mondo odierno, non si trovano molti progetti gestiti mediante processi puramente Agili, bensì la maggior parte è organizzata in maniera ibrida, per semplicità di adattamento, da parte delle aziende.

Alcune forme Ibride stanno crescendo e diventando sempre più diffuse. tra queste troviamo la **SAFe**, una struttura tipica del settore IT di grandi imprese, che combina i principi di Lean, Agile e DevOps (combinazione delle parole 'development' e 'operations').

Altri framework Ibridi possono essere: **ScrumBan** cioè un misto tra i due principali framework agili; **LeSS**, una particolare tipologia di Scrum su larga scala applicato a progetti che coinvolgono più team di sviluppo.

2 Maggiori framework Agile: Scrum e Kanban

In questa sezione, sono analizzate le due strutture identificate come principali, in termini di diffusione, della metodologia Agile.

Partendo dalle origini, sono poi illustrati gli strumenti e le ricorrenti cerimonie (se previste dalla struttura stessa), oltre che una serie di rispettivi pro e contro.

I due metodi, per quanto analoghi se si fa riferimento ai principi seguiti, differiscono per l'approccio di gestione del lavoro e dei processi.

2.1 Framework Scrum

E' sicuramente la struttura più diffusa nei contesti aziendali odierni. Se si vuol fornire una definizione questa può essere:

Definizione 2.1.1. *Scrum è un framework leggero che aiuta le persone, i team e le organizzazioni a generare valore attraverso soluzioni adattive per problemi complessi. [10]*

Scrum è particolarmente semplice, la teoria pertinente e la struttura stessa aiutano a raggiungere gli obiettivi di progetto e a creare valore.

Una delle più importanti caratteristiche è quella di essere costituito, non solo dalle sue linee guida, ma anche dall'intelligenza delle persone che lo utilizzano. Infatti, non fornisce indicazioni dettagliate volontariamente, in modo che i fruitori possano adattarlo come meglio credono e alle proprie interazioni e relazioni interne.

All'interno di Scrum si possono combinare varie tecniche, processi e metodi, rendendo visibile l'efficacia della gestione presente in un determinato momento, in modo da individuare ed apportare eventuali miglioramenti.

I team di lavoro coinvolti hanno collettivamente tutte le competenze necessarie allo sviluppo del progetto e potrebbero acquisirne altre se lo svolgimento lo richiedesse.

Per poter applicare appropriatamente lo Scrum è necessario fare affidamento all'esperienza empirica del contesto al quale si vuole implementare. Inoltre, è opportuno seguire una linea d'azione 'lean' ovvero concentrarsi solo sull'essenziale riducendo gli sprechi. Sono incorporate alcune formalità, ovvero degli eventi ricorrenti, che funzionano perché rispecchiano i pilastri di Scrum: trasparenza, ispezione e adattamento.

2.1.1 Origini della metodologia

Dopo essere state analizzate le radici di pensiero di Scrum, verranno ora presentate le origini.

Le prime idee rese pubbliche risalgono alla seconda metà del XX secolo, più precisamente tra il 1980 e il 1990, quando fu pubblicato un articolo dai due studiosi giapponesi Hirotaka Takeuchi e Ikujiro Nonaka, nel quale è descritto un approccio di sviluppo prodotto ispirato al Rugby (da qui il termine Scrum, che significa mischia), dove il collettivo lavora come una singola unità. Loro descrissero lo svolgimento del lavoro come attività svolta da piccoli team multidisciplinari che collaborano per il raggiungimento delle consegne. [12]

Proseguendo con gli anni, entrando nel decennio del 1990, Jeff Sutherland, un informatico veterano dell'aviazione, prendendo spunto dall'articolo dei due studiosi giapponesi, ha cominciato a sviluppare delle linee guida per poter applicare questi principi allo sviluppo software. Fino a che nel 1995, insieme a Ken Schwaber pubblicò un documento, presentato formalmente alla conferenza OOPSLA (Object-Oriented Programming, System, Language and Applications), che stabilì le basi dell'intera metodologia.

Nel 2001 viene pubblicato il 'Manifesto Agile', redatto dai 17 pionieri, ovvero un documento che definisce i valori e i principi fondamentali dello sviluppo Agile. Generalmente, approcciando agli anni 2000 si nota un aumento della letteratura riguardante Scrum.

Analisi della letteratura

Questo fatto è messo in evidenza tramite una ricerca su Scopus, uno dei più grandi e rinomati database di documentazione scientifica, essenziale per ricercatori accademici e professionisti di varie discipline.

Effettuando la ricerca per Titolo, Abstract e Keyword, inserendo nel prompt solo la parola 'Scrum framework', sono risultati disponibili 294 documenti. Precisando che i dati ricavati dalla risorsa elettronica non sono antecedenti al 2006 e attuando un focus sulla data di pubblicazione, si nota un trend crescente nel numero di testi pubblicati nel corso degli anni:

Documenti per anno

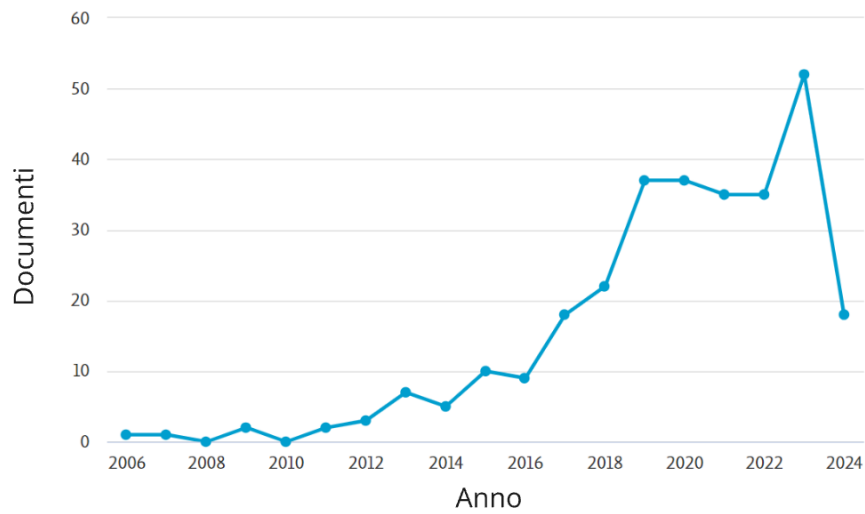


Figura 2.1: Grafico del numero di documenti relativi al framework Scrum pubblicati per anno

Un altro dato, importante da visualizzare è quello del numero di documenti raggruppati per tipologia. Il grafico precedente e la natura dell'argomento stesso, ci suggeriscono che lo studio in materia sia relativamente 'giovane' e che quindi la documentazione esistente non sia composta prevalentemente da libri, bensì da articoli e conference paper.

Documenti per tipo

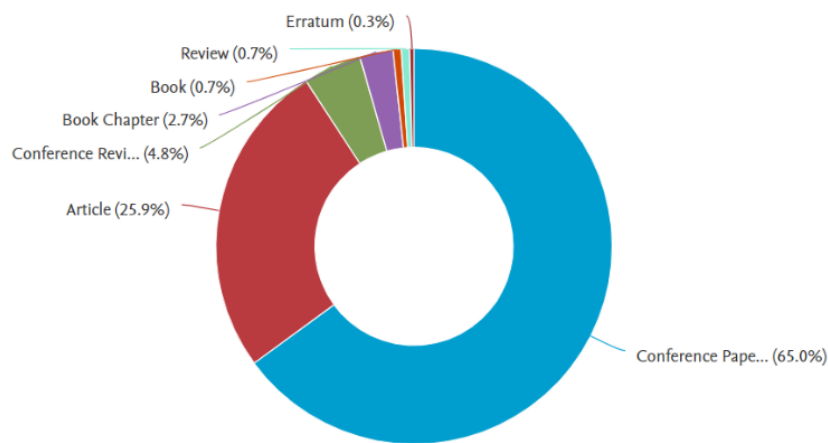


Figura 2.2: Numero dei documenti per tipo

2.2 Principali strumenti Scrum

2.2.1 Sprint

Lo Sprint è l'entità fondamentale dello Scrum, in cui le idee sono trasformate in valore. E' un'unità di lunghezza prefissata che generalmente va dalle due alle quattro settimane. All'interno di questo periodo vengono iniziate una serie di attività relative al progetto, con l'obiettivo di riuscire a concluderle entro il termine stabilito. Gli Sprints di progetto sono consecutivi, non appena se ne conclude uno, ne comincia un altro. Durante questo lasso temporale vengono svolte anche le cerimonie relative alla gestione, tipiche di Scrum.



Figura 2.3: Rappresentazione dei cicli Sprint consecutivi

Durante la singola iterazione ci sono alcuni punti fondamentali di cui tenere conto:

- Non sono fatti alcuni cambiamenti che possano mettere in pericolo lo Sprint Goal;
- La qualità non diminuisce;
- I requisiti del prodotto possono essere modificati, in seguito ad una rinegoziazione con il Product Owner dovuta al chiarimento dello scopo.

La struttura degli Sprint permette di migliorare la prevedibilità, mettendo in evidenza il Product Goal almeno una volta al mese. Per questo motivo è importante non programmare delle iterazioni di durata eccessiva, che porterebbero con sé il rischio di perdere di vista l'obiettivo.

2.2.2 Product Backlog

Il Backlog di prodotto, è un componente fondamentale del framework Scrum. Consiste in un elenco ordinato di tutte le caratteristiche, funzionalità, requisiti, miglioramenti e correzioni necessari alla realizzazione del prodotto. Gli elementi costituenti l'elenco sono caratterizzati da alcuni attributi primari come: la descrizione, che è utile per rendere chiaro e comprensibile il requisito da implementare; l'ordine, in termini di priorità; la stima dell'effort, cioè quanto tempo si pensa di impiegare allo svolgimento di una determinata attività.

Per una questione di convenzione, gli elementi ordinati in alto sono sempre più chiari e meglio dettagliati rispetto a quelli in basso.

La persona responsabile della stesura del Product Backlog e della prioritizzazione delle attività in esso contenute, è il Product Owner, il quale comunque avrà un supporto dalle varie figure del team nella fase di affinamento.

E' lo Scrum Team a decidere, in base alle esigenze, quando il raffinamento sia completato e nella maggior parte dei casi questa attività non occupa più di un decimo del tempo totale di sprint. L'elenco in questione diventa sempre più ampio, esaustivo e di valore, quando il prodotto di riferimento è in uso e si continuano a ricevere feedback costruttivi dal mercato. Si può dunque definire come un 'artefatto vivente', in quanto non smette mai di cambiare. [10]

Elementi del product backlog

Durante la gestione di un progetto si creano degli items che hanno validità e importanze diverse. Ogni item corrisponde ad una attività, una caratteristica o una funzionalità che deve essere sviluppata in ottica di concludere un progetto.

Esistono items di alto livello e di dettaglio, che sono differenziati prevalentemente per la mole di lavoro da svolgere contenuta all'interno.

Ogni organizzazione che opera seguendo un framework Scrum sceglie la struttura del Product Backlog che ritiene più adatta, utilizzando solo gli item ritenuti necessari e utili al corretto svolgimento delle attività, mantenendo una gerarchia tra i vari livelli.

Ecco gli items più utilizzati all'interno di un Backlog di prodotto, secondo un framework Scrum:

- **Epic:** Rappresenta un obiettivo di alto livello, il più alto analizzato. E' un'iniziativa che per essere completata ha bisogno di moltissimo tempo (anche anni) e quindi, di conseguenza, è sviluppata nel corso di molti Sprint. Un esempio può essere la creazione di un nuovo software.
- **Feature:** Analizzando un livello più nello specifico, troviamo le Features, cioè le singole unità che compongono una Epic. Queste possono essere viste come un requisito significativo di prodotto che rappresenta una funzionalità completa o un set di funzioni correlate. Un esempio può essere l'implementazione di una funzionalità di autenticazione di un utente all'interno di un software.

- **PBI:** Ad ogni Features sono associati dei Product Backlog Items (PBI), ovvero dei requisiti specifici, dei miglioramenti o delle funzionalità dettagliate che devono essere sviluppati. Sono molto importanti perché sono prioritari per il team e dovrebbero essere schedulati e completati all'interno del singolo Sprint. Un esempio può essere la richiesta di un utente di avere la possibilità di recuperare le credenziali tramite mail.
- **Task:** Il Task è l'unità di lavoro più piccola che viene presa in considerazione e il suo effort è misurabile nell'ordine di grandezza delle ore e non delle settimane. Rappresenta uno specifico passo da compiere per riuscire a sviluppare un PBI. Si tratta di una piccola unità, facile da assegnare ad un membro del team, costituita da una mole di lavoro tale da rendere fattibile il suo svolgimento in autonomia.
- **Bug:** Rappresenta un malfunzionamento, un errore, o un difetto del software che deve essere risolto per garantire qualità all'utilizzatore. Generalmente un Bug ha la stessa grandezza di un PBI, ed è quindi assegnato ad un singolo individuo e non ad un team intero. Possono essere dedicati task specifici per semplificare il lavoro. Un esempio può essere il non-funzionamento di un bottone all'interno di una pagina web.
- **Evolutiva:** le Richieste di Evolutiva, sono modifiche al progetto che migliorano le funzionalità già esistenti oppure aggiungono nuove capacità, senza però modificare l'obiettivo principale del progetto. Anche in questo caso la granularità è simile a quella del PBI, e, così come nel caso dei Bug, possono essere creati dei task dedicati di suddivisione del lavoro.

Quelli elencati non sono tutti i tipi di work item esistenti, infatti ne esistono molti altri che possono essere usati per diverse funzioni, tra i quali: i ticket, le issues, i document, gli impediment o i test case.

Tutti gli elementi citati, se si guarda alla piramide di granularità, sono sullo stesso piano dei task, e sono quindi generalmente legati come dei 'child' al PBI. Possono essere rappresentati gerarchicamente nel modo illustrato nella pagina successiva:

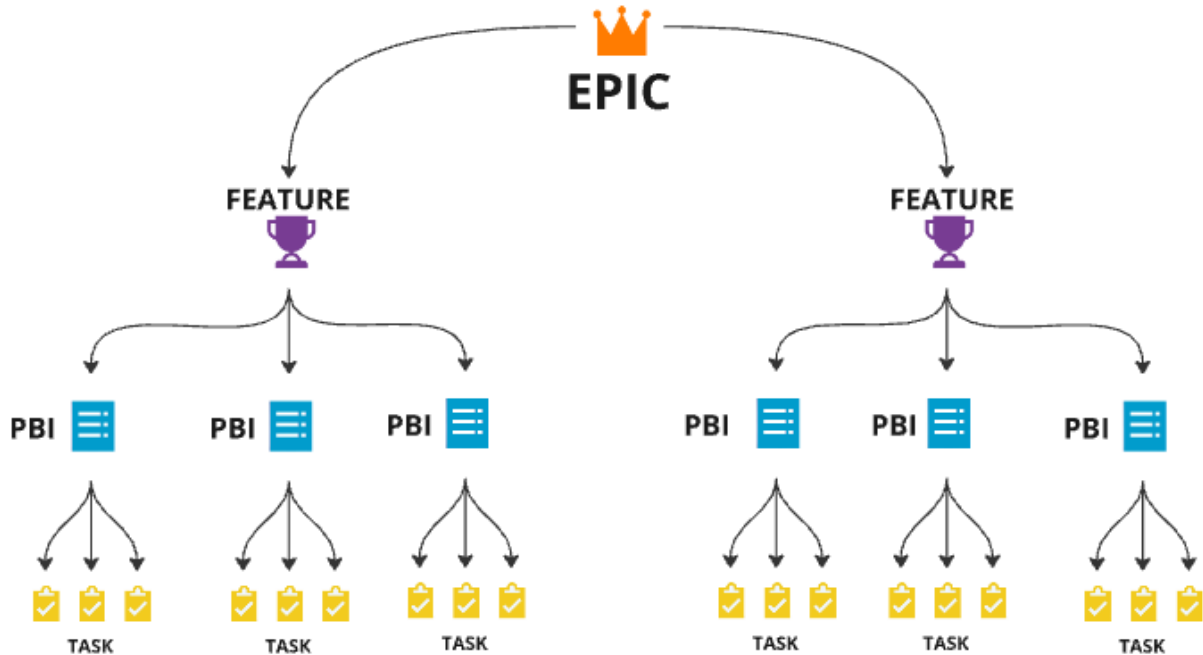


Figura 2.4: Rappresentazione della gerarchia degli item di progetto

Sprint backlog

Lo Sprint Backlog è essenziale per il corretto funzionamento della struttura Scrum.

Rappresenta l'elenco degli elementi pescati dal Backlog di prodotto con lo scopo di essere svolti nell'arco di durata dello Sprint corrente.

Gli elementi in questione vengono scelti dal Product Owner insieme al team di sviluppo in base alla priorità e alle varie esigenze di sviluppo e di capacità. Generalmente vengono inseriti i Product Backlog Items, i quali portano con sé una serie di unità di minor dimensione, i Task.

E' fondamentale che quest'ultime attività inserite siano molto brevi (tendenzialmente risolvibili in un giorno lavorativo) per non rappresentare un ostacolo troppo grande allo svolgimento dello sprint.

Lo Sprint Backlog è inoltre modificabile nell'arco temporale dello Sprint, sia per aggiungere eventuali attività impreviste con alta priorità, sia per modificare lo stato di avanzamento dei singoli task e rendere il progresso molto fedele alla realtà, agevolando chiunque abbia bisogno di esserne al corrente.

2.2.3 Monitorare l'avanzamento in uno Sprint

Il monitoraggio dell'avanzamento delle attività previste in uno Sprint è, come detto precedentemente, importantissimo per capire se il Team stia procedendo nella direzione giusta e per capire se siano nati dei problemi in corso d'opera. Riconoscere subito i problemi può avere grandissimi vantaggi in termini di tempo sulla risoluzione, senza compromettere l'avanzamento generale del progetto. Di seguito i principali metodi utilizzati nella fase di monitoraggio:

Burndown Chart

Il Burndown Chart è un grafico che mostra sull'asse delle ordinate la quantità di lavoro da svolgere, e sull'asse delle ascisse il tempo rimanente all'interno dello Sprint. Iniziando da un punto massimo, si sottrae giornalmente la quantità di lavoro svolto per giungere (teoricamente) alla fine delle settimane prestabilite con una quantità di lavoro restante pari a zero.

Capita frequentemente che a causa di imprevisti rimanga del lavoro al termine, che verrà dunque preso in carico nell'iterazione successiva. Ecco qui un esempio:

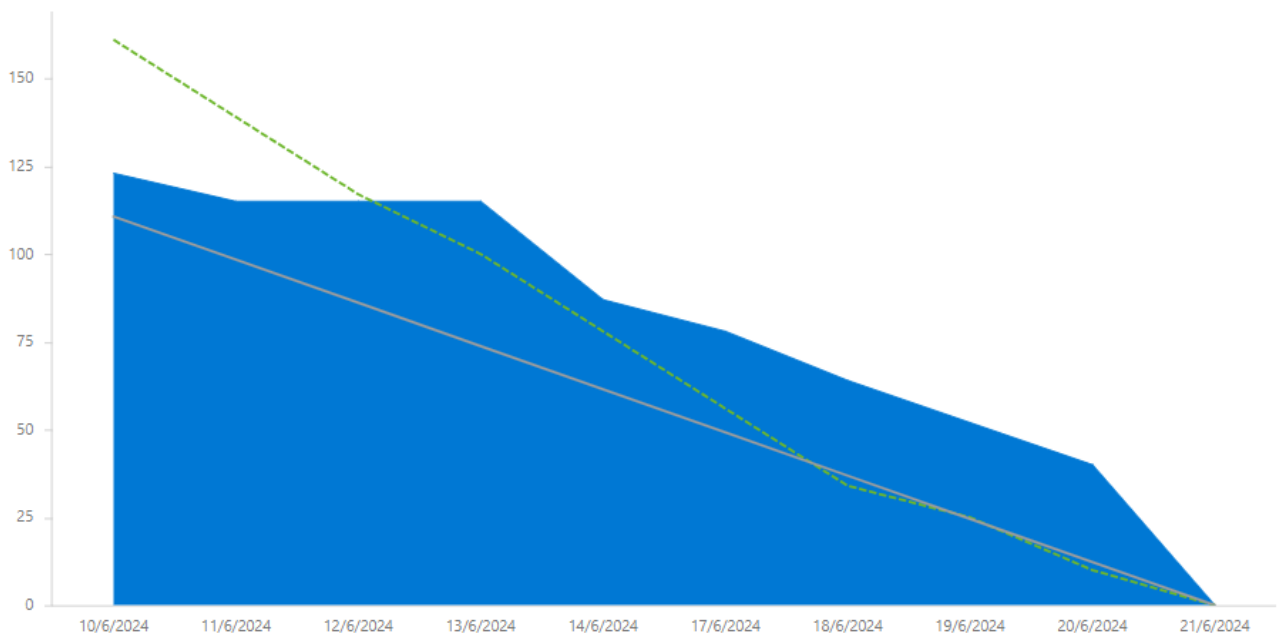


Figura 2.5: Rappresentazione di un grafico Burndown

Sull'asse delle ordinate troviamo la capacità in ore, mentre sulle ascisse i giorni appartenenti ad una determinata iterazione. La linea verde rappresenta la capacità totale rimanente, la linea grigia corrisponde all'andamento ideale e l'area blu è la capacità necessaria a svolgere i task schedulati all'interno dell'iterazione. Come ci si aspetta, man mano che le attività vengono svolte e quindi 'chiuse' l'area blu diminuisce, fino a raggiungere lo zero in concomitanza con l'ultimo giorno di Sprint. Questo è un caso particolare in cui l'area blu rimane al di sopra della linea verde per buona parte della durata totale. Ciò non succede perché è stato schedulato troppo lavoro per la durata dello Sprint, ma è dovuto semplicemente al ritardo dei membri del team nell'aggiornare sul software le attività svolte.

Velocity Chart

Il Velocity Tracking è un altro strumento utile a controllare come lo Sprint stia procedendo, permettendo di avere in risalto la quantità di work item completata in diverse iterazioni. Aiuta a capire come è sfruttata la capacità produttiva ed è di supporto per la pianificazione degli Sprint successivi.

Con il termine 'velocità' si intende proprio il numero medio di work items svolti da parte di un team nel corso degli ultimi periodi. Qui è riportato un esempio:

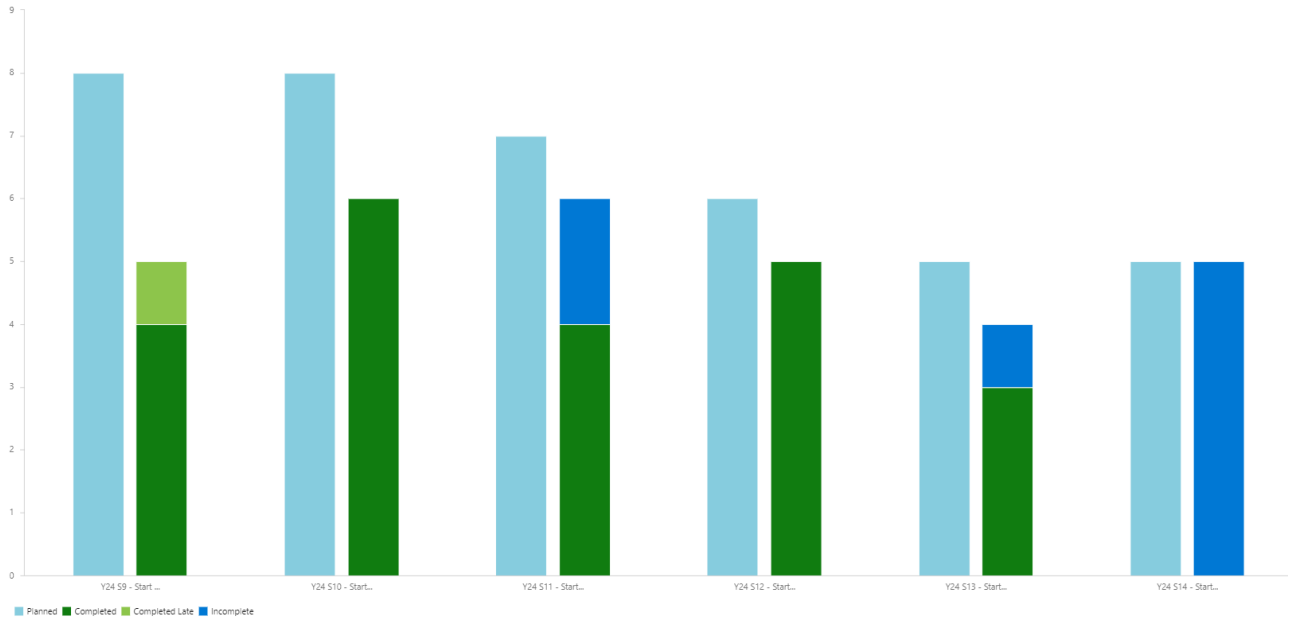


Figura 2.6: Rappresentazione di un grafico Velocity

Sull'asse verticale troviamo il numero di work items, mentre su quello orizzontale sono riportati i diversi Sprint.

In questo grafico, possiamo distinguere sei cluster di colonne, ognuno di questi relativo ad un diverso periodo. Sono presenti quattro diversi colori: azzurro, che rappresenta il numero di work items programmati per le due settimane dedicate; verde scuro, per rappresentare il lavoro completato; verde chiaro, per mostrare il lavoro completato in una data successiva a quella di scadenza; blu scuro, che mostra il numero di Work items che, per un qualunque motivo, non siano stati lavorati.

Cumulative Flow Diagram

Un Cumulative Flow Diagram (CFD) corrisponde ad uno strumento visivo utilizzato in sistemi di gestione Agile con lo scopo di monitorare l'avanzamento dei lavori nel tempo, fornendo una panoramica del lavoro da svolgere, in corso e completato. Ecco un esempio preso da Azure DevOps:

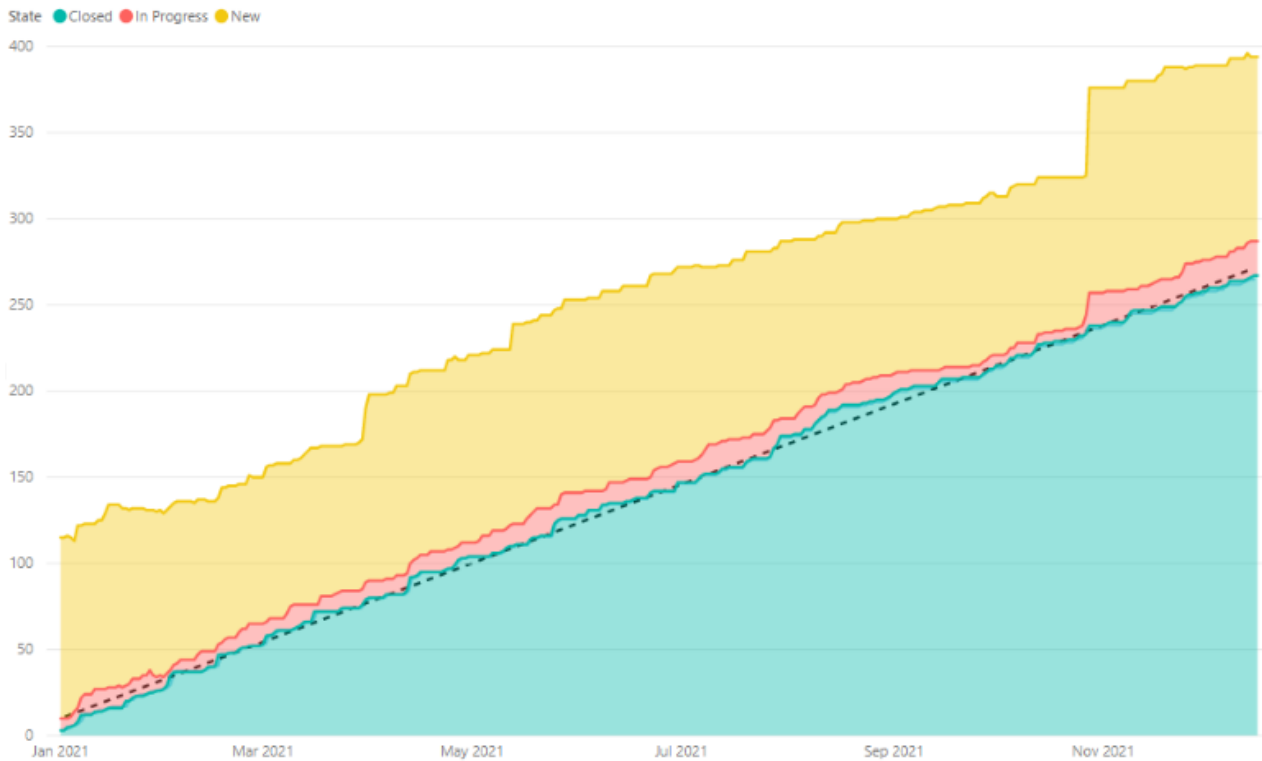


Figura 2.7: Rappresentazione di un CFD

Il caso riportato, è un caso esemplificativo molto semplice, in cui sono distinti solamente tre stati per i work items: New, In Progress e Closed. Quando si affronta un progetto reale, nella maggior parte dei casi si trovano molti più stati corrispondenti alle varie fasi alle quali l'item è sottoposto durante il suo ciclo vita.

Questo tipo di visualizzazione è molto utile per svariati motivi, tra cui:

- Una facile visualizzazione dei colli di bottiglia: l'ispessimento di una delle bande nel corso del tempo indica un accumulo di Work Items in un determinato stato, evidenziando la necessità di una miglioria per far proseguire al meglio il lavoro.
- Una ottima capacità di previsione perché, semplicemente analizzando le tendenze storiche, è possibile dedurre (anche se non alla perfezione) una ipotetica data di completamento delle attività.

2.3 Principali cerimonie Scrum

Come già detto in precedenza, per un'ottimale gestione Agile secondo i principi del framework Scrum, è necessario sottostare ad alcune formalità.

Queste non sono altro che alcune cerimonie ricorrenti (quattro principali), con lo scopo di far allineare ogni elemento del team in merito ai progressi raggiunti, al controllo del lavoro svolto, alla programmazione del lavoro futuro ed alla libera espressione delle proprie opinioni, esprimendosi sul team, sul progetto o sulle modalità di esecuzione.

Ora le quattro cerimonie principali saranno analizzate in ordine sequenziale così come vengono affrontate durante lo Sprint.

2.3.1 Sprint Planning

Lo Sprint Planning è la prima fase dello Sprint, ed è antecedente all'inizio, in quanto senza l'outcome di questa cerimonia non si avrebbe nemmeno idea della mole di lavoro da svolgere.

Consiste in una pianificazione delle attività da parte dell'intero Scrum Team. Generalmente questa cerimonia ha una durata relativamente breve, tant'è che a livello teorico si consiglia di riservare un box temporale di massimo 8 ore per uno Sprint della durata di un mese. Nella realtà però, la maggior parte delle volte l'evento dura molto meno della durata citata, e il suo tempo di esecuzione si aggira intorno ai 60 minuti.

Questo tempo è influenzato da diversi fattori, quali l'organizzazione del team in vista della cerimonia, la bravura dello Scrum Master nel gestire la riunione, la chiarezza delle attività da svolgere da parte del Product Owner e la familiarità dell'intero team con la metodologia stessa. [10]

In generale uno Sprint Planning risponde a queste tre domande:

1. Qual è lo Sprint Goal?
2. Quale incremento può essere consegnato nell'iterazione?
3. Come si organizzerà il Team per riuscire a svolgere il lavoro necessario?

Rispondendo a queste generiche domande si può considerare uno Sprint Planning concluso.

Sprint Goal

Lo Sprint Goal è l'obiettivo che il team si impegna a raggiungere durante lo Sprint, e viene aggiornato di iterazione in iterazione a seconda dell'incremento raggiunto. Viene creato nello Sprint Planning e fornisce una sorta di guida al team nel corso del lavoro. Gli sviluppatori lavorano tenendo costantemente a mente il Goal da raggiungere, implementando le tecnologie e le funzionalità necessarie per poter considerare i compiti svolti un successo.

Se il lavoro si rivelasse diverso da quanto stabilito, sarebbe compito del team collaborare con il Product Owner per rinegoziare e riadattare l'ambito del Backlog durante lo Sprint.

Nello schema sottostante sono presentati alcuni consigli per far sì che una sessione di Sprint Planning sia completata con successo e non risulti inconcludente:

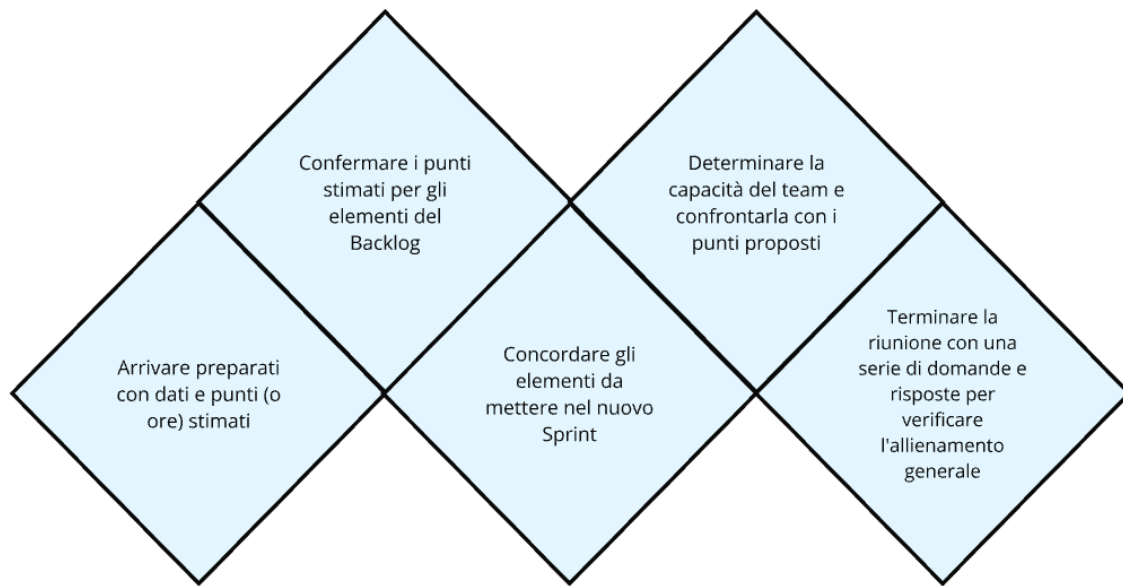


Figura 2.8: Rappresentazione schematica di 'tips' per lo svolgimento di uno Sprint Planning

2.3.2 Daily Scrum

Il Daily Scrum è un'attività (come evidenziato dal nome stesso) giornaliera, della durata massima di mezz'ora, alla quale partecipa il team di sviluppo per programmare le attività da svolgere nel corso delle 24 ore successive alla riunione.

Per questione di semplicità ed organizzazione, si cerca di tenere il meeting con una modalità (remoto o presenza), un luogo e un orario costanti.

In generale, è possibile affermare che la riunione serve a monitorare lo stato di avanzamento verso lo Sprint Goal.

Per condurre appropriatamente un Daily Scrum è consigliato seguire un'insieme di linee guida, secondo le quali ogni membro del team dovrebbe porsi 3 domande: [10]

1. Cosa ho fatto nella giornata passata per procedere verso lo Sprint Goal?
2. Cosa farò oggi per aiutare il Team a procedere verso lo Sprint Goal?
3. Ho notato la presenza di ostacoli al progresso dello Sprint?

Esistono alcune strutture che è opportuno non modificare in modo che la riunione sia svolta nel migliore dei modi e sia quindi efficiente:

- Rimanere nello slot di tempo prestabilito;
- Utilizzare un supporto visivo;
- Focalizzarsi sulla collaborazione coinvolgendo tutti i membri del team.

2.3.3 Sprint Review

La Sprint Review può essere considerata come una sorta di resoconto dello Sprint passato, perciò la riunione è schedulata al termine di ogni iterazione.

Durante questo incontro, il team si riunisce insieme allo Scrum Master per ispezionare il prodotto sviluppato nelle precedenti settimane e discutere dei metodi utilizzati.

La cerimonia ha una durata prevista di massimo due ore, ma viene poi adattata alle esigenze e alla mole di lavoro presente.

Generalmente è strutturata nel seguente modo:

dopo essersi radunati tutti i partecipanti (che sia tramite l'utilizzo di piattaforme online o fisicamente), lo Scrum Master sancisce l'inizio del meeting ed essendo munito dello Sprint Backlog passa in rassegna tutte le attività dell'iterazione trascorsa, rivolgendosi direttamente all'assegnatario del Task in questione. Ad esso viene chiesto un aggiornamento in merito alla percentuale di avanzamento, all'effort impiegato (in termini di ore lavorative) e alla presenza di eventuali difficoltà riscontrate nell'operare.

Se l'attività in oggetto è stata conclusa, si aggiorna lo stato in 'Closed' e si cataloga come compiuta. Altrimenti, dopo aver segnato le ore impiegate nello Sprint ormai terminato, se ne crea una copia che viene portata allo Sprint successivo.

Ciò, idealmente, non dovrebbe accadere poiché si sa che nella singola iterazione debbano essere schedulate solo attività con una durata relativamente breve, da completare nel time-box prestabilito.

In realtà, purtroppo, questo non sempre accade ed è necessario svolgere una singola attività nel corso di più sprint.

Un tassello fondamentale per la buona riuscita di una Sprint Review consiste nella partecipazione attiva del Product Owner, il quale può dare delucidazioni in merito alle eventuali difficoltà e può prendere visione del lavoro svolto dai propri dipendenti nelle ultime settimane in modo da avere un quadro completo della situazione e poter organizzare al meglio l'iterazione successiva.

2.3.4 Sprint Retrospectives

La Sprint Retrospective è una cerimonia per certi versi simile alla precedente, in quanto anch'essa si svolge al termine dello Sprint stesso. A differenza della Review, che ha come principale focus il prodotto, questa si concentra sui processi, la comunicazione e più in generale le dinamiche di squadra, perciò anche in questo caso è necessaria la presenza dell'intero team e del Product Owner.

Il meeting ha una durata di circa un'ora ed è suddiviso in tre parti sequenziali. Utilizzando un tool grafico di supporto, lo Scrum Master dà inizio alla cerimonia raccogliendo i pareri dei partecipanti in merito allo Sprint appena concluso.

I partecipanti, infatti, hanno a disposizione circa 15 minuti per esprimersi liberamente in modo anonimo, scrivendo su dei post-it virtuali ciò che pensano sia andato bene e che cosa invece sia da cambiare. I commenti possono spaziare tra diverse tematiche, come l'organizzazione del lavoro, la chimica di squadra, la mole di lavoro o la chiarezza delle funzionalità da sviluppare.

Dopo aver terminato la prima parte segue la fase di votazione, dove ognuno dei partecipanti ha a disposizione 5 voti da assegnare ai post-it con i quali si trova più

in accordo. (possono essere anche assegnati 5 voti allo stesso, in modo che risalti di più nelle fasi successive).

Segue poi la cosiddetta 'clusterizzazione'. Qui, il team di sviluppo e lo Scrum Master rileggono tutti i post-it insieme con il fine di formare diversi gruppi per argomento trattato ed essere maggiormente organizzati per l'ultima parte, quella dell'individuazione delle action.

Si iniziano a scorrere tutti i post-it raggruppati in ordine di punteggio, dal più alto al più basso, e per ogni tema si discute per riuscire a formulare delle azioni da intraprendere nello sprint successivo per migliorare gli aspetti negativi individuati.

2.4 Framework Kanban

La seconda struttura Agile per lo sviluppo di progetti Software più famosa in ordine di diffusione a livello globale è proprio quella Kanban. Non esiste una vera e propria definizione relativa alla metodologia, una che esprime al meglio il concetto, anche se a livello generale, può essere:

Definizione 2.4.1. *Kanban è un metodo per la gestione del flusso di lavoro che utilizza una rappresentazione visiva per gestire i processi e ottimizzare l'efficienza. [1]*

La sua struttura utilizzata nell'ambito dell'Information Technology, rappresenta proprio quelli che sono i fondamenti della filosofia Kanban. A livello pratico, il lavoro è rappresentato da "card". Ognuna di esse rappresenta un identificativo per i PBI esistenti, mostrando il codice, il nome, l'assegnatario e il monte ore previsto per il suo completamento.

La Kanban Board, invece, consiste in una lavagna suddivisa in colonne, ognuna delle quali rappresenta una fase del processo, alcuni esempi di colonne possono essere: "New", "Waiting for Technical Analysis", "Doing", "Waiting for Test", "Test", "Done", ma sono comunque personalizzabili in base alle esigenze in modo da poter monitorare al meglio lo stato di un progetto. Per ognuna di esse è fissato un numero massimo di Item che possono esserne parte, in modo da non sovraccaricarsi e creare confusione.

In base alla capacità dei membri del team, un numero di card viene messo in circolazione e prioritizzato. Ogni carta agisce come un meccanismo di segnalazione, comunicando in che stato si trovi un determinato PBI. La card è libera di muoversi tra le varie colonne seguendo l'andamento reale muovendosi nella Board.

Quando non ci sono più slot liberi, non è più possibile avviare un nuovo lavoro e ogni nuovo PBI deve attendere in coda finché non si libera uno spazio.

Quando un lavoro viene completato, libera della capacità utile ad iniziare un altro compito che era in attesa di essere svolto. Questo meccanismo è noto come sistema "Pull".

2.4.1 Origini della metodologia

Il "padre fondatore" di Kanban è riconosciuto essere l'ingegnere Giapponese Taiichi Ohno, il quale negli Anni '40 decise di mettere alla prova un metodo di produzione sperimentale, ispirandosi a come venivano riforniti gli scaffali dei supermercati. Fu il primo che riuscì ad adattare questo modus operandi a scopi industriali, rendendolo parte del "TPS" (Toyota Production System) con il fine di ottimizzare la gestione dell'inventario e migliorare l'efficienza della catena di produzione, mirando alla riduzione degli sprechi e fornendo un gran vantaggio all'azienda. [5] Con il passare degli anni, fu poi David J. Anderson, che negli ultimi anni del XX secolo fu in grado di adattare il metodo alle aree di produzione e sviluppo Software.

Analisi della letteratura

Per confermare quanto detto sui tempi dell'evoluzione della metodologia, osserviamo la storia della letteratura e quando hanno iniziato ad intensificarsi i documenti pubblicati.

E' sufficiente effettuare una ricerca mirata su Scopus (banca dati online) per titolo, abstract e keywords per identificare i documenti scientifici pubblicati nel corso degli anni.

In questo caso il prompt di ricerca utilizzato è il seguente:

'TITLE-ABS-KEY (("Kanban framework") OR ("Kanban method") OR ("Kanban system") OR ("Kanban software development") OR ("Kanban Agile")'

Risultano 703 documenti emessi nel corso degli ultimi cinquant'anni, distribuiti in maniera non costante con un evidente trend ascendente. Qui riportato il grafico dei rilasci nel tempo:

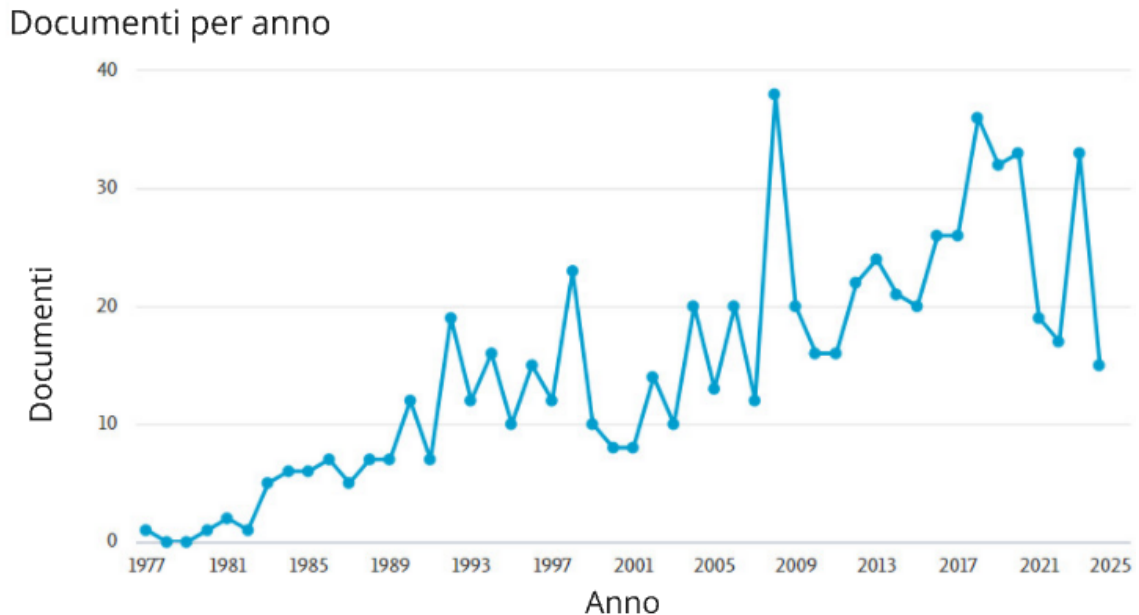


Figura 2.9: Rappresentazione del numero di documenti relativi a Kanban pubblicati per anno

Il grafico conferma quanto detto, non vi era una grande diffusione e conoscenza prima che il metodo venisse applicato anche nelle aree IT delle aziende.

I documenti presenti sono prevalentemente statunitensi, giapponesi e cinesi e relativi all'ambito dell'ingegneria.

Si può notare, dal seguente grafico, come la composizione della letteratura sia molto simile a quella relativa al framework Scrum, con prevalenza di Articoli e Conference Papers, cioè documenti prevalentemente accademici tipicamente redatti in merito a nuove scoperte, idee o teorie.

Documenti per tipo

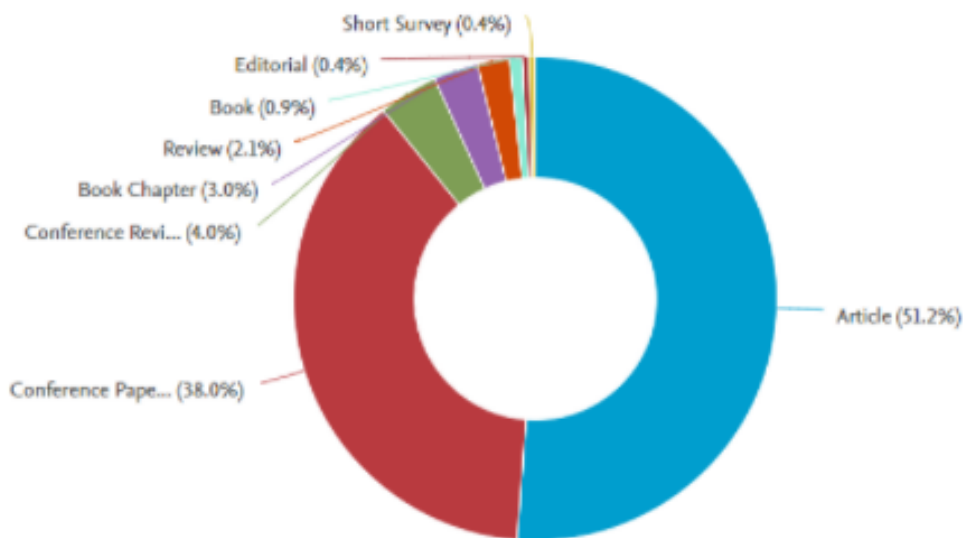


Figura 2.10: Rappresentazione del numero di documenti relativi a Kanban pubblicati per tipo

2.4.2 Strumenti e cerimonie

L'approccio in questione, quando applicato in ambito di sviluppo software, non è caratterizzato da un piano prefissato di meeting e ricorrenze, ma è molto flessibile e si adatta di giorno in giorno a quelle che sono le disponibilità delle risorse e le necessità del team per avanzare incrementalmente verso l'obiettivo prefissato.

L'utilizzo di questi metodi è consigliabile quando lo scopo del team di sviluppo è caratterizzato da una forte incertezza come può essere, per esempio, un'applicazione nuova o un nuovo sito internet. In questi casi le funzionalità da sviluppare sono mutevoli rendendo dunque determinante un'ottimale organizzazione del lavoro.

Infatti, spesso dopo aver implementato una qualsiasi caratteristica, può essere necessario applicare delle modifiche o sostituirla, in base a test interni o feedback da parte degli utilizzatori.

Avvalersi di una Board quindi, garantisce la trasparenza del lavoro svolto e da svolgersi per ogni risorsa impiegata e, in caso di modifiche successive ad un primo rilascio, permette di inserire nuove card senza creare confusione o ritardi di consegna tra le altre funzionalità da sviluppare.

Per chiarire il concetto, si può paragonare lo sviluppo di un progetto IT ad un puzzle, considerando le tessere come dei task di sviluppo. Se si seguisse un classico metodo a cascata, si dovrebbe iniziare a comporre da un angolo (corrispondente alle fasi preliminari del progetto) e si procederebbe andando a collocare le tessere al primo pezzo, preso da riferimento, in ordine sequenziale.

Seguendo una filosofia Agile, la risoluzione sarebbe differente. Si dovrebbero cercare tessere simili per colore o soggetto rappresentato, creando una molteplicità di punti di riferimento (macro-componenti del progetto) e si procederebbe collocando le tessere al proprio riferimento, indipendentemente l'una dall'altra.

Nel caso in cui sia necessaria una modifica dovuta ad un errore.

Con il primo metodo questa risulterebbe molto dannosa perché sarebbe necessario distruggere tutto il lavoro successivo all'errore stesso.

Nel secondo caso invece, l'errore comporterebbe meno danni, in quanto, avendo più riferimenti indipendenti il lavoro da distruggere sarebbe limitato solamente ad uno di essi.

2.5 Confronto tra i diversi framework

Per capire a fondo le differenze tra le diverse tecniche è opportuno che esse vengano analizzate in parallelo, confrontando i principali aspetti. Sia Scrum che Kanban sono metodi utilizzati per risolvere dei problemi in ambito di gestione progettuale, fanno entrambi parte della 'famiglia' Agile, ma differiscono per caratteristiche e contesti di applicazione. Sorvolando la diversa origine dei metodi, si focalizza l'attenzione sulle discrepanze percepibili dal punto di vista operativo.

Infatti, una delle principali differenze consiste proprio nella **struttura del lavoro**. In un caso si avanza incrementalmente seguendo la cadenza degli Sprint, nell'altro invece si ha un flusso di lavoro continuo gestito direttamente dal team interessato.

Da questa sostanziale differenza deriva una separata **pianificazione del lavoro** che, nel caso di Scrum, risulta essere effettuata a priori e molto chiara, mentre per Kanban, essendo lecita la modifica del carico di lavoro in corso d'opera, può essere definita flessibile, ottimale per progetti in cui si ha un alto grado di incertezza nei quali è necessario essere in grado di rispondere rapidamente al cambiamento.

Per adattarsi al meglio alla diversa conformazione delle due metodologie, si utilizzano separati **strumenti di visualizzazione del lavoro**. Secondo l'omonimo metodo, la nota Kanban Board è sostanzialmente l'unico tool utile a visualizzare e tenere sotto controllo l'avanzamento delle mansioni di progetto, essendo il WIP limite delle colonne il vincolo da rispettare. Scrum invece, utilizza una board differente, dove gli stati possibili sono ben definiti e opera all'interno della durata di Sprint.

Un'altra sostanziale differenza è quella delle **cerimonie**. Imprescindibili per la metodologia Scrum, vengono meno in Kanban. Qui si trovano meno imposizioni,

lasciando più libertà ai diretti interessati. Ricorre semplicemente una riunione di allineamento, per fare tutti insieme il punto della situazione e monitorare l'avanzamento del progetto.

Anche il processo di **miglioramento continuo** è differente. Nel caso di Scrum, può essere definito come puntuale in quanto avviene solamente al termine di ogni Sprint in sede di Retrospective e mira a migliorare i processi. Per Kanban invece, è un processo di miglioramento continuo, che avviene attraverso il monitoraggio del flusso di lavoro e l'intervento immediato.

2.6 Framework Hybrid

In riferimento a contesti aziendali il cambiamento da metodi tradizionali a quelli Agile porta con sé delle difficoltà ed è proprio in questa fase che si trovano degli scogli da aggirare per riuscire ad effettuare la transizione al meglio.

Spesso ci si trova in ambienti in cui la cultura aziendale è molto rigida e quindi è possibile che si crei tra i dipendenti una sorta di resistenza al cambiamento, in più, il radicale cambiamento da pianificazione rigida a pianificazione flessibile, crea frequente confusione in quanto può sembrare di non essere stati abbastanza specifici o organizzati in fase preliminare.

Per supportare al meglio il cambiamento è necessario che venga effettuata una approfondita formazione del personale e che l'azienda metta a disposizione strutture e strumenti adeguati.

Trovandosi faccia a faccia con le difficoltà, le aziende spesso apportano delle modifiche alla struttura classica dei framework che vorrebbero utilizzare, dando origine ai cosiddetti framework hybrid.

Il più famoso tra questi è il cosiddetto Scrumban. Come è intuibile dal nome stesso, questo metodo è una sorta di miscela dei due principali framework: Scrum e Kanban. Spesso è utilizzato quando un team di sviluppo software vuole transitare da una gestione rigida ad una un po' più flessibile. Dato che questo metodo, in certi contesti, porti con sé svariati vantaggi, capita che le aziende decidano di adottare direttamente la versione ibrida riuscendo ad estrarre il massimo del valore. [6]

Generalmente si mantiene la cadenza degli sprint per svolgere una determinata quantità di lavoro in un determinato periodo anche se in modo meno rigido essendo questi tempi modificabili a piacimento, e si mantengono i ruoli ben distinti all'interno del team. Dal metodo Kanban invece viene utilizzata la Kanban Board, utile a monitorare e aggiornare il flusso del lavoro in modo continuo, mantenendo comunque un limite al WIP per essere capaci di massimizzare l'efficienza evitando dei controproducenti sovraccarichi.

Ad oggi, quello appena presentato è il maggior framework ibrido presente nelle realtà aziendali.

3 Framework agili applicati a progetti IT di una società di viaggi (Alpitour s.p.a.)

3.1 Alpitour s.p.a.: Introduzione all'azienda

L'azienda, fondata nel 1947, con sede principale a Torino e altre due sedi a Milano e Pesaro, è una delle maggiori realtà nel panorama turistico italiano.

Nel corso del tempo ha affermato la propria presenza grazie ad un'ampia offerta che va dai pacchetti vacanza standard ai viaggi personalizzati e di lusso, fino alle crociere.

Nata operando solo l'offerta di pacchetti viaggio, si è evoluta inglobando altre aziende in modo da internalizzare molti servizi ed essere in grado di ridurre i costi dei propri prodotti, traendone del vantaggio competitivo.

Vanta svariati marchi in diversi settori di mercato tra cui i maggiori sono: la compagnia aerea Neos, il brand specializzato in villaggi turistici Bravo Club e quello dedicato agli alberghi VoiHotels. Il gruppo Alpitour possiede anche marchi di vacanze ricercate e di lusso come Francorosso e Turisanda.

Attualmente l'azienda ha una forte presenza non solo in Italia, ma in tutta la zona del Mar Mediterraneo. Sta comunque crescendo la presenza di destinazioni esotiche.



Figura 3.1: Principali Brand di Alpitour

Le principali aree operative dunque sono:

- **Tour Operating:** principale area dell'azienda, che si occupa di organizzare e vendere pacchetti vacanza;
- **Aviation:** per la gestione di voli e charter;
- **Hotel Management:** utile a fornire sempre strutture adeguate;
- **Incoming:** si occupa di organizzare transfer, escursioni e altri servizi locali per migliorare l'esperienza del cliente durante la sua vacanza;
- **Distribution:** per la distribuzione sia fisica che digitale e raggiungere i clienti;
- **Digital and Innovation:** si occupa di sviluppare le piattaforme digitali e le soluzioni tecnologiche per migliorare l'efficienza aziendale e l'esperienza dei clienti.

Al fine di operare al meglio in tutte le aree elencate, l'azienda dev'essere organizzata internamente in modo ottimale, infatti si distinguono svariate funzioni.

3.1.1 Organizzazione dell'area Information Technology

In questa sezione, sarà analizzata nello specifico la struttura dell'area Information Technology dell'azienda in questione.

Per agevolare nel migliore dei modi l'organizzazione del lavoro e delle varie cerimonie l'area in esame è divisa in Teams, ognuno dei quali opera principalmente in un campo differente. L'azienda opera contemporaneamente su una moltitudine di progetti di piccole, medie o grandi dimensioni. Tendenzialmente i progetti più corposi sono definiti 'cross-team' in quanto richiedono l'impiego di svariate risorse provenienti da diversi teams, al contrario dei progetti modesti che sono svolti in autonomia dalle singole squadre.

Nella analisi che segue, verrà analizzato il flusso di lavoro che i progetti seguono durante il loro svolgimento e l'impatto che le tecniche Agili hanno all'interno dell'area precedentemente mostrata.

Ad operare secondo tecniche Agile, attualmente sono 10 team suddivisi nelle rispettive aree come segue:

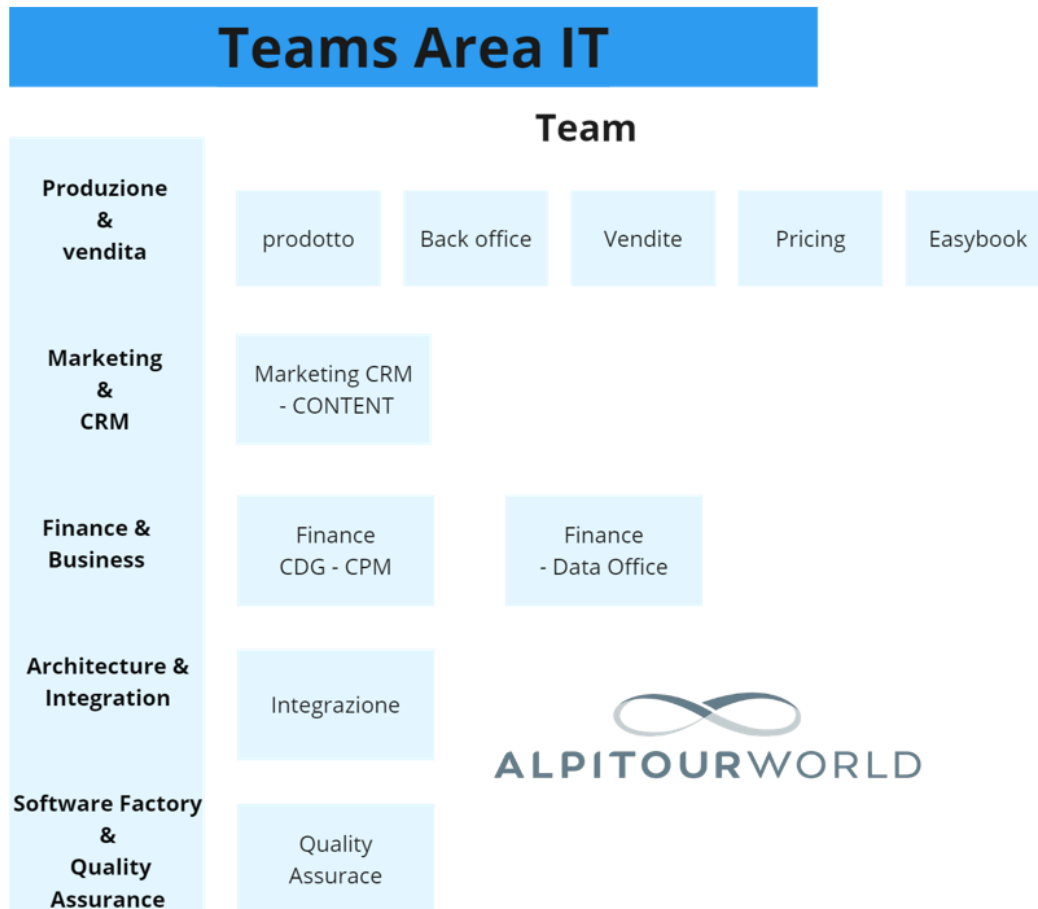


Figura 3.2: Organizzazione teams Area IT Alpitour

3.2 Presenza e impatto dell'Agile da un punto di vista interno

Per comprendere al meglio le dinamiche dei metodi Agile è stata effettuata un'analisi dell'area IT relativa all'azienda in questione, ponendo delle domande ai dipendenti. Questo permette di avere un quadro generale di quanto il reparto sia confortevole con l'uso della metodologia, di quali siano gli aspetti positivi e quali i negativi.

Un ulteriore aspetto da considerare nella lettura dell'analisi è che l'azienda ha introdotto l'Agile da poco tempo, poco più di un paio di anni, perciò le risposte raccolte saranno da interpretare come una sorta di 'primo impatto' e non come un 'impatto definitivo'.

3.2.1 Raccolta dei dati

Per raccogliere informazioni utili a sviluppare un'analisi riguardante l'impatto dell'introduzione di tecniche Agili in un'azienda di tour operating, è stato redatto un questionario.

Questo ha portato grandi vantaggi e ha facilitato l'analisi in corso, poiché tramite esso è stato possibile raccogliere dati quantitativi (di tutti i dipendenti) e qualitativi, per avere una visione completa della situazione. Si è deciso di optare per la compilazione anonima, in modo da non falsare la veridicità delle risposte e far sì che tutti rispondessero in modo onesto. Come piattaforma è stata scelta 'Google Forms' utilizzata sia per la stesura, che per la vera e propria raccolta delle risposte, le quali sono state pensate 'chiuso' in modo da facilitare la successiva analisi dei dati tramite l'utilizzo di appositi software. Il questionario risulta suddiviso in 5 distinte sezioni.

La prima è dedicata alla descrizione dell'ambito del questionario, agli scopi della raccolta dati e al ringraziamento dei partecipanti.

Le domande cominciano nella sezione successiva, la quale essendo composta da domande anagrafiche a risposta chiusa è fondamentale per avere un quadro generale dei rispondenti.

Segue un modulo dedicato alla comprensione della metodologia stessa, dove si prova ad capire quanto i dipendenti conoscano l'argomento trattato, se lo utilizzino nel lavoro quotidiano e, in caso di risposta affermativa, quanto si sentano padroni delle tecniche impiegate.

La parte numero quattro mira a comprendere l'impatto positivo dell'implementazione di Agile in azienda e quanto quest'ultima abbia contribuito in termini di formazione e fornitura degli strumenti adatti.

L'ultima sezione, invece, è volta ad identificare con quale intensità si siano verificate le principali difficoltà di organizzazione o implementazione.

Negli ultimi due moduli, per capire effettivamente l'intensità delle risposte degli interessati si è optato per domande con scala Likert, ovvero un metodo comunemente utilizzato nelle survey per misurare opinioni e percezioni con risposte numeriche che vanno da 1 a 5. Generalmente all'1 corrisponde essere 'completamente in disaccordo' e al numero 5 corrisponde 'totalmente d'accordo'. E' stato deciso di utilizzare un intervallo di quest'ampiezza e granularità in modo da non ottenere una risposta binaria, ma una risposta più veritiera facendo esprimere gli intervistati secondo diverse sfumature. Inoltre è anche un metodo molto facile da comprendere, e rende il questionario abbastanza scorrevole dal punto di vista di chi lo sta compilando, e permette all'analista di trasformare i dati qualitativi (come opinioni e pareri) in dati quantitativi, rendendo anche possibile l'elaborazione statistica.

Nel caso di questa analisi, lo scopo è quello di mettere su carta l'impatto che l'implementazione di Agile ha da un punto di vista interno, e proprio tramite l'utilizzo della scala Likert è possibile rilevare variazioni tra i diversi gruppi che formano l'Area IT.

Nel corso dell'ultima settimana di luglio, il questionario è stato somministrato in due diverse ondate, in via telematica, a tutti i dipendenti dell'area Information Technology dell'azienda, che formano un campione di circa quaranta individui.

Le risposte sono state raccolte per la maggior parte il giorno stesso alla prima diffusione e sono risultate essere 26, fornendo un tasso di partecipazione di circa il 70%.

Considerando il periodo estivo, caratterizzato da un'alta concentrazione di ferie dei dipendenti, il tasso ottenuto è considerato soddisfacente e abbastanza esteso da essere rappresentativo dell'intera popolazione.

3.2.2 Descrizione del campione

Come detto precedentemente, il primo modulo del questionario è utile a identificare i partecipanti tramite domande di carattere anagrafico.

In media, le persone intervistate hanno tra i 30 e i 40 anni, sono nel 65% dei casi maschi e possiedono nell'81% una laurea, che sia di primo o di secondo livello.

E' importante, per interpretare i dati nel modo corretto, scoprire il ruolo che ogni persona possiede all'interno dell'azienda e da quanto tempo essa sia stata assunta.

Logicamente, le impressioni ricevute da una persona che lavora in azienda da meno di un anno hanno un'importanza limitata, perché può darsi che essa non si sia ancora totalmente ambientata e non si senta ancora a proprio agio con i metodi utilizzati, falsando quindi la veridicità delle sue risposte.

D'altro canto però, sebbene ristrette per numero (solamente il 4%), queste opinioni possono far riflettere su come l'azienda effettui la formazione in materia e come si ponga nei confronti dei nuovi arrivati.

Secondo un ragionamento analogo, le persone che lavorano in azienda da tanti anni (un cospicuo 27%), possono essersi abituate ai metodi tradizionali nel corso del tempo e quindi potrebbero esprimersi nei confronti delle tecniche innovative in modo negativo, definendole come confusionarie o poco utili solo perché differenti da quelle applicate per moltissimo tempo, senza neppure sforzarsi di comprenderle appieno.

Anche il ruolo ricoperto è determinante alla percezione del metodo, in quanto ogni individuo avrà anche dei compiti differenti all'interno delle varie cerimonie.

E' molto difficile che uno Scrum Master percepisca negativamente l'utilità del metodo, così come uno sviluppatore, che deve svolgere task precisi, potrebbe trovare monotone e ridondanti le riunioni settimanali, sentendosi in grado di organizzare il proprio lavoro da sé.

Dal punto di vista di un Solution Leader invece, essendo il responsabile dell'attività svolta, queste riunioni frequenti possono risultare molto utili, un'occasione per far frequentemente mente locale e non dimenticare le attività da completare.

Di seguito sono presenti i grafici a torta e a barre che racchiudono la totalità dei dati raccolti nella prima sezione:

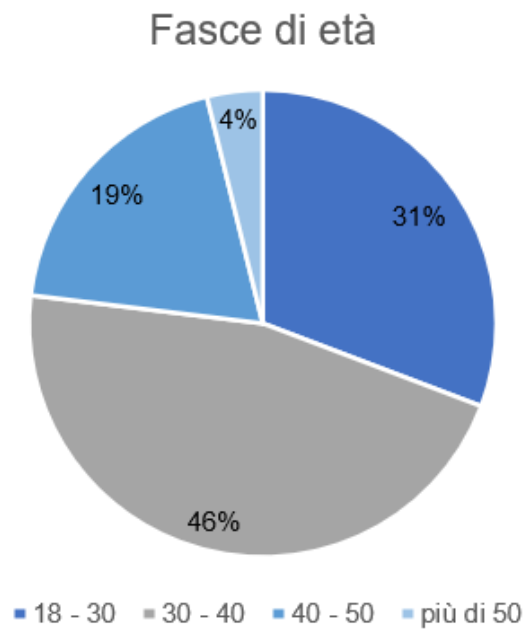


Figura 3.3: Grafico a torta dell'età dei rispondenti

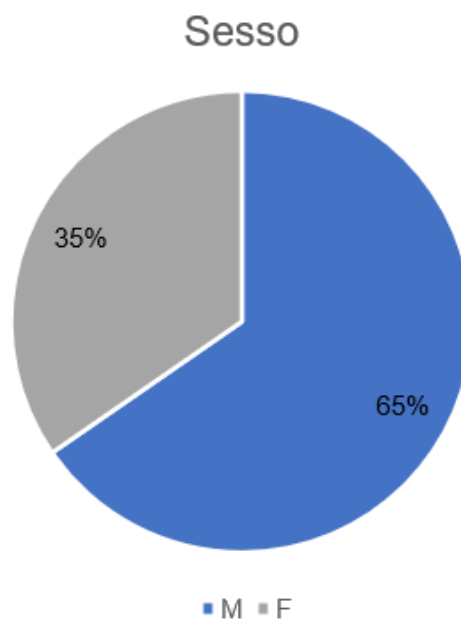


Figura 3.4: Grafico a torta del sesso dei rispondenti



Figura 3.5: Grafico a torta del titolo di studio dei rispondenti

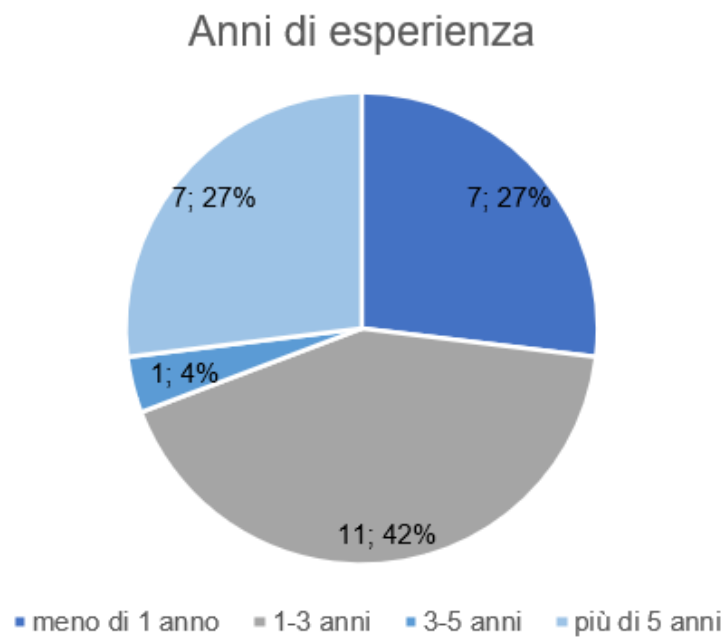


Figura 3.6: Grafico a torta degli anni di esperienza dei rispondenti

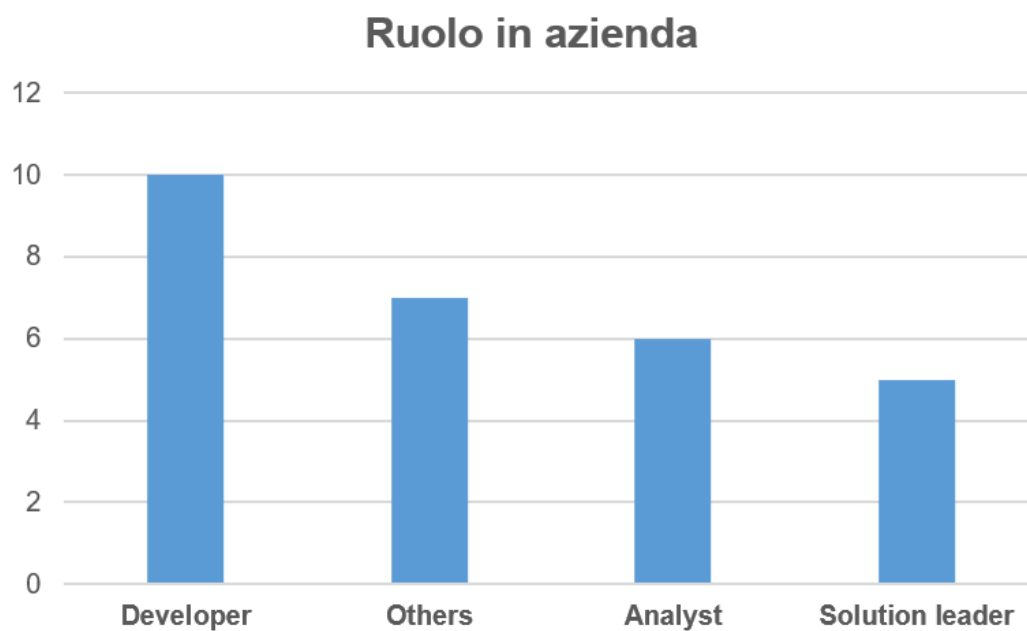


Figura 3.7: Grafico a barre dei ruoli in azienda dei rispondenti

3.2.3 Test non parametrico di Kruskal-Wallis

Quello di Kruskal-Wallis è un test statistico non parametrico, in quanto non si avvale di alcun tipo di ipotesi sulla distribuzione dei dati presi in analisi. E' utilizzato per verificare se i gruppi differiscano significativamente tra loro in termini di mediana, basandosi sui ranghi delle osservazioni.

La procedura di svolgimento è molto semplice ed intuitiva. Bisogna verificare la veridicità dell'ipotesi nulla, la quale afferma che non ci sono differenze significative tra i vari gruppi, al contrario dell'ipotesi alternativa.

Dopo aver individuato le variabili categoriche e i diversi gruppi, è necessario ordinare tutte le risposte di una specifica categoria e assegnare dei ranghi, le quali medie poi verranno confrontate tra i diversi gruppi.

E' stato scelto questo test perché molto vantaggioso. Infatti, come detto precedentemente, non richiede che i dati seguano alcun tipo di distribuzione, e possono essere utilizzati dati ordinali (proprio come le risposte in scala Likert), rendendolo un test perfetto per l'analisi dei dati ricavati da questionari. Non richiede omogeneità delle varianze, il che lo rende ancor più flessibile ed applicabile a svariati data-set. Inoltre, un'altro grande vantaggio è costituito dalla semplicità, sia di esecuzione del test stesso, che dell'interpretazione delle conclusioni le quali sono molto chiare e dirette.

Nel caso di studio in questione, il test di Kruskal-Wallis è stato eseguito avvalendosi di 'Minitab' un tool digitale statistico utile all'analisi dei dati e allo svolgimento di test.

Dopo aver organizzato le risposte del questionario ed eliminato quelle non pertinenti, è stato sufficiente importare i dati all'interno del software e procedere con il test andando ad incrociare ogni variabile categorica con tutte le domande a risposta numerica per ottenere un output e permettere di fare delle valutazioni.

I gruppi sono identificati dalle domande di tipo anagrafico, e risultano essere principalmente quattro: Età dei dipendenti, Titolo di studio, Anni di esperienza in azienda e Ruolo in azienda. Si è pensato che se dovesse esistere una differenza significativa nella percezione del metodo Agile all'interno dell'azienda, questa con ogni probabilità sarebbe ricaduta su una di queste variabili. Sono state scelte poi le domande del questionario da analizzare e sulle quali è opportuno focalizzare l'attenzione.

Nelle pagine seguenti è riportata una tabella creata estrapolando i dati rilevanti dall'output ottenuto tramite il test su Minitab.

Creare una tabella di questo tipo è sicuramente molto vantaggioso, infatti permette di effettuare le valutazioni dell'analisi avendo sempre sott'occhio i principali dati, ovvero le mediane e il 'p-value'. Quest'ultimo valore è il primo campanello di allarme per la valutazione, infatti se quest'ultimo è inferiore al 5% significa che sicuramente ci sono delle differenze significative tra i gruppi in analisi e le risposte date. Anche le mediane sono un importante aspetto da valutare, poiché se tra i risultati si trovano dei valori di mediana molto distanti tra loro, può significare che il valore della risposta non rappresenti fedelmente la realtà.

CAPITOLO 3. FRAMEWORK AGILI APPLICATI A PROGETTI IT DI UNA SOCIETÀ DI VIAGGI (ALPITOUR S.P.A.)

	Livello familiarità	p-value	Efficacia dell' implementazione	p-value	Supporto aziendale (strumenti)	p-value	Utilità delle cerimonie	p-value	Difficoltà aggiornamento backlog di prodotto	p-value
Età dei dipendenti										
18-30	3		3		4		3		3	
30-40	3,5	0,160	3	0,687	3	0,575	3,5	0,275	3	0,539
40-50	4		3		3		4		3	
Più di 50	5		4		4		5		4	
Titolo di studio										
Diploma	3		3		3		4		3	
Laurea di I livello	3	0,168	3	0,510	4	0,581	3	0,385	3	0,858
Laurea Magistrale o master	4		3		3		3		3	
Anni di esperienza in azienda										
Meno di 1 anno	3		3		4		3		3	
1-3 anni	3	0,696	3	0,281	3	0,413	3	0,121	3	0,310
3-5 anni	4		1		2		1		3	
Più di 5 anni	3		3		4		4		4	
Ruolo in azienda										
Developer	3		3		3		3		4	
Others	4	0,907	3	0,288	4	0,644	4	0,231	4	0,255
Analyst	4		2,5		3,5		3		3	
Solution leader	3		4		4		4		3	

Figura 3.8: Risultati del Test di Kruskal-Wallis

CAPITOLO 3. FRAMEWORK AGILI APPLICATI A PROGETTI IT DI UNA
SOCIETÀ DI VIAGGI (ALPITOUR S.P.A.)

	Miglioramento produttività personale	p- value	Resistenza al cambiamento	p- value	Mancanza di formazione	p- value	Strutture inefficaci	p- value	Scarsa gestione del tempo	p- value
Età dei dipendenti										
18-30	3		3		1		2,5		3,5	
30-40	3	0,394	4	0,161	4	0,308	2	0,233	3	0,916
40-50	4		4		4		1		4	
Più di 50	4		5		1		2		3	
Titolo di studio										
Diploma	3		4		4		2		3	
Laurea di I livello	3	0,986	4	0,975	4	0,702	2	0,959	3	0,098
Laurea Magistrale o master	3,5		4		2		2		5	
Anni di esperienza in azienda										
Meno di 1 anno	3		3		1		2		3	
1-3 anni	3	0,455	4	0,043	4	0,589	2	0,224	4	0,642
3-5 anni	2		2		1		4			
Più di 5 anni	3		3		4		2		3	
Ruolo in azienda										
Developer	3		3		2		3,5		4	
Others	4	0,252	4	0,094	2	0,357	2	0,011	3	0,939
Analyst	3		4		4		1		3	
Solution leader	4		4		4		1,5		3,5	

Figura 3.9: Risultati del Test di Kruskal-Wallis

Si analizzano ora le tabelle riportate, in riferimento all'area IT dell'azienda Alpitour S.P.A.

Osservando la prima riga delle tabelle 3.8 e 3.9, si nota come l'età dei dipendenti sembra non evidenziare alcuna tendenza particolare riportando p-value elevati e mediane molto simili tra loro.

L'unico caso che può far riflettere è l'incrocio con la mancanza di formazione, in quanto i giovani tra i 18 e i 30 anni di età sembra che considerino la formazione ricevuta adeguata e sufficiente per poter operare secondo certi standard. Le persone più anziane invece sembrerebbero percepire in modo più pesante questa difficoltà, ritenendo la formazione ottenuta insufficiente. Questa distanza molto marcata tra le mediane fa sì che per la nostra analisi questo dato diventi inaffidabile, in quanto potrebbe non rispecchiare l'effettiva situazione aziendale mostrando una differenza tra i gruppi che potrebbe non esistere.

Relativamente al titolo di studio, tramite il test siamo in grado di affermare che non esistono differenze significative tra chi ha il diploma, chi ha una laurea di primo livello e chi una di secondo livello e che questo dato sia certamente attendibile per via delle mediane molto simili e vicine tra loro.

Nella riga degli anni di esperienza in azienda, si trova il primo p-value inferiore al 5% , precisamente 4,3% , a comunicare che sicuramente c'è una differenza significativa tra i gruppi con diversa esperienza. Con più precisione è possibile affermare che chi lavora da meno tempo all'interno dell'azienda trovi nella resistenza al cambiamento un grande scoglio da superare per il miglioramento al contrario di chi invece in azienda lavora da tre o più anni. Ciò è perfettamente logico, poiché generalmente le persone che lavorano in uno stesso posto da tanto tempo hanno sviluppato una routine lavorativa e possono essere ostili nei confronti del cambiamento andando anche a danneggiare l'operato dei più inesperti.

Anche in riferimento al ruolo in azienda, troviamo un p-value interessante dell'1,1%. Il dato ci fornisce l'informazione che tra le varie categorie di ruoli aziendali sicuramente ci sono delle differenze significative in merito alle strutture di comunicazione. Analizzando le mediane è possibile stabilire che per gli Analyst ci sono molte più difficoltà di comunicazione rispetto a chi ricopre altri ruoli.

Ciò però è un dato importante, poiché essendo gli analisti le persone che più hanno bisogno di comunicazione nel corso del loro lavoro ed essendo loro a trovare delle grandi difficoltà, significa che sicuramente queste strutture dovrebbero sicuramente essere migliorate. Gli sviluppatori non hanno evidenziato difficoltà probabilmente perché essi comunicano poco proprio per la natura del ruolo stesso.

3.3 Ciclo-vita dei progetti

Come visto precedentemente, l'azienda possiede un'area IT interna, nata in seguito allo sviluppo tecnologico affrontato nel corso degli anni. E' quindi molto importante che essa sia in grado di gestire i progetti relativi a quest'area dalla nascita fino al completamento.

La gestione di un progetto IT segue 5 fasi sequenziali imprescindibili:

1. **Richiesta di un progetto:** Esso nasce quando ne viene effettuata la richiesta; Essa può avvenire per un progetto nuovo o per apportare modifiche a quelli già esistenti. In entrambi i casi le business unit interne preparano una proposta progettuale tramite un tool di supporto (Wrike), all'interno del quale il Product Owner deve compilare un apposito form chiamato 'Proposta Nuovo Progetto' indicando un responsabile di progetto, i benefici attesi e le priorità da seguire.
2. **Definizione dei requisiti:** Gli analisti incaricati e il Responsabile di progetto stilano una lista di requisiti, ovvero delle caratteristiche da sviluppare per poter considerare il progetto un successo e coinvolgono le aree business necessarie.
3. **Analisi preliminare:** Segue la fase di analisi in cui, consultando nuovamente Wrike, il Business Analyst IT cerca di generare una previsione in termini di tempi e costi per rendersi conto della mole di lavoro e verificare la fattibilità. Infatti già in questo momento, l'analista cerca di capire anche quante risorse interne saranno utilizzate e in che percentuale in modo da non rovinare il lavoro di altre aree (o altri progetti) che potrebbero essere danneggiate dall'impiego di risorse già adibite a svolgere altri compiti.
4. **Approvazione:** In questa fase il progetto viene approvato (o respinto) e si predispongono l'organizzazione delle risorse per l'avvio dello stesso. E' proprio l'IT PMO che si occupa di inviare una email agli stakeholder contenente: Report, costi/benefici, Requisiti, Pianificazione. Se il progetto è respinto, si possono ripercorrere le fasi precedenti per apportare modifiche e migliorie o semplicemente annullarlo.
5. **Esecuzione:** Essa è suddivisa in pianificazione di alto livello (tramite Wrike), pianificazione di dettaglio (tramite DevOps) e sviluppo (tramite DevOps). In generale si può dire che Wrike sia utilizzato per la pianificazione e l'avanzamento delle macro attività, mentre DevOps è usato per le micro-attività IT.

3.4 Valutazione dell'andamento di progetti agili tramite KPI

3.4.1 Key Performance Indicator

I KPI (Key Performance Indicator) sono, come dice il termine stesso, degli indicatori chiave di prestazione e sono fondamentali per monitorare l'impatto che una qualsiasi attività ha durante il suo svolgimento. Non si limitano a questo, infatti guidano anche le scelte da fare nell'imminente futuro e fungono da strumento di comunicazione tra i team e gli stakeholders, i quali tramite la loro consultazione possono avere un'idea dell'andamento generale del progetto.

E' importante, per prendere decisioni che venga stabilito un range all'interno del quale debbano stare gli indicatori e dei valori target, da provare a raggiungere per un ottimale svolgimento.

Per far sì che siano significativi è necessario che gli obiettivi da raggiungere siano chiari e precisi e che gli indicatori siano misurabili facilmente senza perdere di vista il risultato da ottenere. E' inoltre molto importante che le metriche in questione siano frequentemente aggiornate e che siano reperibili facilmente e velocemente cosicché possano essere disponibili nel momento del bisogno. Un indicatore fondamentale può risultare inutile se non è disponibile quando serve.

Si utilizzano in vari settori e campi di applicazione ma nel testo si fa riferimento a KPI relativi a progetti IT dei quali poi sono riportati anche esempi pratici. L'attenzione è rivolta solamente agli indicatori utili per l'analisi di progetti svolti nell'ambito delle tecniche Agile.

3.4.2 Accuratezza dell'effort previsto

Relativamente all'area IT in questione, quando si pianificano le attività inerenti ad un progetto, sia che si tratti di PBI, sia che si tratti di Task, si utilizza il tool di supporto Azure DevOps.

Qui è necessario durante la pianificazione, compilare un campo relativo all'effort necessario per lo svolgimento dell'attività. A priori però, non è sempre facile prevedere la quantità di effort corretta per via di possibili complicanze o ritardi in fase di svolgimento.

E' il Team Leader ad avere l'ultima parola in termini di previsione di effort, ma solamente previa consultazione con l'assegnatario e il Business Analyst. Si può dunque dire che sia il Team che il Leader debbano avere una buona conoscenza delle attività e delle generali problematiche per far in modo da prevedere un effort che sia poi rispettato facendo sì che non ci sia accavallamento di task tra le varie settimane, affidandosi oltre che alle tecniche di stima anche alla loro esperienza personale.

Parlando di effort, si fa riferimento principalmente allo scostamento della previsione dall'impegno effettivo necessario calcolato a posteriori. In questo ambito sono interessanti due principali KPI: lo scostamento dell'effort e il suo indice di accuratezza.

$$\text{ScostamentoEffort} = \text{EffortPianificato} - \text{EffortReale}$$

$$\text{IndiceAccuratezzaEffort} = \frac{\text{EffortReale}}{\text{EffortPianificato}}$$

Nel caso dell'azienda in analisi, l'unità di effort alla quale si fa riferimento è l'ora lavorativa. Possono anche essere utilizzate come unità i giorni o addirittura le settimane.

Sebbene le ore lavorative disponibili durante una giornata siano otto, per motivi di precauzione si è cominciato a calcolare le proprie capacità come se invece fossero al massimo sette, in modo da poter permettere le pause obbligatorie senza che ci siano delle incongruenze con la pianificazione.

Il primo KPI presentato è di molto facile comprensione:

- **Scostamento > 0:** significa che l'Effort relativo all'attività in questione è stato sovrastimato, ed è stato necessario meno sforzo per raggiungere il completamento.
- **Scostamento = 0:** allora la pianificazione è stata perfetta e perfettamente congrua con l'impegno reale.
- **Scostamento < 0:** L'attività in questione è stata sottostimata e nella realtà dei fatti ci è voluto più tempo di quanto preventivato.

l'indice di accuratezza dell'effort è come logica molto simile al precedente anche se di meno diretta visualizzazione. Infatti se:

- **Indice > 1:** L'effort previsto è stato sovrastimato.
- **Indice = 1:** Perfetta stima dell'effort.
- **Indice < 1:** L'effort è stato sottostimato.

Come detto in precedenza è necessario avere un target entro il quale bisogna cercare di trovarsi per poter procedere in maniera più efficiente possibile. Nel caso dello scostamento ovviamente il valore target è 0, mentre nell'altro caso 1.

Nella situazione in cui l'effort non è stimato in modo corretto si incorre in dei problemi: Se la stima è superiore al tempo necessario, si crea il problema della mancata allocazione di capacità e ci si rende conto che si sarebbe potuto fare più di quanto fatto, velocizzando il percorso verso lo Sprint goal. D'altro canto, se la stima fatta è troppo bassa, si genera il problema opposto, in cui non si riescono a completare tutte le attività schedulare, creando delle sovrapposizioni e ritardando il raggiungimento dell'obiettivo.

3.4.3 Lead Time

Il Lead Time è un'altro indicatore di fondamentale importanza. Viene definito come il lasso temporale compreso tra l'inserimento di una nuova attività nel sistema e il momento in cui essa può essere considerata completata.

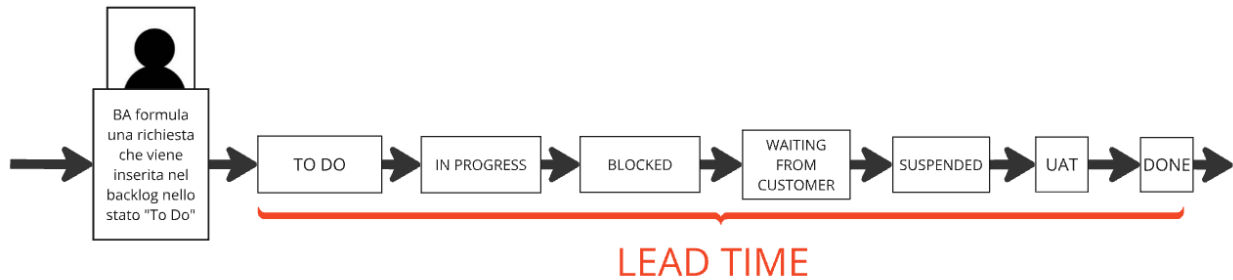


Figura 3.10: Grafico che rappresenta il lead time

Facendo riferimento alla figura 3.10, che riporta gli stati utilizzati nella gestione dei task di Alpitour, si può definire Lead Time come il tempo che un task impiega dallo stato "TO DO" a raggiungere lo stato "DONE". Un'accortezza da avere nello specifico caso dell'azienda è quella di cominciare a calcolare questo tempo dal momento in cui il task viene assegnato ad una persona, perchè capita che alcune attività vengano inserite in DevOps e rimangano in attesa di un assegnatario per settimane o addirittura mesi.

Gli stati obbligatori sono "To Do", "In Progress" e "Done", gli altri sono solo eventuali e generalmente quando si passa per questi ultimi si genera un ritardo ottenendo un Lead Time maggiore del dovuto. Questo è anche influenzato dalla priorità dei Task, può capitare che uno in priorità bassa venga assegnato e poi lasciato in attesa per via dell'aggiunta in corso d'opera di un nuovo compito con priorità più alta.

3.4.4 Cycle Time

Il Cycle Time è il tempo effettivo impiegato da un dipendente, nel caso dell'area IT in questione quasi sempre uno sviluppatore, a completare un'attività dal momento in cui inizia a lavorarci.

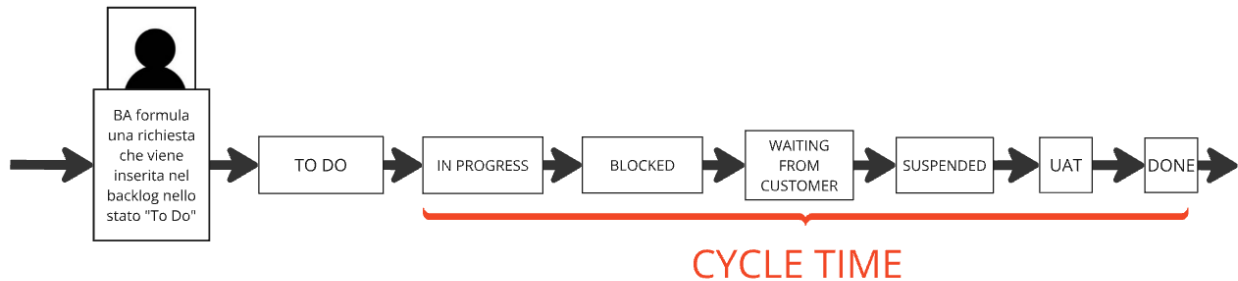


Figura 3.11: Grafico che rappresenta il Cycle Time

Facendo riferimento alla figura 3.11 che riporta gli stati utilizzati su DevOps in azienda si può affermare che il Cycle Time corrisponda al periodo di tempo che trascorre dal momento in cui ad un task viene assegnato lo stato "In Progress" fino al suo completamento, cioè allo stato "Done".

Un'osservazione in merito a questo KPI è che sarà sempre inferiore al Lead Time in quanto compreso in esso. Questa differenza si assottiglia moltissimo per i task categorizzati come ad alta priorità, poiché questi ultimi vengono presi in carico (e quindi messi "In Progress") immediatamente dopo la loro creazione, riducendo quasi a zero il tempo che intercorre tra lo stato "To Do" e lo stato "In Progress". Anche in questo caso eventuali sospensioni e tempi di attesa causano un aumento del indicatore osservato.

In questo caso, così come nel precedente, non è presente un vero e proprio target se non la previsione effettuata in fase di pianificazione. In linea generale si può dire che si ha come obiettivo la minimizzazione di questi tempi, senza tralasciare le qualità del prodotto.

3.4.5 Numero di PBI svolti nella durata di uno Sprint

Il numero di Product Backlog Items svolti nella durata di uno Sprint è sicuramente un indicatore che è opportuno monitorare nello svolgimento di un progetto, per poter misurare l'avanzamento di quest'ultimo.

Per i progetti di dimensioni rilevanti, che durano diversi mesi e quindi molteplici Sprint si tiene sotto controllo il numero medio, così da poter rilevare eventuali rallentamenti e poter agire sulle cause.

Questo numero medio è dato da:

$$NumMedioPBI = \frac{\sum NumPBISingoloSprint}{NTotSprintSvolti}$$

L'indicatore è anche utile in fase di pianificazione poiché fornisce un'idea del numero totale di PBI che è opportuno schedulare nel corso delle sequenziali iterazioni anche in base ai dati passati.

In merito a quest'ultimo indicatore si deve tendere a massimizzare questo numero, senza però sovrallocare i dipendenti.

Segue una valutazione di alcuni progetti aziendali conclusi, tramite l'utilizzo dei citati KPI.

3.5 Valutazione di progetti aziendali, mediante l'utilizzo di KPI

In questa sezione dell'elaborato, è presentata un'analisi dettagliata dei KPI citati precedentemente in relazione a progetti IT gestiti secondo tecniche agili conclusi nel corso degli ultimi anni.

I progetti presi in esame sono quattro e sono i seguenti:

- **WebCola;**
- **Turisanda;**
- **Nuovo Flusso;**
- **Via Milano Parking.**

Di questi progetti viene fatta una prima una presentazione relativa all'ambito e allo scopo e poi un'analisi tramite indicatori.

3.5.1 WebCola



Figura 3.12: Logo WebCola

Come già detto l'azienda si occupa di gestione e offerta di pacchetti viaggio, nei quali oltre all'alloggio sono forniti anche i biglietti aerei di andata e ritorno.

L'acquisto da parte dell'azienda non avviene attraverso le stesse piattaforme che utilizzano gli utenti privati, bensì per avere accesso alle tariffe più vantaggiose Alpitour si trova costretta a passare attraverso dei canali tecnici.

L'accesso a questi canali specifici comporta il pagamento di una tassa importante e dato il fatto che ogni compagnia aerea possiede un diverso canale, è impossibile per l'azienda avere libera scelta per trovare sempre il prezzo più vantaggioso.

WebCola è appunto una piattaforma che permette l'accesso a tariffe di biglietti aerei tramite NDC, e lo scopo del progetto è proprio quello di integrarla nelle dinamiche aziendali.

NDC corrisponde a New Distribution Capability, ovvero uno standard sviluppato dall'IATA (International Air Transport Association) utile a migliorare la distribuzione delle tariffe e servizi delle compagnie aeree senza dover passare per il canale tradizionale GDS (Global Distribution System).

L'integrazione della piattaforma citata, sebbene attualmente fornisca un servizio relativo solamente a Emirates NDC, sarebbe un importante passo per ottenere un vantaggio competitivo nel futuro, in quanto WebCola ha un margine di miglioramento grandissimo e si pensa che con l'avanzare del tempo sarà possibile ottenere tramite questo canale le tariffe di molteplici compagnie aeree.

In questo modo l'onere sostenuto per l'integrazione sarebbe sicuramente inferiore ai guadagni che si potrebbero ottenere, oltre che in termini di denaro, anche in termini di riduzione del time to market e della semplificazione della gestione interna di questo tipo di flussi.

Il progetto si è sviluppato nell'arco di 42 Sprint, per un totale di tre mila ore, nelle quali sono stati eseguiti i seguenti Work Items:

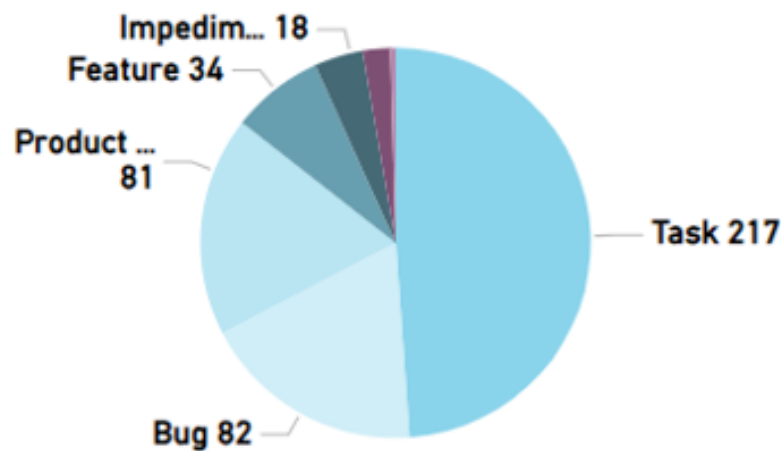


Figura 3.13: Grafico a torta dei Work item per tipo relativi a WebCola

Numero di PBI svolti nella durata di uno Sprint

Come evidenziato dal grafico a torta precedente i tipi di work item sono molteplici. Per l'esame di questo KPI però si fa riferimento ai soli PBI, spiegati nel Capitolo 2, che risultano essere ottantuno. Conoscendo il numero finale degli Sprint che è quarantadue, possiamo calcolare l'indicatore:

$$NPBIMedio = \frac{NTotPBI}{NTotSprint} = \frac{81}{42} = 1,93$$

In media, nella durata del progetto, sono stati svolti poco meno di due PBI per ogni Sprint della durata di due settimane (con l'eccezione del mese di Agosto dove è stato svolto un unico Sprint della durata di trenta giorni).

Scostamento dell'effort

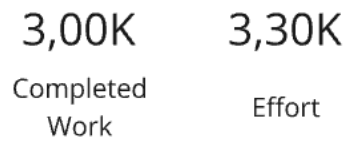


Figura 3.14: Effort e Completed work

Nell'immagine è presentato il conteggio delle ore totali dedicate al progetto a confronto con le ore che sono state preventivate a priori. Si nota una differenza tra i due valori, più precisamente si può affermare che le attività svolte sono state in media sovrastimate e ci è voluto meno tempo del previsto a completarle in quanto lo scostamento è appunto > 0 (300h). Infatti sono state preventivate 3300 ore ma ne sono bastate solo 3000 per giungere alla conclusione del progetto.

L'indice di accuratezza dell'effort comunque non è da considerarsi negativo poiché la previsione è stata abbastanza precisa:

$$IndiceAccuratezzaEffort = \frac{EffortReale}{EffortPianificato} = \frac{3000h}{3300h} = 0,9091 = 90,91\%$$

Lo stesso ragionamento si può applicare alle singole attività effettuando un'analisi più accurata. Nel caso in esame si distinguono diverse tendenze in relazione al tipo di work item.

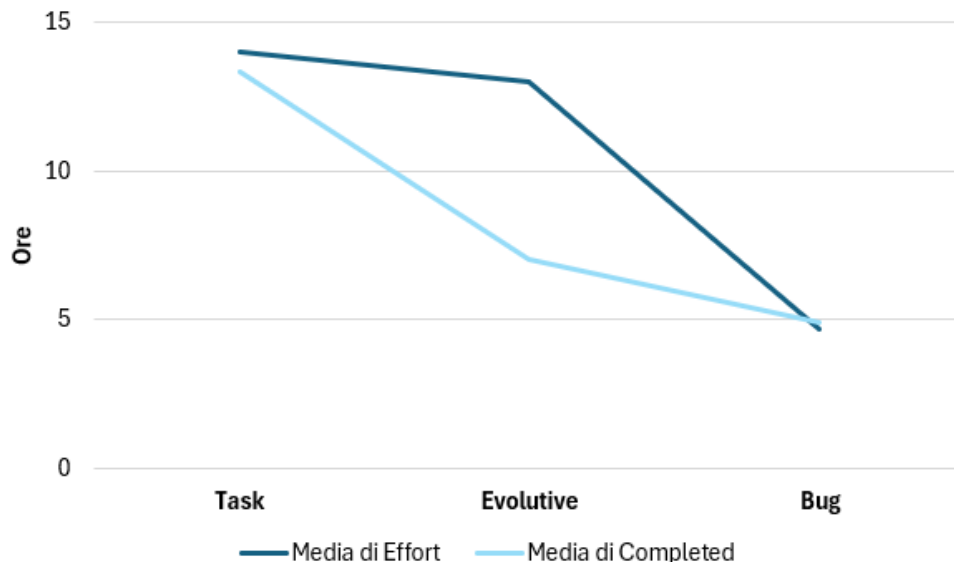


Figura 3.15: Medie di Effort e Completed work per Work item type

Come si può denotare dal grafico nel quale sono riportati sull'asse delle ascisse i principali work item e sulle ordinate le ore, l'effort previsto relativo ai Task è lievemente superiore al lavoro effettivo svolto, mentre quello relativo ai bug è di

poco inferiore. Questi non sono dati preoccupanti poiché i due valori sono molto vicini tra loro per entrambe le categorie di work item (in media meno di un ora di differenza).

Il dato relativo alle Richieste di Evolutiva invece è più rilevante in quanto lo scostamento tra i due valori in esame è molto maggiore. Infatti la media di effort stimato per ogni Evolutiva è di 13 ore, mentre il lavoro svolto per il completamento è in media di 7 ore:

$$\text{IndiceAccuratezzaEffort} = \frac{\text{EffortReale}}{\text{EffortPianificato}} = \frac{7h}{13h} = 0,5385 = 53,85\%$$

Questa percentuale non è soddisfacente poiché è molto bassa. E' stata allocata molta capacità lavorativa dei dipendenti per svolgere dei compiti che ne hanno richiesta molta meno. Ciò comporta un basso sfruttamento della risorsa, e comunica che il progetto possa essere svolto più velocemente di quanto fatto.

Lead Time

La figura sottostante riporta un grafico a barre raffigurante il lead time dei principali work items del progetto:

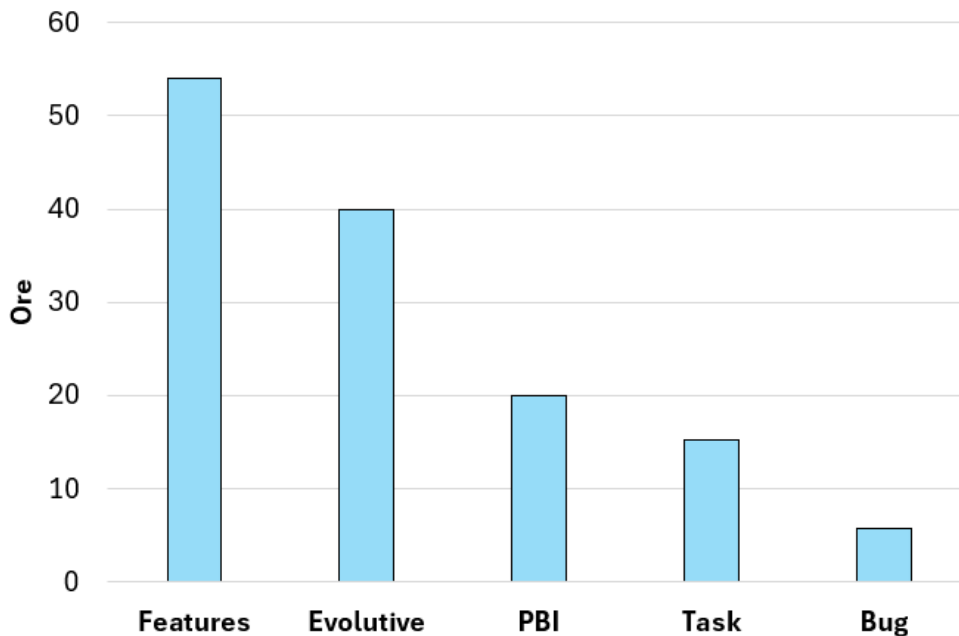


Figura 3.16: Lead Time per Work Item Type

Si nota dalla Figura 3.16 che le features progettuali hanno riportato un lead time molto maggiore rispetto agli altri items, superando le 50 ore. Questo è un dato di tendenza, poiché le Features essendo work items di alto livello, dipendono dallo svolgimento di attività minori come appunto Task, PBI Evolutive e Bug, ed è normale che il tempo in questione sia elevato. Il dato relativo alle richieste di evolutiva attira l'attenzione essendo molto elevato rispetto all'entità dell'item.

Ciò fa capire che effettuare migliorie o altri cambiamenti dovuti a necessità, sono stati determinanti sulla durata dell'intero progetto.

Considerando i seguenti lead time:

- Features: 54h
- Evolutive: 40h
- PBI: 20h
- Task: 15,3h
- Bug: 6h

Il lead time medio indipendentemente dal tipo di oggetto di lavoro è di 27,06 ore lavorative che, considerando una capacità giornaliera di 7 ore, corrispondono a 3,87 giorni lavorativi.

Cycle Time

Il grafico a barre sottostante riporta tutti i valori medi del cycle time divisi per work item:

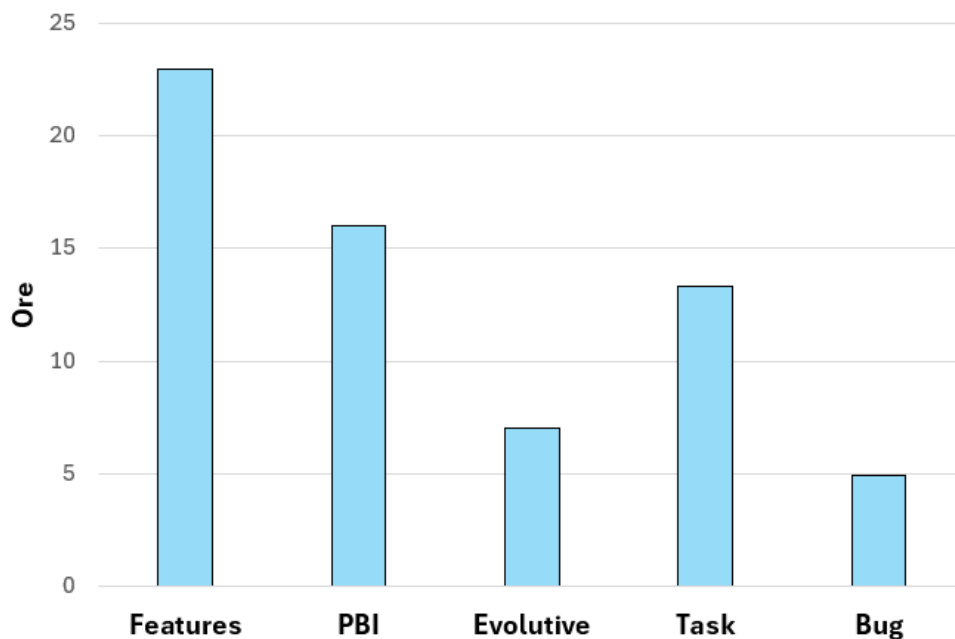


Figura 3.17: Cycle Time per Work Item Type

Chiaramente, come ci si poteva aspettare, i valori di questo indice sono inferiori rispetto ai precedenti, dato che lo stato 'In Progress' (stato dal quale si inizia a calcolare il tempo ciclo) è successivo allo stato 'To Do'. Lo scarto tra i due stati citati rappresenta il tempo medio di attesa di un work item, a partire dal momento in cui viene assegnato fino a che non comincia il suo sviluppo.

Considerando i seguenti valori di tempo ciclo:

- Features: 22,9h
- Evolutive: 7h
- PBI: 16h
- Task: 13,3h
- Bug: 4,9h

Si è ottenuto un tempo ciclo medio di 12,82h (minore delle 27,06 ore del lead time), che considerando la capacità lavorativa giornaliera di 7h corrispondono a 1,83 giorni.

3.5.2 Turisanda (Eventi)



Figura 3.18: Logo Turisanda

Turisanda è un tour operator italiano che vanta ormai 100 anni di storia e si distingue dagli altri competitor del settore per la possibilità di prenotare viaggi 'di lusso' altamente personalizzabili e per la cura del cliente. Mira infatti ad una fetta di mercato costituita da persone con un elevato budget di viaggio che prediligono il comfort e la bellezza delle strutture alla ricerca di offerte meno costose.

Attualmente Turisanda fa parte del gruppo Alpitour già da diversi anni, infatti l'analisi presentata farà riferimento a Turisanda Eventi, ovvero una sezione del motore di ricerca dedicata appunto all'inserimento di eventi all'interno dei pacchetti viaggio. Lo scopo era quello di aggiungere la sezione citata, comprensiva di charter, linee charterizzate e prepack, oltre che al classico soggiorno singolo.

L'idea è quella di un layout sviluppato su due colonne: quella di sinistra adibita alla scelta della macro-categoria di evento (musicale, sportivo, teatrale, ecc),

mentre quella di destra diventa utile per scegliere l'evento nello specifico dopo aver deciso l'ambito.

E' stato deciso di implementare questa funzione per vari motivi, tra questi sicuramente spiccano per importanza il miglioramento della User Experience e soprattutto l'aumento delle vendite del canale e quindi il ritorno economico.

Il progetto si è svolto nell'arco di circa otto mesi attraverso 25 Sprint, con una durata totale effettiva di 1720 h. Nel grafico a torta successivo sono presentati i work item principali del progetto.

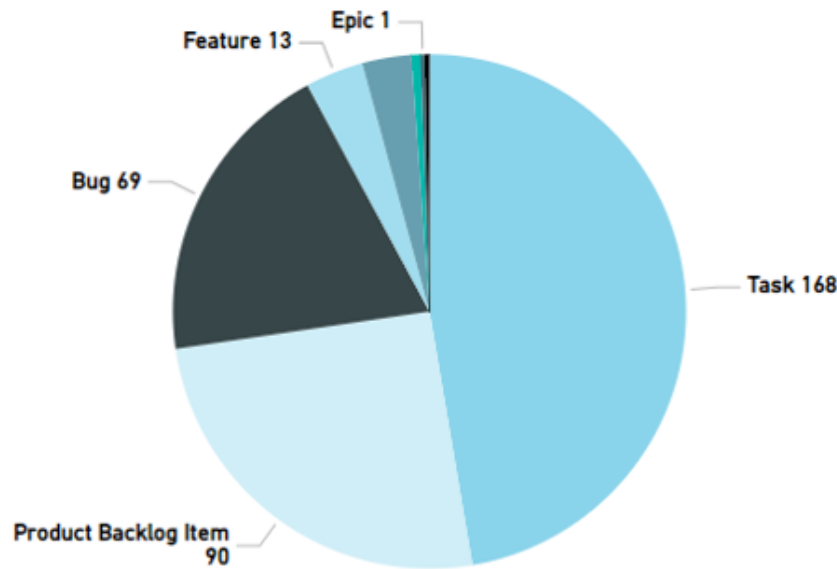


Figura 3.19: Work Items Turisanda

Per visualizzare l'andamento del progetto in termini di svolgimento delle attività pertinenti, è interessante osservare il Burndown Chart complessivo, che comprende l'intera durata del progetto.

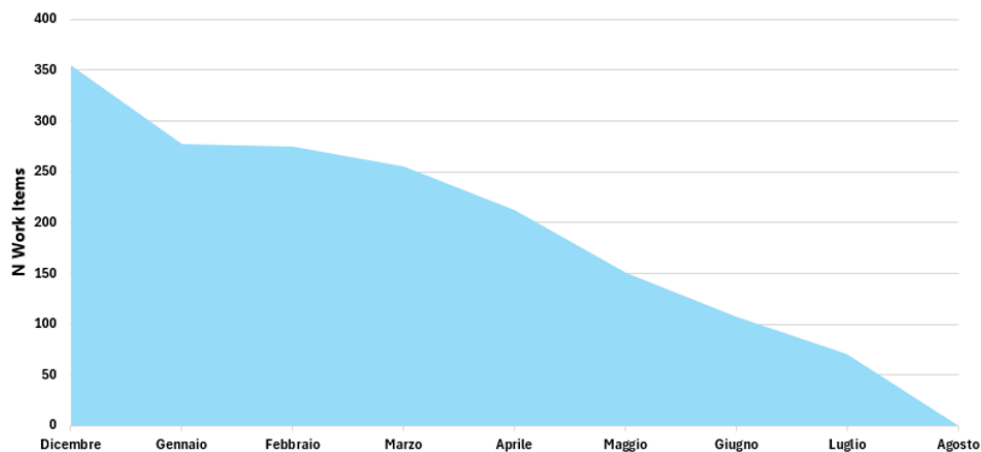


Figura 3.20: Burndown Chart Turisanda

Il grafico sembra rispecchiare le caratteristiche tipiche descritte nella sezione 2.2.3, non destando alcuna preoccupazione, nonostante mostri un evidente rallentamento nei mesi di Gennaio e Febbraio.

Numero di PBI svolti nella durata di uno Sprint

Anche in questo progetto sono stati svolti vari work items diversi. Sapendo che gli Sprint progettuali sono 25 si può calcolare il KPI in analisi come:

$$NPBIMedio = \frac{NTotPBI}{NTotSprint} = \frac{90}{25} = 3,6$$

Il dato è positivo, ma comunque nella media in quanto è possibile affermare che sono stati completati più di tre PBI durante ogni Sprint. (quasi il doppio rispetto al progetto presentato precedentemente).

Scostamento dell'effort



Figura 3.21: Effort e Completed work a confronto

I numeri sovrastanti propongono un confronto tra le ore totali dedicate al progetto e le ore effettivamente utilizzate per lo sviluppo delle attività. Anche in questo caso, come nel precedente, lo scostamento tra Effort pianificato e Completed Work è > 0 (390h).

In questo caso l'indice di scostamento dell'effort è dato da:

$$IndiceAccuratezzaEffort = \frac{EffortReale}{EffortPianificato} = \frac{1720h}{2100h} = 0,8190 = 81,9\%$$

Analizzando nel dettaglio le tendenze di qualche work item esemplare:

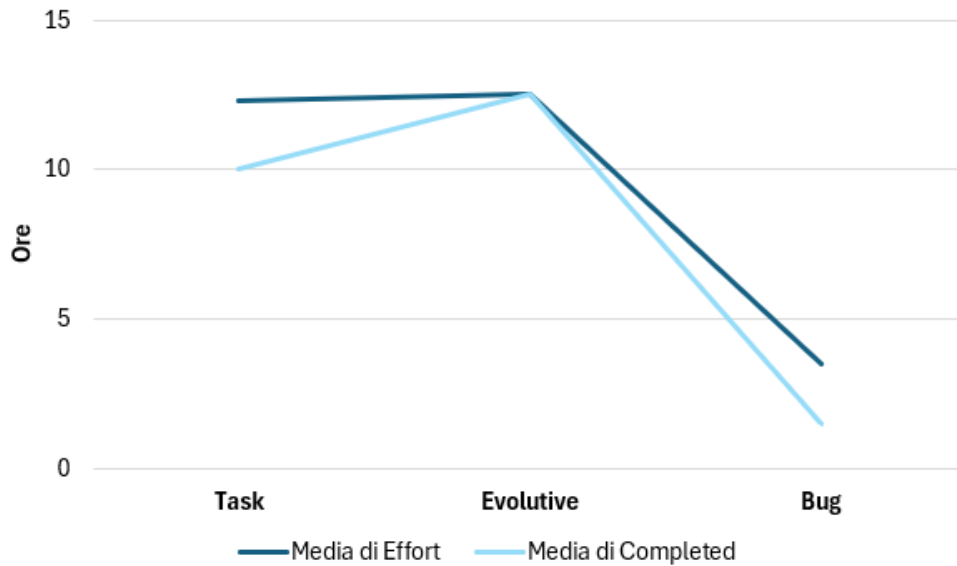


Figura 3.22: Effort e Completed work a confronto per Work item

L'effort richiesto allo svolgimento delle Evolutive è stato previsto in maniera ottima tant'è che lo scostamento relativo a questa specifica categoria di item è nullo. Il discorso però non vale per task e bug i quali hanno un completed work inferiore all'effort, e sono quindi stati sovrastimati, causando una sub-allocazione delle risorse. Più precisamente è possibile affermare che in media, per i task è stato riservato uno slot temporale di 12,5 h, ma per l'attuazione ne sono bastate solamente 10. In modo analogo per i bug sono state riservate in media 3,5 h, numero maggiore di 1,5, ovvero le ore utili all'esecuzione.

Lead Time

Il sottostante grafico a barre rappresenta il Lead Time del progetto Turisanda (eventi) suddiviso per il tipo di work item. Il grafico sembra ordinario, rispecchiando una situazione tipica (lead time decrescente al crescere del livello di dettaglio dei work items).

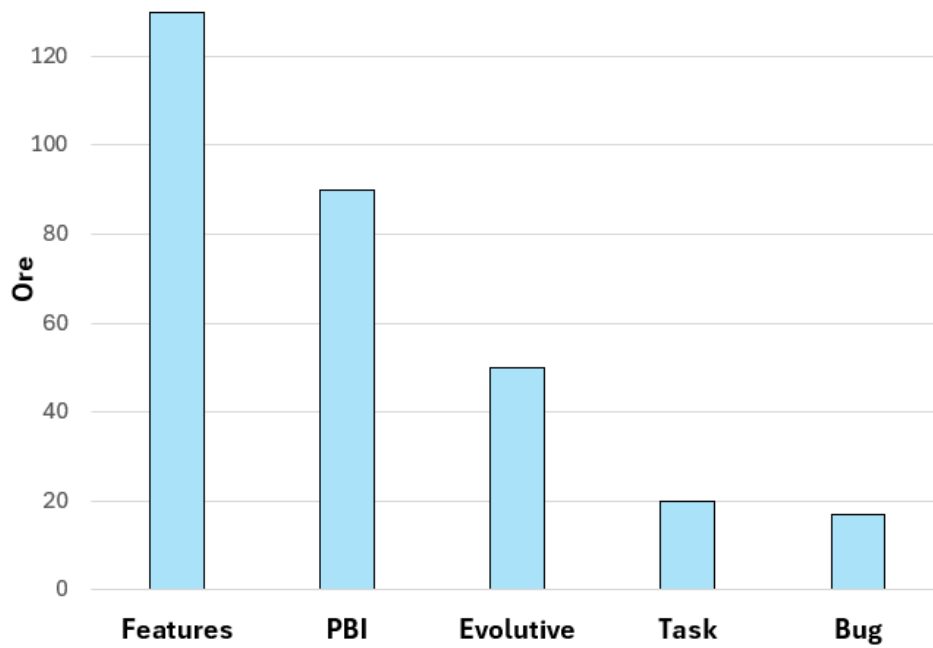


Figura 3.23: Lead Time Turisanda

Considerando i seguenti lead time,

- Features: 130h
- PBI: 90h
- Evolutive: 50h
- Task: 20h
- Bug: 17h

è possibile calcolare il tempo medio impiegato dal generico work item, che risulta essere 61,4 ore, che corrispondono a 8,77 giorni lavorativi.

Cycle Time

Nel seguente grafico sono riportati i valori medi di cycle time divisi per tipo di work item:

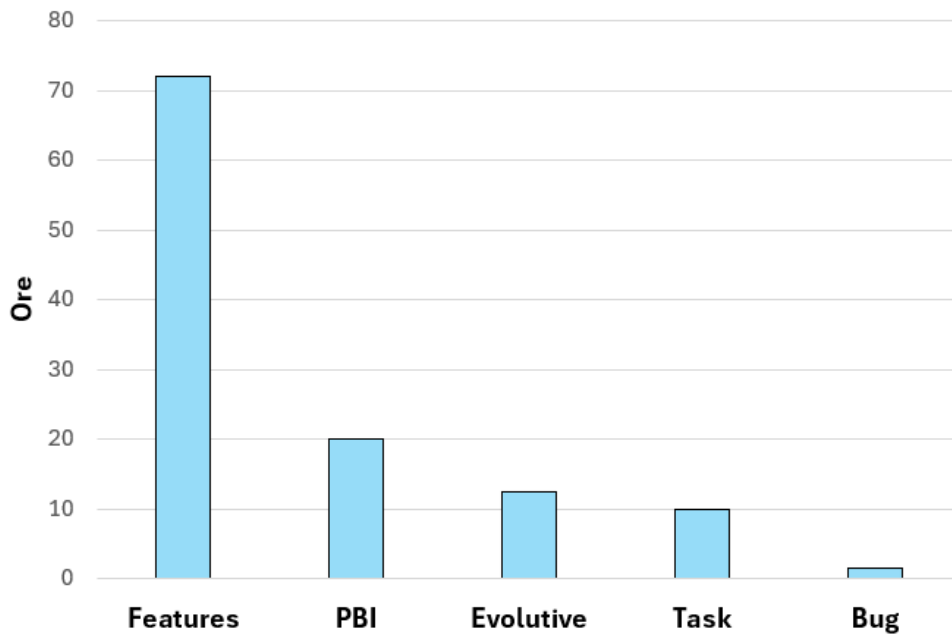


Figura 3.24: Cycle Time Turisanda

Si nota come ci sia una grande differenza tra i due indicatori presentati, che comunicano dei tempi di attesa in cui le attività non vengono lavorate sebbene siano state schedulate. Questo può essere dovuto a diverse cause, tra le quali le risorse limitate o problemi di coordinamento interno.

Considerando i seguenti valori di cycle time:

- Features: 72h
- PBI: 20h
- Evolutive: 12,5h
- Task: 10h
- Bug: 1,5h

si determina il cycle time medio generale, indipendente dal tipo di item, che risulta essere 23,2 ore, ovvero 3,31 giorni lavorativi.

3.5.3 Nuovo Flusso

L'azienda utilizza Travelgate, una piattaforma di gestione dei viaggi aziendali che offre soluzioni per semplificare la pianificazione e la prenotazione da parte dell'azienda tramite la quale è possibile:

- Prenotare un viaggio;
- Gestire le spese;
- Personalizzare itinerari;
- Supporto e assistenza.

Il progetto denominato 'Nuovo flusso' si propone appunto di realizzare un flusso di distribuzione denominato "Banca Letti", che consentirà ai clienti di cercare e prenotare prodotti SAT (Servizi Aggiuntivi Turistici) offerti da Alpitour attraverso diverse marche.

Per raggiungere questo obiettivo, è stato deciso di implementare un set di API (Interfacce di Programmazione Applicativa) nel formato richiesto da Travelgate. Ciò contribuirà ad aumentare il livello di compatibilità dell'integrazione tra i sistemi, facilitando la ricerca e la prenotazione di prodotti SAT di Alpitour attraverso le piattaforme di Travelgate.

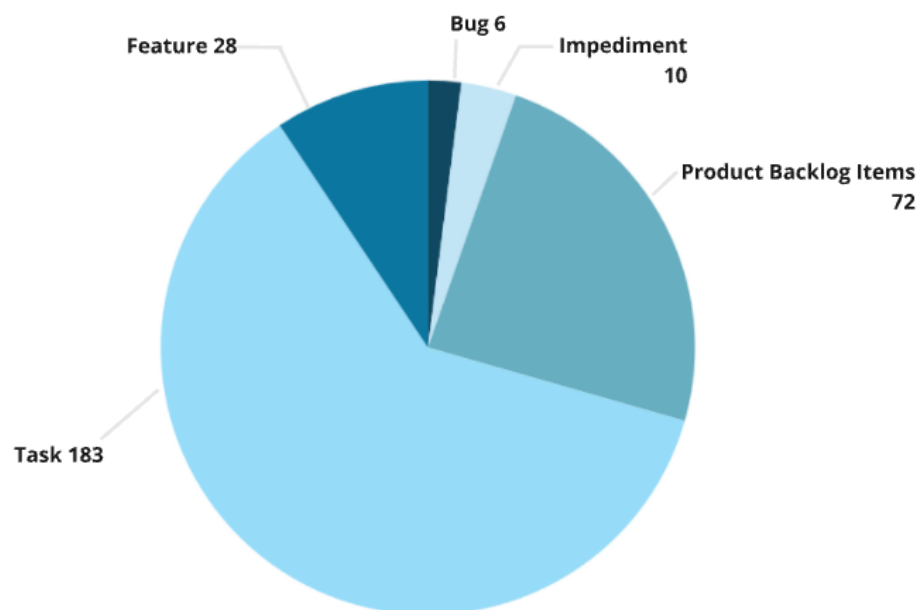


Figura 3.25: Grafico a torta relativo ai Work Items di Nuovo Flusso

Nuovo flusso è un progetto che si è svolto nell'arco di poco più di 2100 ore nelle quali sono state svolte le attività mostrate nel grafico 3.25.

Numero di PBI svolti nella durata di uno Sprint

Lo svolgimento è avvenuto attraverso 27 Sprint, dato che permette di calcolare il numero di PBI medio:

$$NPBIMedio = \frac{NTotPBI}{NTotSprint} = \frac{72}{27} = 2,67$$

Dato che non desta preoccupazioni, in quanto sia un valore in media con le tendenze aziendali.

Scostamento dell'effort



Figura 3.26: Effort e Completed work a confronto

Il confronto tra le ore effettive e quelle preventivate ricalca quella che sembra essere una costante tra i vari progetti IT aziendali, ovvero la sovrastima della capacità oraria da dedicare allo svolgimento delle attività. Anche in questo caso lo scostamento è > 0 (310h) e l'indice è uguale a:

$$IndiceAccuratezzaEffort = \frac{EffortReale}{EffortPianificato} = \frac{2620h}{2310h} = 0,8816 = 88,16\%$$

Facendo particolare attenzione alle attività più dettagliate dello svolgimento, tipicamente in grande numero, si vede ancora una volta una previsione imprecisa, anche se di poco, perché troppo elevata.

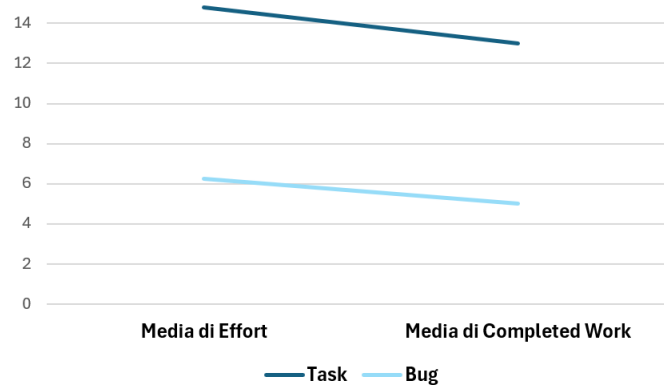


Figura 3.27: Effort e Completed work di task e bug a confronto

Per i task è stato riservato in media uno slot di circa 15 ore e ne sono bastate 13 per l'esecuzione, fornendo un indice di accuratezza di:

$$IndiceAccuratezzaEffort = \frac{EffortReale}{EffortPianificato} = \frac{13h}{15h} = 0,8667 = 86,68\%$$

Che non è negativo come risultato, ma considerando il numero sempre molto elevato di task, sarebbe opportuno che la previsione di questi fosse più precisa.

Discorso analogo per i Bug, per i quali le previsioni di effort sono state troppo elevate di circa un'ora, ma che essendo solo 6, non causano troppi problemi.

Lead Time

Il grafico a barre rappresenta il Lead Time dei principali Work Items del progetto Nuovo Flusso.

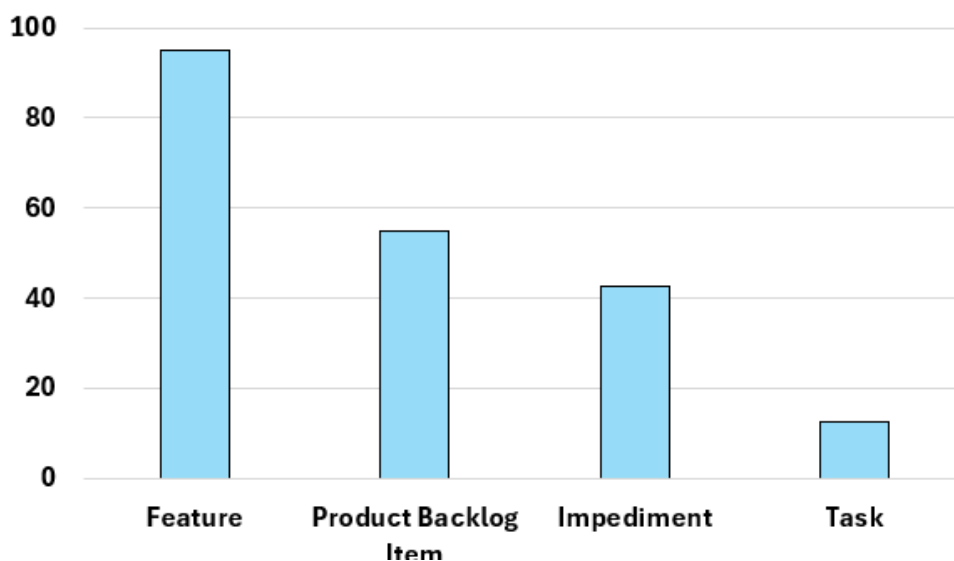


Figura 3.28: Grafico a barre del Lead Time per Work Item di Nuovo Flusso

Una differenza rispetto alle situazioni dei progetti precedenti è che in figura sono presenti degli Impediment. Nelle altre analisi non ne sono stati trovati, in quanto lo svolgimento non ne ha richiesti. Qui alcune attività sono state bloccate a causa di mancanze o di ritardi di altri comparti, rallentando l'esecuzione. In uno svolgimento ideale non si trovano Impediment durante il percorso, ma purtroppo questi non sono prevedibili e può capitare di doverne affrontare.

Considerando come Lead time i seguenti:

- Feature: 95h;
- PBI: 55h;
- Impediment: 42,5h;
- Task: 12,5h.

si trova un Lead Time medio di 51,25 ore, che corrispondono a 7,32 giorni di lavoro. Il dato si discosta da quello del numero di PBI svolti all'interno di uno Sprint in quanto da quel calcolo sembrava che lo svolgimento delle attività fosse rapido e fluido, ma da questo si denota un lead time medio molto elevato, che dimostra che non tutti i work item di un determinato progetto procedono allo stesso modo.

Cycle Time

E' riportato il grafico dei tempi ciclo dei principali item progettuali:

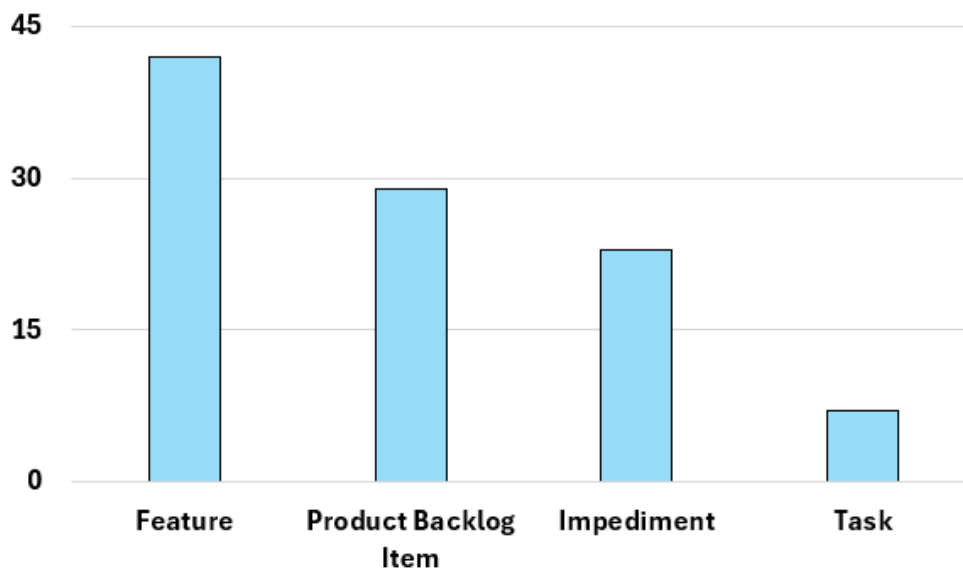


Figura 3.29: Grafico a barre del Cycle Time per Work Item di Nuovo Flusso

Qui nuovamente è molto evidente la presenza di tempi ciclo molto elevati in relazione agli Impediment, i quali hanno richiesto ben 42,5 ore in media per essere risolti. La discrepanza presente tra Lead Time e Cycle time di questo tipo di item comunica che questi siano rimasti molto in attesa prima che iniziasse il loro sviluppo, e che comunque sono stati di onerosa risoluzione. Il dato che dice che sono di

più gli Impediment dei Bug, insieme al fatto che gli impediment non fossero prontamente risolti, porta come conseguenza che i problemi non fossero relativi proprio allo sviluppo, bensì alla gestione delle attività e al coordinamento interno.

Considerando i tempi ciclo degli item come:

- Feature: 42h;
- PBI: 29h;
- Impediment: 23h;
- Task: 7h.

si calcola il tempo ciclo medio che risulta essere 25,25 ore o 3,6 giorni lavorativi.

3.5.4 Via Milano Parking



Figura 3.30: Logo Via Milano Parking

Il progetto denominato 'Via Milano Parking' è un progetto business con richiesta di servizi IT, di piccole dimensioni. La prenotazione del posteggio nel pacchetto viaggio avveniva tramite una mail automatica ricevuta dai fornitori, che dovevano inserire manualmente la prenotazione all'interno dei propri sistemi. Il cliente, una volta tornato dalla vacanza, nonostante abbia già pagato la sosta, deve comunque accedere al totem help desk e scannerizzare il voucher che attesta il suo avvenuto pagamento. Successivamente riceverà un ticket che dovrà inserire all'uscita del parcheggio.

L'obiettivo che il progetto si pone è quello di Integrare i sistemi di booking e easy book con quelli degli scali di Milano Malpensa, Linate e Bergamo Orio al Serio in modo da fornire al cliente un pin al momento della prenotazione stessa, da digitare sulla colonnina di ingresso per ottenere un ticket utile per uscire dal posteggio, al rientro dalla vacanza, senza dover passare per il totem help desk.

Il progetto ha avuto uno svolgimento differente dagli altri poiché, viste le ridotte dimensioni e il fatto che fosse di tipo cross-team, non è stato organizzato in Sprint, ma è stato pianificato uno Scrum of Scrums settimanale (in linea di massima rispettato, con alcune eccezioni). Esso è nato il 24/10/2023 ma gli sviluppi sono cominciati solo ad inizio 2024, per essere poi concluso il 31/07/2024 in seguito a 15 SOS. Ecco qui una rappresentazione dei principali work items utilizzati:

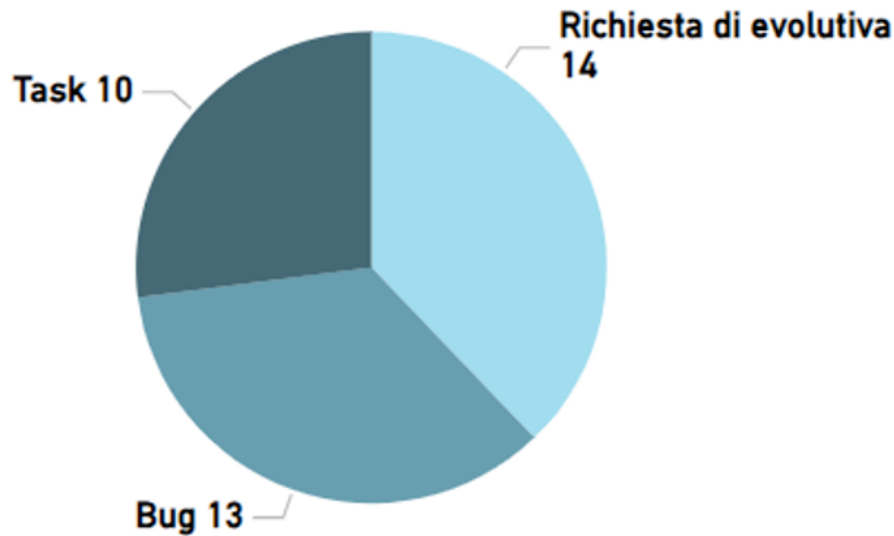


Figura 3.31: Grafico a torta dei work item di Via Milano Parking

Numero di PBI svolti nella durata di uno Sprint

In questo caso l'indicatore perde di significato non essendoci state iterazioni di sprint. E' possibile però adattare il KPI allo svolgimento del progetto contando, non più il numero di sprint, ma il numero dei SOS, in modo da valutare il numero medio di attività svolte di volta in volta. Un'altra caratteristica che rende il progetto particolare è che non sono stati predisposti dei PBI, ma solamente dei task in quanto si trattava di un progetto abbastanza semplice e diretto. Nel calcolo quindi si considera il seguente indicatore:

$$NWorkItemMedio = \frac{NTotWorkItem}{NTotSOS} = \frac{37}{15} = 2,46$$

Dato, sebbene non esattamente comparabile con quelli ottenuti nei progetti precedenti, comunque utile all'analisi della velocità di svolgimento, il quale comunica che nella distanza tra una riunione e l'altra sono stati svolti in media più di due work items.

Scostamento dell'effort

Per l'analisi dello scostamento dell'effort, si osservano a confronto l'effort riservato e l'effettivo completed work dei principali items riportati in Figura 3.31 tramite un grafico a linee:

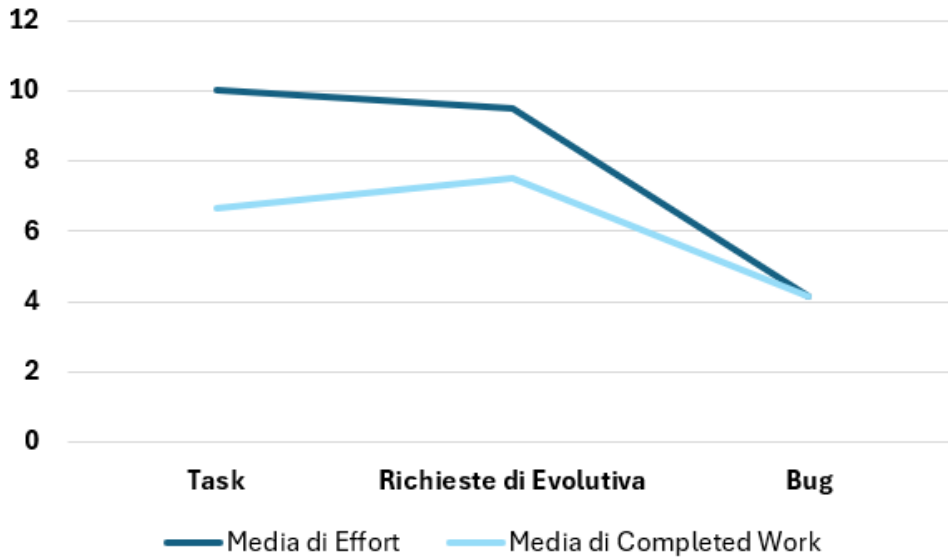


Figura 3.32: Grafico di Effort e Completed Work di Via Milano Parking

Anche in questo caso l'effort allocato per lo svolgimento supera l'effettivo tempo impiegato. Nello specifico si vede come i bug siano stati effortati alla perfezione, mentre per i Task e per le Evolutive ciò non sia avvenuto. Si calcola l'indice:

$$IndiceAccuratezzaEffortTask = \frac{EffortReale}{EffortPianificato} = \frac{6,67h}{10h} = 0,667 = 66,67\%$$

Il quale non risulta particolarmente soddisfacente in quanto troppo basso. Nel caso del progetto ciò non ha portato a particolari danni o rallentamenti per via delle ridotte dimensioni del progetto. Effortare con questa precisione un progetto comprendente centinaia di task potrebbe risultare dannoso per l'azienda non sfruttando appieno la capacità oraria dei lavoratori.

Lead Time

E' riportato il grafico a barre relativo ai Lead Time del progetto:

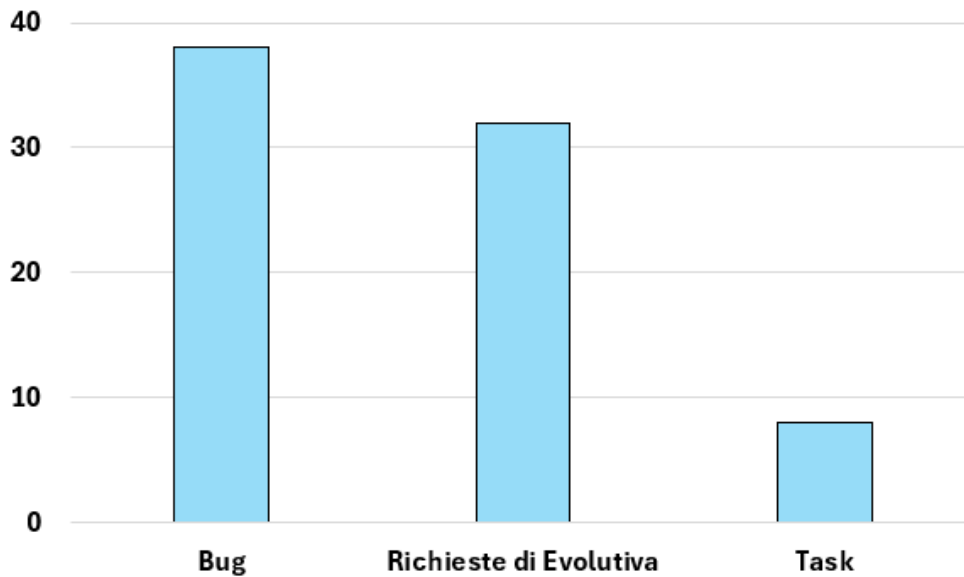


Figura 3.33: Grafico a barre sul Lead Time degli item di Via Milano Parking

Si nota come il tempo relativo a Bug ed Evolutive sia molto superiore rispetto a quello dei task. Ciò comunica che in fase di sviluppo non si sono gestiti in maniera ottimale gli imprevisti, sia che fossero dovuti a malfunzionamenti, sia ad aggiornamenti in corso d'opera.

I lead time elevati però, sebbene siano una nota negativa, lasciano grande spazio ai miglioramenti che, se apportati, possono garantire un'ottimale flusso di lavoro e rendere l'attività aziendale complessivamente migliore.

Considerando i seguenti valori medi di Lead time:

- Bug: 38h;
- Evolutive: 32h;
- Task: 8h;

Si calcola il Lead time medio che risulta essere di 26 ore, che corrispondono a 3,71 giorni lavorativi.

Cycle Time

Il grafico a barre presentato riporta i valori medi di Cycle time per tipo di item:

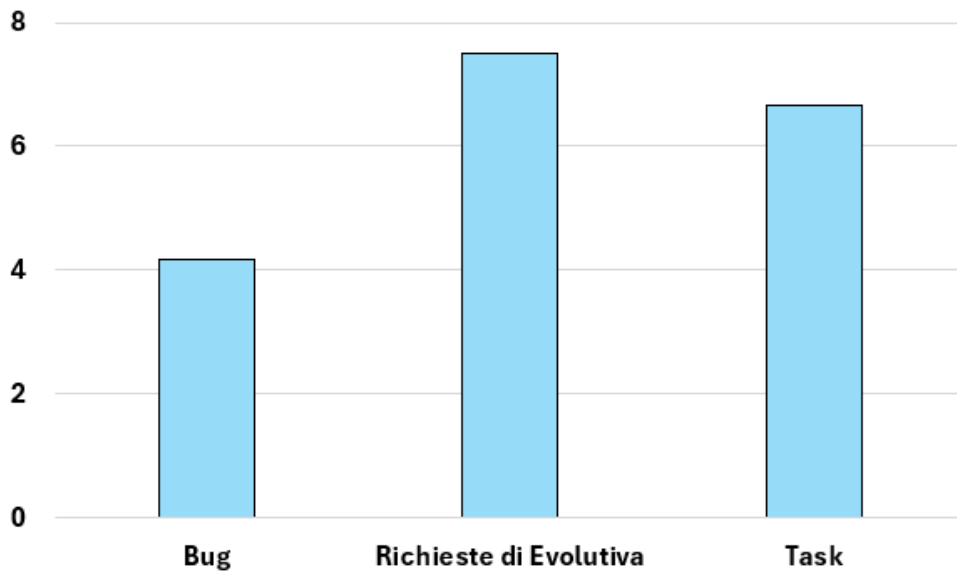


Figura 3.34: Grafico a barre sul Cycle Time degli item di Via Milano Parking

Questi valori, in relazione ai lead time visti precedentemente forniscono un'importante informazione in fase di analisi. E' importante fare attenzione alla differenza presente tra i tempi precedenti e quelli in Figura 3.34. I Lead time di Bug ed Evolutive sono di molto superiori rispetto ai tempi trascorsi per la sola lavorazione segnalando dei lunghi periodi di attesa o inattività, dovuti molto probabilmente a problemi di coordinamento interno o a colli di bottiglia.

Il dato riguardante i task è invece rassicurante, essendo che il tempo di inattività di questa tipologia di item non raggiunge nemmeno le 1,5 ore.

Considerando i seguenti valori medi di Cycle time:

- Bug: 4,16h;
- Evolutive: 7,5h;
- Task: 6,67h;

Si calcola il valore medio generico corrispondente a 6,11 ore, cioè 0,87 giorni.

3.5.5 Confronto dei KPI raccolti

Per avere sotto controllo tutti i KPI insieme e poter effettuare una valutazione è opportuno formulare una tabella che raccoglie i dati trovati.

	KPI			
	Numero di PBI svolti in uno sprint	Accuratezza dell'effort	Lead Time	Cycle Time
WebCola	1,93	Generico: 90,91% Task: 95,2% Bug: 98%	27,06 h	18,82 h
Turisanda	3,60	Generico: 81,90% Task: 68,9 % Bug: 40%	61,4 h	23,2 h
Nuovo flusso	2,67	Generico: 88,16% Task: 86,68% Bug: 80,6%	51,25 h	25,25 h
Via Milano Parking	2,46*	Generico: \ Task: 66,67% Bug: 100%	26 h	6,11 h

*Valore relativo al generico Work Item e non propriamente ai PBI.

Figura 3.35: Tabella contenente i KPI trovati

Dalla tabella emergono alcune considerazioni interessanti da fare in relazione ad ognuno dei progetti.

In merito al **numero di PBI svolti in uno Sprint** si può affermare che Web-Cola ha il valore più piccolo (1,93), il che potrebbe indicare un focus su task più complessi o una bassa efficienza nello svolgimento dei singoli item. Turisanda ha invece il numero maggiore (3,60) che suggerisce una buona capacità di svolgimento delle attività. Via Milano Parking è più difficile da confrontare sotto questo aspetto in quanto il valore riportato non faccia riferimento ai PBI ma ai generici work item.

Per quanto riguarda l'**accuratezza dell'effort** invece, WebCola ha un'ottima accuratezza per tutte le tipologie di task indicando un buon livello di previsione dell'effort necessario per completare i lavori. Turisanda ha più variabilità, avendo un'inferiore accuratezza per quanto riguarda i task (68,9%) e una molto bassa precisione per i bug (40%) suggerendo possibili problemi di gestione o di comprensione delle complessità. Sia per Nuovo Flusso che per Via Milano Parking si individuano delle aree di miglioramento. Nel primo caso gli indicatori non sono negativi, mentre

nel secondo c'è ampio margine per quanto riguarda la precisione nei task, solo al 66,67%.

In merito ai **Lead Time** WebCola e Via Milano Parking hanno lead time simili (rispettivamente 27,06 h e 26 h), suggerendo efficienza nei tempi di completamento del processo rispetto agli altri. Il tempo di Turisanda è il più elevato che troviamo (61,4 h), il che suggerisce possibili inefficienze o complessità nel processo di completamento, tuttavia bisogna dire che l'indicatore dipende anche dalla natura delle attività. Se i Work Item del progetto hanno in media una complessità maggiore, è normale che il Lead Time sia più elevato rispetto ad altri progetti nei quali le attività schedate sono molto semplici.

Tenendo a mente quest'accortezza, si vede come il **Cycle Time** di Via Milano Parking sia per distacco il più basso (6,11 h), il che significa che la lavorazione diretta è molto rapida rispetto al lead time, segnalando un grande periodo di attesa prima dell'inizio della lavorazione, forse dovuto anche alla bassa urgenza del progetto.

WebCola presenta un cycle time di 18,82 h ma con una differenza più ridotta rispetto a Via Milano Parking.

Turisanda e Nuovo flusso hanno cycle time più elevati (rispettivamente 23,2 h e 25,25 h), in linea con i loro lead time più lunghi, suggerendo che sia la lavorazione effettiva sia i tempi di attesa sono più alti.

In generale, Via Milano Parking si distingue per il basso lead time e cycle time, indicando un processo più snello e probabilmente un'ottimizzazione delle risorse. Per Turisanda l'azienda sembra avere avuto difficoltà a prevedere accuratamente l'effort, in particolare sui bug, e registra il lead time più alto. Potrebbe essere utile in futuro, effettuare una revisione dei processi di stima e gestione delle attività per progetti cross-team di questo tipo, in modo da massimizzare l'efficienza.

Inoltre, guardando i dati di Turisanda e Nuovo Flusso l'idea che sorge è che talu progetti abbiano dei colli di bottiglia che danno origine a tempi di attesa molto maggiori. In sintesi, l'analisi suggerisce che WebCola e Via Milano Parking abbiano processi più efficienti in termini di gestione del tempo, mentre Turisanda potrebbe trarre vantaggio da una revisione più approfondita dei suoi processi di gestione dei task e dei bug.

Conclusioni

Ricapitolando quanto analizzato, in seguito alle sezioni riservate alla teoria dell'Agile, nelle quali è stato mostrato sia il confronto tra le tecniche di gestione tradizionali e innovative sia la struttura dei principali framework relativi a queste ultime, si è visto come l'impatto dell'implementazione di Agile in un sistema che non è abituato, possa portare a dei miglioramenti ma anche alla creazione di problemi.

Dalla survey effettuata nella Sezione 3.2 sono emersi dei dati interessanti. Tra i dati positivi, spiccano sicuramente la conoscenza diffusa del metodo e la familiarità con esso, sintomi del fatto che le tecniche utilizzate siano ormai divenute parte integrante e consolidata del sistema.

A rafforzare quest'ultima affermazione è il fatto che una delle principali difficoltà riscontrate dai dipendenti, fosse proprio la scarsa documentazione fornita dall'azienda e la scarsa formazione in materia effettuata. Ciò significa che, nonostante le difficoltà iniziali di implementazione, emerse anch'esse dalla survey ed evidenziate dal test di Kruskal-wallis, il metodo è stato ormai metabolizzato dal reparto e i dipendenti stanno diventando via via sempre più preparati, migliorando costantemente il processo.

Inoltre, dai dati e dall'esperienza personale trascorsa nell'area IT in questione, si nota anche un forte entusiasmo nei confronti delle riunioni periodiche previste dai framework utilizzati, anche se talvolta non sono sfruttate nel migliore dei modi. Nonostante questo, quasi tutti ritengono che siano assolutamente indispensabili. Parlando degli aspetti negativi, oltre alla formazione inefficace appena citata, è emerso malcontento riguardante le strutture e i metodi di comunicazione, specialmente nei progetti cross-team, nello svolgimento dei quali i diversi team faticano a coordinarsi per riuscire a effettuare uno svolgimento ottimale, e non riescono a ridurre i lead time di progetto. Anche dal test di Kruskal-Wallis è emersa questa caratteristica in relazione alla categoria degli analisti, i quali risentono maggiormente delle difficoltà di comunicazione nel corso del proprio lavoro. Una minore caratteristica negativa, risiede nella difficoltà di aggiornamento dei backlog per i membri dei team i quali, per dimenticanza o per mancanza di conoscenza, non aggiornano tempestivamente il backlog di prodotto relativo allo Sprint, non permettendo a chi di dovere di monitorare lo stato di avanzamento dei lavori correttamente.

Dalla conclusiva analisi dei KPI è emersa una notevole differenza tra i Lead Time e i Cycle Time medi, che segnala un tempo di attesa tra la schedulazione e l'inizio dello svolgimento delle attività non indifferente. Nei progetti cross-team (Turisanda e Via Milano Parking) questa differenza è ancor più marcata ed è attribuibile alle difficoltà di comunicazione e coordinamento citata precedentemente. Anche negli altri progetti la differenza è grande, ma non così tanto significativa.

Un'altro particolare che sarebbe utile migliorare per trasferire un miglioramento ai processi, è dato dalla previsione di effort delle attività. Se gli interessati dedicassero più tempo ad effettuare un'analisi approfondita dei Work Item, fornendo previsioni più accurate, si otterrebbe un'assegnazione ottimale del numero di attività per ciascuno Sprint. Questo consentirebbe di sfruttare al meglio la capacità lavorativa disponibile, evitando di assegnare compiti che richiedono meno risorse del previsto.

In conclusione, si può affermare che il metodo Agile è ormai saldamente radicato nella gestione dei progetti software e sempre più aziende lo stanno adottando.

Sebbene la sua integrazione risulti complessa, generando talvolta attriti interni e disfunzioni nel processo, i benefici che apporta sono numerosi e possono essere progressivamente ottimizzati grazie all'esperienza e al miglioramento continuo, al fine di raggiungere un processo quasi perfettamente efficiente.

Ringraziamenti

Desidero ringraziare innanzitutto il professor Giulio Mangano, il mio relatore, per la guida e il supporto datomi durante questo progetto.

Ringrazio inoltre l'intera Area IT dell'azienda Alpitour S.P.A., non solo per avermi fornito i dati necessari, ma soprattutto per la grande disponibilità e collaborazione mostrata nei miei confronti durante l'intero periodo trascorso con loro.

Un ringraziamento speciale ai miei coinquilini Marco e Stefano, nonché migliori amici da ormai tanti anni, che hanno reso questi anni indimenticabili. La loro presenza, il supporto e le risate condivise mi hanno accompagnato nei momenti più intensi, rendendo leggere anche le giornate più impegnative. Senza di voi, questo percorso non sarebbe stato lo stesso.

Il ringraziamento più importante però va alla mia famiglia.

Un grazie ai miei genitori, che mi hanno sempre sostenuto e hanno creduto in me incoraggiandomi e spronandomi ad affrontare questo percorso e dare il meglio di me in ogni situazione. Un grazie a mio fratello che, con le sue poche ma dirette parole, mi ha sempre aiutato a non mollare e ad andare fino in fondo. Grazie a mia nonna che con il suo affetto e la sua saggezza è stata una fonte costante di ispirazione. Un grazie di cuore va ad Elisa, la mia fidanzata, che con il suo affetto e la sua comprensione mi è stata accanto nei momenti più complessi.

Senza di voi tutto questo non sarebbe stato possibile.

Bibliografia

- [1] David J. Anderson. *Kanban: Successful Evolutionary Change for Your Technology Business*. David J. Anderson, 2010.
- [2] Russell D. Archibald. *Project management. La gestione di progetti e programmi complessi*. Franco Angeli, 2003.
- [3] Pier Luigi Guida. *Il Project Management secondo le norme UNI ISO 21500 e 21502*. Franco Angeli, 1985.
- [4] Project Management Institute. *A Guide to the Project Management Body of Knowledge 6TH edition*. Project Management Institute, 2018.
- [5] Daniel Jones James P. Womack, Daniel Roos. *The Machine That Changed the World: The Story of Lean Production*. James P. Womack, Daniel Roos, Daniel Jones, 1990.
- [6] Corey Ladas. What is scrumban? *Agile Alliance*, 2021.
- [7] Antonio Nieto-Rodriguez. evoluzioni del project management:da waterfall ad agile, con digitale e change management. *Il project Manager*, 2024.
- [8] Daniel Raymond. Cons or disadvantages of agile methodology. *projectmanagers*, 2023.
- [9] Dave Ross. Finding your fit: How to get started with hybrid agile. *Miro*, 2024.
- [10] Ken Schwaber and Jeff Sutherland. *The Scrum Guide*. Share-Alike license of Creative Commons, 2010.
- [11] StarAgile. Agile methodology: Advantages and disadvantages. *StarAgile Blog*, 2023.
- [12] Hirotaka Takeuchi and Ikujiro Nonaka. The new new product development game. *Harvard Business Review*, 1986.
- [13] Kent Beck Mike Beedle Arie van Bennekum Alistair Cockburn Ward Cunningham Martin Fowler James Grenning Jim Highsmith Andrew Hunt Ron Jeffries Jon Kern Brian Marick Robert C. Martin Steve Mellor Ken Schwaber Jeff Sutherland Dave Thomas. *Manifesto for Agile Software Development*. Kent Beck Mike Beedle Arie van Bennekum Alistair Cockburn Ward Cunningham Martin Fowler James Grenning Jim Highsmith Andrew Hunt Ron Jeffries Jon Kern Brian Marick Robert C. Martin Steve Mellor Ken Schwaber Jeff Sutherland Dave Thomas, 2001.