



UNIVERSITÀ POLITECNICA DELLE
MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Gestionale

Tecniche di gestione agile dei progetti

Agile project management techniques

Relatore: Chiar.mo
Prof. Maurizio Bevilacqua

Tesi di laurea di:
Beatrice Brancatello

Anno Accademico 2019/2020

A Gabriele, il Sole della mia vita

Indice

Indice	1
Introduzione	2
Che cos'è il project management	4
Il progetto e il <i>Project Management</i>	4
Dai principi <i>lean</i> all'approccio agile nel project management	9
Waterfall e Agile a confronto	19
Sequenzialità vs iterazioni	19
Pianificazione vs adattamento	22
L'agile più utilizzato: il metodo Scrum	25
Lo Scrum Team	28
Il Product Owner	29
Il Team di Sviluppo	30
Lo Scrum Master	31
Gli artefatti	31
Il Product Backlog	32
Lo Sprint Backlog	33
L'incremento	34
Gli eventi	35
Scrum e waterfall: quale scegliere	42
Conclusioni	47
Bibliografia	48

Introduzione

L'uomo si è dedicato alla realizzazione di progetti fin dalle proprie origini. L'esigenza di definire metodologie che consentono di gestirli e completarli nel modo più efficiente ed efficace più possibile però è molto più recente. Fino ai primi del '900 si credeva infatti che il modo migliore per aumentare la produttività fosse lavorare più a lungo e più duramente.

La metodologia di project management tradizionale nasce in quegli anni con il taylorismo e con l'ideazione di alcuni strumenti tipici di questa disciplina - prima il diagramma di Gantt, poi il PERT e il CPM.

L'introduzione di metodologie per gestire i progetti risulta talmente vantaggiosa da essere impiegata nell'ambito del programma spaziale americano e conduce alla creazione di una vera e propria disciplina, che verrà formalizzata a partire dagli anni '60.

La tradizionale metodologia di project management - metodologia *waterfall* - fornisce strumenti e metodologie validi per la gestione di progetti, soprattutto nella fase di pianificazione e controllo delle attività. La sua applicazione permette di definire con esattezza quali attività saranno realizzate e in quale ordine e di elaborare documenti che attestano il lavoro svolto e i progressi raggiunti.

Dalla fine del XX secolo, si è cominciato ad avvertire come tali principi non fossero più così facilmente applicabili. Ciò risulta ancora più valido oggi, in un contesto caratterizzato dal continuo cambiamento tecnologico, sociale e di *business*.

Per rispondere alla nuova esigenza di gestire il cambiamento - e non subirlo - sono state sviluppate nuove metodologie di project management caratterizzate da maggiore flessibilità, le metodologie agili.

Tali metodologie si contraddistinguono per la possibilità di cambiare nel tempo i requisiti del progetto che si sta sviluppando, adattandolo a nuove esigenze che insorgono durante la fase di sviluppo senza perdere quanto prodotto fino a quel momento.

Il loro utilizzo e la loro popolarità sono cresciuti nel tempo, portando in particolare alla diffusione di alcune di esse, come la metodologia Scrum.

Un buon project manager però non può scegliere a priori quale metodologia utilizzare. Per assicurare il successo del progetto, deve adottare la metodologia di gestione che meglio si adatta alle sue caratteristiche.

Che cos'è il project management

Il progetto e il *Project Management*

L'umanità intraprende progetti sin dagli albori delle prime civiltà organizzate. La Grande Muraglia Cinese o le piramidi sono esempi di progetti complessi su larga scala. Il termine progetto viene utilizzato frequentemente all'interno delle nostre conversazioni quotidiane. Per parlare di gestione di progetti è opportuno iniziare dunque dalla definizione di che cosa è un progetto.

Secondo la definizione del Project Management Institute ("PMI"), un progetto è "un'attività temporanea intrapresa per realizzare un prodotto, servizio o risultato specifico". Il progetto si differenzia quindi dalle operazioni di un'organizzazione per la sua natura unica e temporanea. Definizioni simili sono fornite da altre fonti autorevoli:

"Un progetto è un'impresa complessa, unica e di durata determinata, volta al raggiungimento di un obiettivo prefissato, mediante un processo continuo di pianificazione, esecuzione e controllo di risorse differenziate e con vincoli interdipendenti di tempi-costi-qualità" - Istituto Italiano di Project Management

"Un progetto può essere definito come un insieme di attività complesse e interrelate, aventi come fine un obiettivo ben definito, raggiungibile attraverso sforzi sinergici e coordinati, entro un tempo predeterminato e con un preciso ammontare di risorse umane e finanziarie a disposizione" (Tonchia & Nonino, 2013).

Sebbene esistano diverse definizioni di "progetto" a seconda del punto di vista dell'autore, le caratteristiche che lo contraddistinguono sono grossomodo condivise, infatti un progetto si caratterizza per: l'unicità; la temporaneità; la relazione logica tra le azioni che lo compongono; i vincoli in termini di costi, tempo e qualità; la complessità (Wysocki, 2019). Da qui l'esigenza di una serie di metodologie e tecniche per la gestione dei progetti, il project management.

Il *project management* è definito infatti come “l'applicazione di conoscenze, attitudini, strumenti e tecniche alle attività di un progetto al fine di conseguire gli obiettivi” (Project Management Institute, 2003).

Se, come abbiamo scritto precedentemente, l'esigenza di realizzare progetti ha contraddistinto l'umanità fin dalle sue origini, la formalizzazione del project management è avvenuta solo in tempi recenti, con lo sviluppo di una vera e propria branca di studi nella seconda metà del XIX secolo.

Alla fine del 1800, negli Stati Uniti, la realizzazione di progetti infrastrutturali di grandi dimensione - quali la costruzione della prima linea ferroviaria transcontinentale avviata nel 1860 - comportò la necessità di organizzare ingenti quantità di risorse e di capitale in maniera efficiente e sinergica. Da qui, l'avvio allo sviluppo delle metodologie di project management. All'improvviso, i dirigenti di impresa si trovavano con la necessità di organizzare il lavoro di migliaia di operai e di processare e assemblare quantità di materie prime come mai prima di allora (Watt, 2014).

In tale contesto, l'ingegnere meccanico statunitense Frederick Taylor (1856-1915) iniziò a studiare il metodo attraverso cui il lavoro veniva organizzato, con l'obiettivo di migliorare l'efficienza nella produzione. Fino a quel momento, si riteneva che l'unico modo per aumentare la produttività fosse quello di lavorare più a lungo e più duramente (PM Consulting, n.d.). Nel 1911, Taylor pubblica il suo trattato “Principi di organizzazione scientifica del lavoro” (“The Principles of Scientific Management”), in cui espone la teoria che diverrà nota come *taylorismo*. Tale sistema si basa sull'idea che qualsiasi operazione del processo produttivo possa essere scomposta in attività (*task*) e sotto attività (*sub task*) più semplici. Al manager spetta il compito di scomporre il processo in unità elementari ed assegnarne una a ciascun operatore, specificando in quanto tempo debba svolgerla. I principi del taylorismo trovarono applicazione nella pratica con l'introduzione della catena di montaggio da parte di Ford (Cassese, 2016).

Nel 1887, un altro ingegnere meccanico si unì a Taylor nei suoi studi sull'organizzazione scientifica del lavoro, Henry Gantt - considerato uno dei fondatori del project management moderno. Per ottimizzare i lavori di costruzione

delle navi da trasporto nella Prima guerra mondiale, Gantt sviluppò un diagramma a barre, che in suo onore prenderà il nome diagramma di Gantt (Tonchia & Nonino, 2013).

Un diagramma di Gantt è un diagramma a barre che illustra la pianificazione di un progetto ed è tuttora uno strumento di uso comune nel campo del project management (Watt, 2014). Strumenti ancora oggi in larghissimo uso nella gestione di progetti - in particolare complessi e pluriennali - si basano sulla gestione dei flussi di lavoro studiati e formalizzati in quell'epoca. Tra i principali, Microsoft Project, che inizia ad essere sviluppato dal 1987 (esattamente un secolo dopo il lavoro di Gantt) e tutt'oggi presenta il diagramma di Gantt come principale visualizzazione dei processi - insieme con strumenti PERT e CPM, che vedremo in seguito.

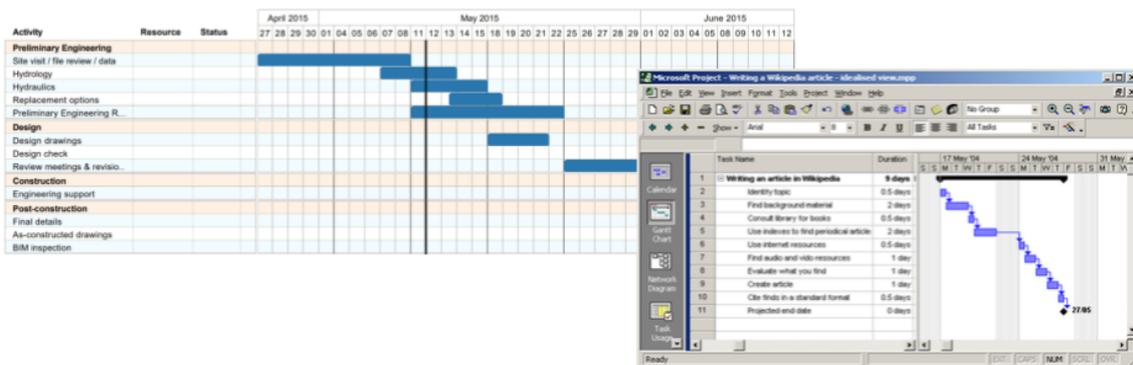


Figura 1 - Esempio di diagramma di Gantt (a sinistra) e diagramma di Gantt all'interno dell'applicativo MS Project (a destra). Fonte: projectengineering.net, wikipedia.com

Nei decenni successivi, a partire dagli anni '50, la crescente complessità dei progetti industriali, soprattutto in ambito militare e aerospaziale, stimolò lo sviluppo e la formalizzazione di nuove tecniche e strumenti per facilitare la gestione di progetti (Tonchia & Nonino, 2013). In particolare, risalgono a questo periodo i due principali modelli matematici per la programmazione delle attività di progetto, il *Critical Path Method* (CPM) e il *Program Evaluation and Review Technique* (PERT).

Il CPM e il PERT

Entrambe le tecniche consistono nell'analisi del reticolo di progetto: il modello logico matematico che rappresenta le attività di progetto e le reciproche dipendenze (Simeoni et al., 2002).

In particolare, il CPM, nato dalla collaborazione tra la DuPont Corporation e la Remington Rand Corporation, è una tecnica di risoluzione reticolare che permette di individuare quella particolare sequenza di attività (cammino critico) che, all'interno del reticolo generale, rende minimo il tempo necessario al completamento del processo produttivo (Tonchia & Nonino, 2013). Ne consegue che qualsiasi ritardo relativo ad un'attività che appartiene al cammino critico si ripercuote in misura uguale sulla data di completamento del progetto.

Sviluppato nell'ambito del programma missilistico americano, anche il PERT mira ad evidenziare il cammino critico di progetto. Tuttavia, a differenza del CPM, il PERT si basa sull'analisi della possibile durata di ciascuna attività di progetto. Per ciascuna di esse, vengono stimati tre valori (pessimistico, probabile e ottimistico) e la valutazione dell'arco temporale complessivo necessario per il completamento del progetto viene stimato su base statistica (Simeoni et al., 2002).

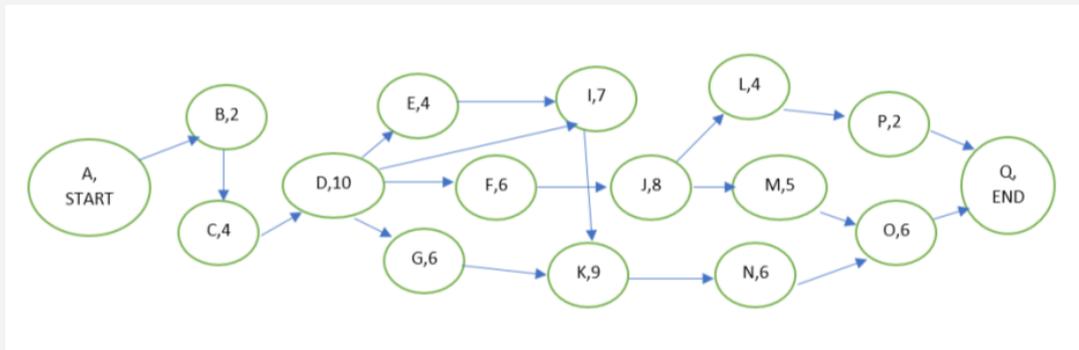


Figura 2 - Esempio di diagramma di PERT. Fonte [Come utilizzare il PERT per determinare la durata di un progetto](#)

Nello stesso periodo furono sviluppati anche il *Work Breakdown Structure* (WBS) e l'approccio dell'*Earned Value*.

Tra gli anni '60 e gli anni '70 si sviluppa la professione del *project manager*. Nel 1967 nasce l'International Project Management Association (IPMA) e nel

1969 viene fondato il Project Management Institute (PMI) per la certificazione e lo sviluppo di standard per il PM. Alla fine degli anni '80, il PMI pubblica la prima edizione del *Project Management Body of Knowledge* (PMBOK), un testo che presenta tecniche e contenuti per il *Project Management*, arrivato ormai alla settima edizione (Tonchia & Nonino, 2013).

Nelle precedenti edizioni, il PMBOK era incentrato prevalentemente sulle metodologie di project management waterfall o “a cascata”. Tuttavia, in un contesto caratterizzato da una continua evoluzione tecnologica e una competizione sempre più forte, il ciclo di vita dei progetti è diventato sempre più breve con requisiti e obiettivi che cambiano nel corso della realizzazione. Le tecniche di project management tradizionali sono poco adatte a progetti che si contraddistinguono per i cambiamenti continui, per i quali risultano più efficaci le metodologie *agile* di project management, che si sono diffuse a partire dagli anni 2000. Per questo motivo, la più recente edizione del PMBOK passa da un approccio focalizzato sui processi ad un approccio basato sui principi, supportando tutte le metodologie di project delivery (Iqbal, 2019).

Dai principi *lean* all'approccio agile nel project management

Siamo nella seconda metà del '900, quando in Giappone, Paese ancora profondamente colpito dalla sconfitta della Seconda Guerra Mondiale e dalla crisi del petrolio che si abbatté sull'economia mondiale, il principale costruttore di automobili giapponese, la *Toyota Motor Company*, sperimenta rimedi per far fronte al radicale calo delle vendite (Liguori, 2016).

Fino a questo momento il mercato era stato caratterizzato da una domanda assolutamente prevedibile, sempre in aumento e guidata dall'offerta, per cui il metodo di produzione industriale principale in uso era la *produzione di massa* ideata dall'americano Henry Ford, consistente nella realizzazione di prodotti con caratteristiche tecnologiche e commerciali, tali da essere facilmente riproducibili e venduti in serie su larga scala.

Ora però in Giappone la domanda si abbassa drasticamente e conseguentemente cambia l'ottica di produzione, andando a focalizzarsi sull'impossibilità di sostenere gli alti costi di investimento iniziale e di giacenza della *Mass Production*, nonché sulla scarsa flessibilità di questo sistema produttivo.

Due ingegneri *Toyota*, Eiji Toyoda e Taiichi Ohno, realizzano dunque che il modello americano centrato sull'industrialità non è applicabile al mercato nipponico, per il quale può invece funzionare un sistema basato sull'efficienza, dove la produzione sia di piccoli volumi e dove la merce sia disponibile "Just In Time" e precisamente nelle quantità richieste. Così sotto la loro guida viene sviluppato il Toyota Production System (TPS), un metodo avanzato di progettazione, controllo e gestione della produzione, che ad oggi viene considerato a tutti gli effetti il precursore della *lean production*, una nuova tecnica di produzione che fa cambiare completamente il punto di vista aziendale, spostando l'attenzione dalla produttività alle esigenze del cliente (Redazione make consulting, n.d.).

Grazie all'introduzione dei nuovi principi produttivi, nell'arco di circa 25 anni, dagli anni '60 alla metà degli anni '80, l'azienda Toyota riesce a diventare

un leader globale nel settore automotive: nella seconda metà degli anni '60 l'azienda cresce da 20.000 a 165.000 unità l'anno e qualche anno dopo mette in produzione il suo più grande successo, l'autovettura Corolla, che è stata prodotta in dieci serie (Toyota, n.d.).

L'esempio pratico del TPS viene teorizzato nel 1990 da tre studiosi del MIT International Motor Vehicle Program, James P. Womack, Daniel T. Jones e Daniel Roos, con la pubblicazione del libro "The machine that changed the world". Il libro descrive l'intero sistema manageriale della produzione snella, lo mette a confronto con il suo opposto, il sistema di massa, e di fatto rende noto al panorama internazionale il rivoluzionario metodo della Lean Production, descrivendone le sue caratteristiche e i suoi principi.



Figura 3 - I principi lean. Fonte: elaborazione propria

1^ Principio: Individuazione ed eliminazione degli sprechi

Questo rappresenta probabilmente il principio fondante della filosofia di produzione *lean*, che viene appunto definita "snella, essenziale", in quanto mirata al continuo abbattimento degli sprechi, insiti nelle attività tipiche della logica a "lotti e code" della *mass production*. Mutuando direttamente il termine dal TPS, i *muda* (sprechi, in giapponese) sono tutte quelle attività che non aggiungono

valore al prodotto finale, dal punto di vista del cliente, e per le quali dunque quest'ultimo non è disposto a pagare (Roos et al., 1990, 47-56).

Possono essere di sette tipi e una volta individuati, devono essere eliminati o ridotti il più possibile al minimo:

1. Attesa: l'attesa è lo spreco più facilmente individuabile, in quanto si manifesta ogni volta che un operatore o un impianto è costretto a sospendere le sue attività perché in attesa di materiale (da parte del fornitore o del magazzino) o di strumenti per la produzione;
2. Sovraproduzione: produrre più unità di quelle richieste;
3. Rilavorazione o sovra lavorazione: reiterare un processo o parte di esso per eliminare errori a monte oppure compiere più lavorazioni di quelle necessarie o di quelle richieste dal cliente;
4. Giacenza: i costi di giacenza sono costi che non vengono ammortizzati nella vendita del prodotto, per cui in ottica lean, "stoccaggio" è sinonimo di "spreco";
5. Trasporto: ogni volta che un prodotto viene trasportato, sono impiegate delle risorse e del tempo aziendale per lo spostamento. Inoltre, si rischia di danneggiare o smarrire il prodotto. Il trasporto non comporta trasformazione né aumento di valore nel prodotto per il cliente, motivo per cui deve essere eliminato o ridotto al minimo costo possibile;
6. Movimento: spostamento improduttivo compiuto dal lavoratore in attesa, dovuto a layout mal disegnati o a posti di lavoro non studiati ergonomicamente;
7. Intelletto: non utilizzare il contributo personale degli operatori, i quali potrebbero avere competenze, particolari abilità o idee migliorative, rappresenta uno spreco;

2^ Principio: Ricerca continua della perfezione

L'individuazione e la successiva eliminazione degli sprechi non è un processo statico, ma continuo, graduale e ciclico; esattamente come il miglioramento delle caratteristiche dei prodotti/servizi nell'ottica del cliente finale.

Questo processo viene definito *kaizen* (“miglioramento continuo”, in giapponese) e nell’ambito industriale/produttivo coinvolge l’intera struttura aziendale: dirigenti ed operai in ugual misura. Più in generale, il management giapponese vede il miglioramento suddiviso in due fasi: *kaizen* e *kakushin/kairyō*. Il *kakushin/kairyō* è l’innovazione, il grande cambiamento; viene deciso dalla direzione e necessita di un grande investimento. Il *kaizen*, come detto sopra, invece, è composto da tanti piccoli miglioramenti, sforzi di piccola entità, attuati ogni giorno da tutta l’azienda (Lean World Class, n.d.).

3^ Principio: Assenza di difetti

La presenza di eventuali difetti su un semilavorato implica la rilavorazione del pezzo, costo non imputabile al cliente, ed un conseguente blocco della linea, la quale dovrebbe invece procedere “a flusso”. Per questi due motivi, i difetti di lavorazione vanno eliminati a monte. Per poter ottenere questo risultato la sequenza di operazioni da svolgere lungo la linea deve essere semplice e a prova di distrazione. *Poka-yoke* (“a prova di errore”, in giapponese) sta ad indicare una scelta progettuale della linea, che ponendo dei limiti alle modalità con cui un’operazione può essere compiuta, obbliga l’operatore a fare la cosa giusta (Liguori, 2016).

4^ Principio: Sistema produttivo *pull*

I sistemi produttivi tradizionali, i sistemi *push*, si basano sulla possibilità di fare una previsione della domanda di mercato. Ogni prodotto viene messo in produzione in un determinato istante, scelto dall’azienda, ed ha una data di consegna associata. Lungo la linea viene sottoposto ad una sequenza di operazioni: una volta conclusasi un’operazione, il prodotto viene spinto alla successiva. Se però la previsione della domanda è errata, al termine il prodotto rimane invenduto e viene stoccato in magazzino (approccio *make to stock*).

Questa è la principale differenza con il nuovo approccio, quello *pull*, nel quale si produce solo se il cliente lo richiede: i prodotti finiti sono tirati a partire dall’operazione finale, in risposta agli ordini dei clienti. L’intero sistema si attiva in funzione delle richieste del cliente, il quale tramite il suo ordine guida la produzione (approccio *make to order*): si innesca una reazione a catena, nella quale ogni stazione richiama il materiale della stazione precedente, così da

azzerare gli stoccaggi di materie prime, semilavorati e prodotti finiti e i relativi costi. L'obiettivo di questa organizzazione è dunque quello di lavorare a scorte zero e di avere il materiale nelle quantità opportune al momento necessario; le due metodologie applicate per controllare il flusso dei materiali sono il sistema *Kanban* e il *Just In Time* (Redazione make consulting, n.d.).

5^ Principio: Creazione di team multifunzionali

Nell'ottica *lean* il personale rappresenta una risorsa e dunque non utilizzarlo pienamente diventa uno spreco. La produzione snella prevede dunque che ogni team di lavoro sia multifunzionale, ovvero che debba occuparsi di più mansioni, tra loro eterogenee. Questo approccio permette così di favorire all'interno del team lo sviluppo di abilità e competenze che diventano trasversali, di tutti, ed anche di eliminare il fenomeno dei funzionari "insostituibili", ovvero operatori che hanno acquisito un ampio bagaglio di esperienze e dalle quali l'azienda dipende in modo esclusivo per determinate attività (Liguori, 2016).

6^ Principio: Sistema di responsabilità decentralizzato

Esattamente come non può essere insostituibile un operaio, così non lo può essere nemmeno il team leader: all'interno dei team multifunzionali non esiste una gerarchia prestabilita, per cui la posizione di leader è spesso solo temporanea e cambia a rotazione tra i membri del team. Ogni individuo con le proprie capacità contribuisce così alla formazione di tutto il gruppo (Liguori, 2016).

Negli anni sono stati registrati alcuni casi di successo dati dall'applicazione delle tecniche del TPS in grandi aziende occidentali (non sono del settore automotive) e quello che è partito dall'essere "il sistema di produzione Toyota" si è diffuso in contesti di produzione e di servizi.

Arriviamo al 2004 quando gli studiosi sopra citati Womack e Jones scrivono un altro "testo sacro" dal titolo "*Lean Thinking*". Il *pensiero snello* viene definito dagli autori un "*potente antidoto contro gli sprechi*" e rappresenta la generalizzazione e la concettualizzazione del sistema Toyota, applicabile a qualsiasi settore.

Anche il *Lean Thinking* è definito da dei principi: 5 principi che, riprendendo in parte quelli della *Lean Production*, li arricchiscono e fanno del pensiero snello una vera e propria filosofia (CONSIDI, n.d.).

1^ Principio: Value

Il valore non deve essere stabilito dall'azienda, ma deve essere riconsiderato mettendo il focus sulla definizione esatta di ciò di cui necessita il cliente.

Per il cliente finale solo una parte delle operazioni e del tempo totale impiegati per produrre, aggiunge effettivo valore. Risulta quindi fondamentale definire chiaramente il valore di uno specifico prodotto o servizio dalla prospettiva del cliente, tutto il resto rappresenterà uno spreco.

2^ principio: Map

A questo punto è necessario specificare il flusso del valore, ovvero l'insieme delle attività necessarie per ottenere il prodotto finito a partire dalle materie prime. Si mappa il flusso e si individuano gli sprechi, anche grazie alla classificazione delle attività in:

- *Value Adding activities*, **VA**: attività il cui costo può essere trasferito al cliente
- *Necessary but Not Value Adding activities*, **NNVA**: attività che non creano valore, ma che non sono eliminabili negli attuali sistemi di produzione
- *Not Value Adding activities*, **NVA**: attività che possono essere eliminate dal sistema di produzione

3^ Principio: Flow

Una volta definito il valore e mappate le attività necessarie, bisogna riorganizzare i processi, in modo che il flusso di produzione scorra senza interruzioni attraverso le varie fasi di aggiunta di valore.

Questo concetto è profondamente diverso dal modo di pensare per "lotti" e comporta interventi radicali sulle attività produttive.

La modifica consiste nello sviluppare layout funzionali e corretti e nel monitorare strettamente l'avanzamento della produzione: ciò permette di ridurre

il *lead time* di prodotto e le energie sprecate, rappresentando così una grande opportunità di miglioramento nell'efficienza qualitativa e quantitativa di una azienda.

4^ Principio: Pull

Esattamente come era nei principi della *lean production*, non è solo importante fornire al cliente ciò per cui è disposto a pagare, ma è anche importante farlo solo quando è richiesto. Bisogna far tirare la produzione e la conseguente fornitura (esterna o magazzino) dalle richieste del mercato. L'intera organizzazione deve essere progettata per poter essere tirata dal cliente: generare attività *Value Adding* in tempi non concordi con le richieste del cliente, fa diventare il valore stesso un *muda*.

5^ Principio: Perfection

Anche nell'ultimo principio viene ripreso il valore del *kaizen*, già elaborato nella *lean production*. Se l'innovazione (*kakushin*) viene messa in atto al momento della riorganizzazione dei processi e della definizione del flusso, è il miglioramento continuo, applicato quotidianamente dall'intera azienda, a dare pieno significato a tutti i passaggi.

Ad oggi il *lean thinking* è applicato con successo a livello internazionale, da varie aziende di tutti i settori e non si limita più ai soli processi manifatturieri, ma si è arrivati a parlare anche di *lean management*, *lean office*, *lean administration* ed anche di *lean nel lavoro individuale* (Liguori, 2016).

Agile è un approccio al project management che deriva dall'applicazione del *lean thinking*. La gestione agile dei progetti significa applicare i principi di *lean* nel settore informatico. Si tratta di una metodologia di PM basata sui seguenti principi:

- controlli frequenti e adattabilità;
- una leadership che incoraggia il lavoro di gruppo, l'autonomia organizzativa e la responsabilità;
- un set di *best practice* che permettono di completare rapidamente progetti di alta qualità;

- un approccio che fa combaciare sviluppo, necessità dei clienti e obiettivi aziendali (Fair, 2012).

La metodologia *Agile* nasce ufficialmente nel 2001, quando un gruppo formato da 17 sviluppatori software si riunisce per discutere di metodologie di project management alternative e con, l'obiettivo di definire un approccio snello, flessibile e orientato al team per lo sviluppo di software, elabora il *Manifesto Agile* (AltexSoft, 2019).

Il *Manifesto Agile* aveva due obiettivi principali:

- ridurre il time-to-market del software, ovvero il tempo necessario perché un software o una funzionalità fosse pronto per l'utilizzo operativo;
- accogliere i feedback degli utenti e del business e adattare il software di conseguenza (RareSeed, 2020).

Il Manifesto Agile

Il *Manifesto Agile* si basa su 4 valori e 12 principi. I valori mirano a preparare il team ad un nuovo modo di lavorare, i principi indicano gli strumenti che si dovrebbe avere a disposizione nel team.

Valori e principi dell'Agile Manifesto



Figura 4 - Valori e principi dell'Agile Manifesto. Fonte: elaborazione propria su (Beck et al., 2001)

Nel corso degli anni, la metodologia agile, sviluppata all'interno del contesto digitale, conosce una forte espansione nel mondo della gestione

progettuale in senso più ampio. Studi, ricerche, corsi universitari e business school adottano la metodologia agile all'interno dei propri ambiti di analisi e programmi (si veda, ad esempio, l'incidenza dei trend di ricerca sulla piattaforma Google riportata nel grafico in basso).

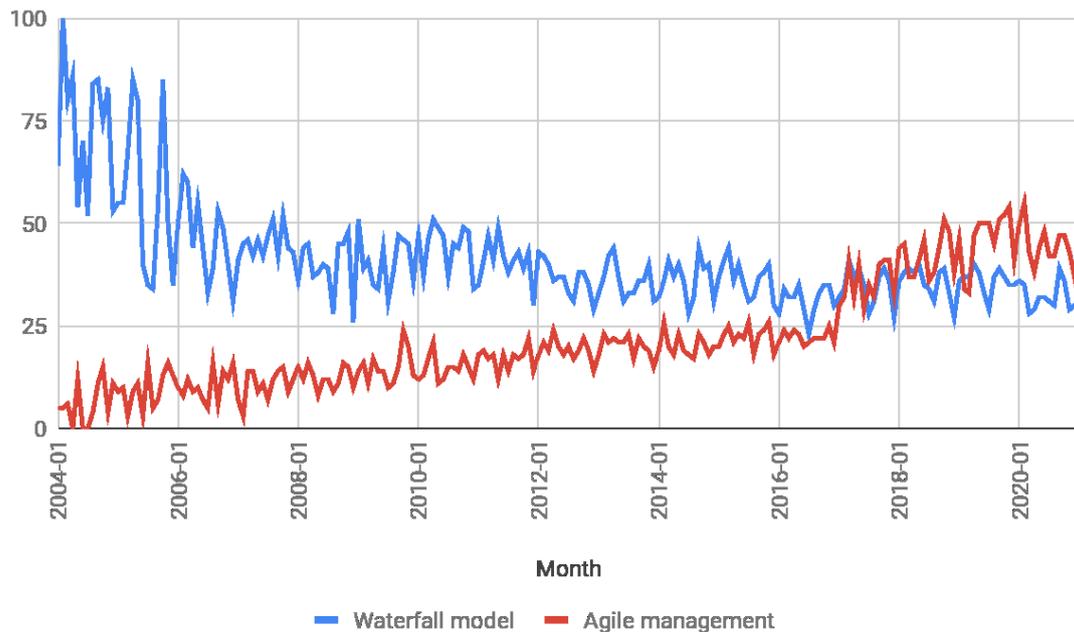


Figura 5 - Popolarità di agile management e waterfall model nel tempo. Fonte: elaborazione su indicatori Google Trends. I valori indicano l'interesse relativo tra i due termini indagati.

Parallelamente, l'approccio agile dimostra una praticità al di fuori dell'ambito del *software development* e viene rapidamente adottato da grandi aziende e gruppi commerciali e di servizi, come Bosch, multinazionale di elettronica e ingegneria, USAA, società di servizi finanziari, Saab, la società svedese di ingegneria per la difesa (The Economist, 2018). Generalmente, la transizione è avvenuta in seguito ad una fase di sovrapposizione tra i sistemi tradizionali e l'approccio agile - focalizzato sui business maggiormente innovativi - per poi applicare la nuova visione di project management su larga scala (Rigby et al., 2018). Nonostante si tratti di un cambiamento sistemico, tutte le principali realtà gestionali si sono interfacciate - o sono in punto di farlo - con l'approccio agile.

Waterfall e Agile a confronto

Le principali differenze tra l'approccio *Waterfall* e *Agile* sono riassunte nella tabella di seguito.

Waterfall	Agile
Piani di progetto dettagliati e a lungo termine con un'unica tempistica	Pianificazione più breve basata su iterazioni e consegne multiple
Gestione del progetto e ruoli del team rigidi e definitivi	Composizione del team flessibile e interfunzionale
Le modifiche ai risultati finali sono costose e dunque scoraggiate	I cambiamenti nei risultati finali sono previsti e di minore impatto
Prodotto interamente completato consegnato alla fine della sequenza temporale delle fasi	Prodotto consegnato in fasi funzionali
Approccio basato sulla definizione dell'ambito e dei requisiti	Approccio collaborativo e interattivo per la definizione dei requisiti
Il cliente è in genere coinvolto solo all'inizio e alla fine di un progetto	Il cliente è coinvolto durante l'iterazione
L'approccio a fasi lineari crea dipendenze temporali: il ritardo di una fase comporta il ritardo di tutte le successive	L'approccio concorrente cerca di ridurre le dipendenze temporali

Tabella 1 - Key differences between a traditional waterfall project and an agile project. Fonte: elaborazione Fair, 2012

Sequenzialità vs iterazioni

Il *Waterfall Model* (anche noto come *Linear Sequential Life Cycle Model*) è un approccio **sequenziale** e lineare per la gestione dei progetti. Esso corrisponde ad una gestione del progetto di tipo tradizionale e sequenziale, basata su una successione a cascata di fasi distinte per lo sviluppo del progetto stesso (Scannapieco, 2020).

Esistono 3 principi fondamentali per questo approccio:

1. Il processo di sviluppo viene suddiviso in fasi sequenziali: l'inizio di una fase segue solo al completamento della precedente e, a differenza dei

modelli iterativi, ogni fase viene eseguita solo una volta. In tale metodologia, il prodotto viene consegnato al cliente esclusivamente alla fine del processo.

2. Ogni fase produce un output che viene usato come input della fase successiva: i risultati di ogni fase rappresentano le precondizioni per la fase successiva.
3. Ogni fase del progetto viene documentata: l'approccio *waterfall* considera fondamentale la documentazione generata nelle varie fasi perché, in questo modo, l'eventuale perdita di un componente del team non causerebbe anche una conseguente perdita di conoscenza, in quanto quest'ultima sarebbe tutta contenuta nella documentazione. Sarà sufficiente che la risorsa alla quale verrà assegnato il nuovo incarico, consulti i vari documenti per familiarizzare con il progetto (RareSeed, 2020).

Nella pratica vengono utilizzate diverse versioni del modello a cascata ed attualmente sono in uso modelli che suddividono i processi di sviluppo in sei fasi:

Analisi dei requisiti Nella prima fase di un progetto viene svolto lo studio di fattibilità, durante il quale il progetto viene valutato in base a costi, profitti e realizzabilità. A questo studio fa seguito una dettagliata definizione dei requisiti: vengono raccolte le informazioni complete sui requisiti che il progetto richiede. L'analisi può essere svolta in vari modi, dalle interviste alle domande al brainstorming interattivo. Entro la fine di questa fase, i requisiti del progetto devono essere fissati e devono essere raccolti in un documento che verrà distribuito al team.

Progettazione Sulla base di requisiti, incarichi e strategie già individuati, in questa fase il team determina come il progetto deve essere svolto e quindi suddiviso nei vari moduli/attività; il risultato della fase di progettazione sarà un documento di bozza con il progetto di costruzione del sistema e la pianificazione dei test per i singoli componenti.

Implementazione In questa fase il progetto viene concretamente svolto: viene realizzata la codifica dell'applicazione. Il risultato della fase di implementazione è un prodotto software che nella fase successiva verrà testato per la prima volta come prodotto completo (test alpha).

Test Durante questa fase, vengono effettuati test per rilevare e segnalare eventuali problemi dell'output che devono essere risolti, chiamati test di sistema e test di integrazione. I tester trovano e segnalano metodicamente qualsiasi problema. Se venissero individuati problemi seri, il progetto potrebbe dover tornare alla fase iniziale per la rivalutazione.

Esecuzione In questa fase l'output è pronto per essere rilasciato al cliente.

Manutenzione Il prodotto è stato consegnato al cliente e viene utilizzato. In caso di problemi, potrebbe essere necessario che il team crei patch e aggiornamenti che possano risolverli. Ancora una volta, problemi rilevanti possono richiedere un ritorno alla fase iniziale. (Twproject Staff, 2020) (IONOS, 2019)

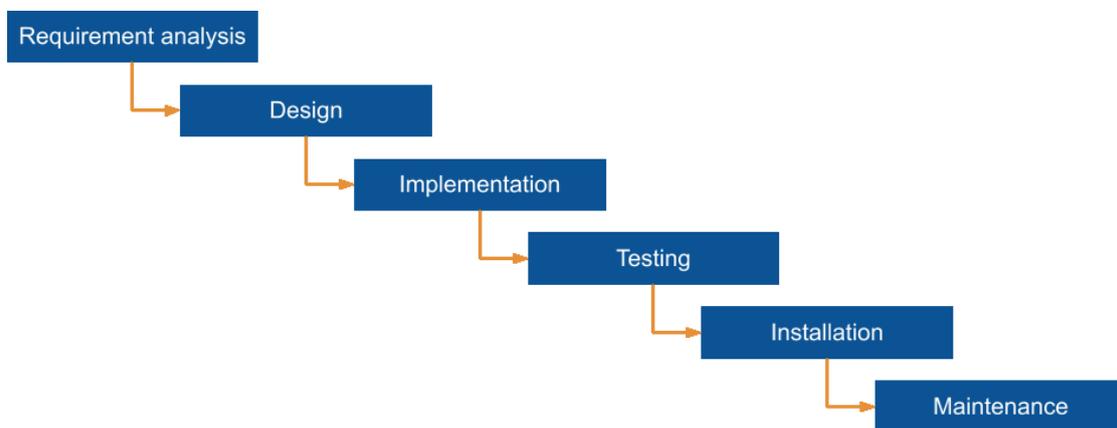


Figura 6 -Fasi del metodo Waterfall. Fonte: Elaborazione propria su Metodologia Agile e PMBOK: waterfall e metodo iterativo

La gestione *agile* del progetto si basa su metodologie iterative per pianificare e guidare i progetti: gli step del metodo *waterfall* vengono ripresi, ma rivisti in chiave iterativa. Il progetto viene suddiviso in piccole parti, ciascuna delle quali deve essere svolta in un dato periodo di tempo, che prende il nome di iterazione. Le iterazioni avvengono secondo uno schema prevedibile e ciclico con date di inizio e fine definite (di solito due settimane). Durante ogni iterazione viene svolto il ciclo di vita dello sviluppo del progetto; il risultato del ciclo di vita viene presentato dal team di progetto agli stakeholder, con lo scopo di ricevere un feedback sull'uso e l'applicazione delle funzionalità. A questo punto, un feedback positivo o un'eventuale critica del risultato, rappresenta l'input per

definire i passaggi successivi nel percorso di realizzazione del prodotto finale. La convalida del risultato dell'iterazione è assolutamente fondamentale nella gestione *agile*, al fine di confermare o sfatare le ipotesi iniziali del progetto sviluppato. (Gathercole, 2018)

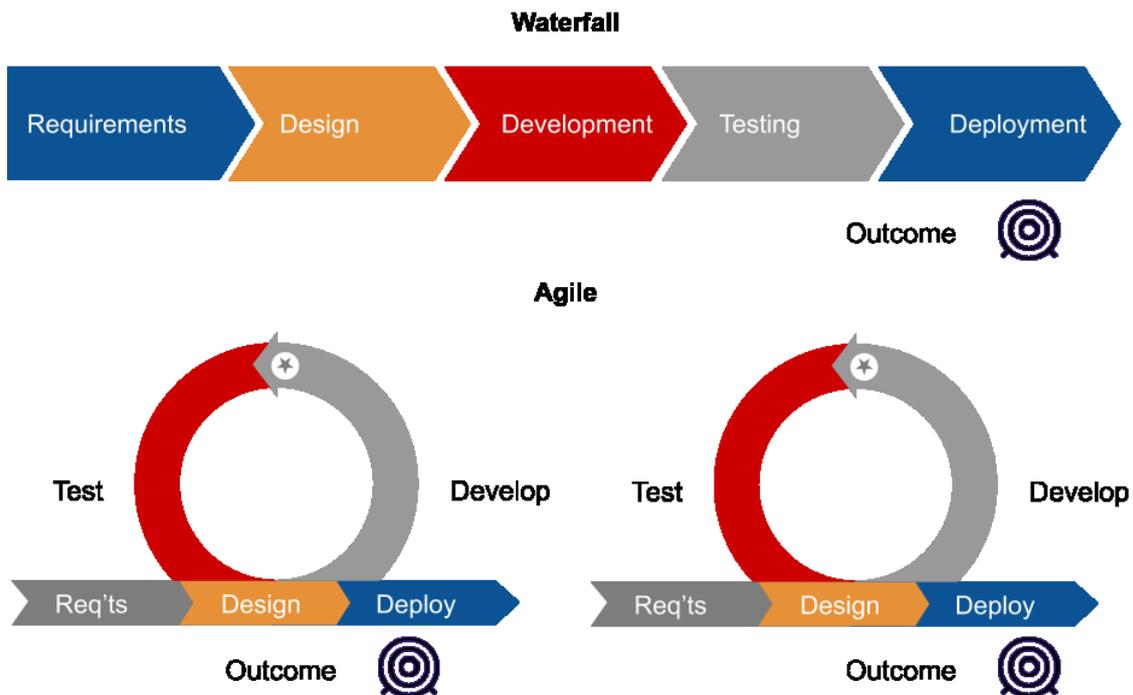


Figura 7 - Agile Project Management VS Waterfall Project Management. Fonte elaborazione propria su (Gathercole, 2018)

Pianificazione vs adattamento

Il metodo *waterfall* viene anche definito predittivo, poiché i requisiti sono ben definiti, concordati e formalizzati e molti potenziali difetti sono “intercettati” nelle fasi preliminari di *analisi* e *pianificazione*. È vantaggioso in quanto è uno dei modelli più facili da gestire: per sua natura, ogni fase ha risultati specifici (*deliverable*) e un processo di revisione. La pianificazione e la progettazione risultano passaggi “semplici”, in quanto gli sviluppatori e i clienti concordano ciò che verrà consegnato già all'inizio del ciclo di vita del processo e - ad eccezione di revisioni, approvazioni o riunioni sullo stato - la presenza del cliente non è strettamente necessaria dopo la fase di analisi dei requisiti; tutto ciò rende possibile una consegna più rapida del progetto. L'interfaccia con i clienti coinvolti, inoltre, risulta chiara, grazie alla documentazione dettagliata di processo e risultati ed alla impostazione definita di tempistiche, attività e relativi costi.

Infine, il metodo è particolarmente adattabile per lo spostamento di squadre: ogni team è a conoscenza dell'output da ottenere e delle tempistiche da rispettare ed a seconda della fase del progetto da svolgere, è possibile che vari membri del team siano coinvolti o continuino con altri lavori. (Scannapieco, 2020)

Si può trarre beneficio dalle caratteristiche del *waterfall* se quest'ultimo viene utilizzato per progetti ordinati e prevedibili, nei quali tutti i requisiti sono chiaramente definiti dal principio e possono essere stimati con precisione.

Nella maggior parte dei settori però, i progetti di questo tipo stanno diminuendo: le richieste dei clienti sono cambiate ed hanno portato a maggiore pressione sulle imprese, le quali devono adattarsi e modificare i propri metodi di gestione dei progetti.

Ed è così che la rigidità del *waterfall*, nel quale le modifiche non sono previste nel corso del progetto, ma sono solo successive all'*implementazione*, e il mancato coinvolgimento dei clienti nelle fasi successive all'*analisi* iniziano a rappresentare dei veri e propri svantaggi: ne è un esempio lo sviluppo di grandi progetti di software, i quali possono durare anche diversi anni, senza che vengano adeguati regolarmente ai più recenti risultati di sviluppo tecnologico e che quindi risultano obsoleti già al momento della loro introduzione sul mercato. (IONOS, 2019)

Ad avere maggior successo negli attuali scenari di mercato, caratterizzati da altissima competizione e richieste di maggiore produttività, velocità e qualità sono i metodi *agili*.

“Agile is a project methodology and approach that is derived using Lean thinking. Agile projects apply “Lean” concepts in the information technology environment. It is the proven project management methodology that encourages the following key concepts:

- *Frequent inspection and adaptation*
- *A leadership philosophy that encourages teamwork, self-organization, and accountability*

- *A set of engineering best practices that allow for rapid delivery of high-quality projects*
- *A business approach that aligns development with customer needs and company goals” (Fair, 2012)*

La gestione *waterfall* dipende da un approccio pianificato e resistente al cambiamento: vengono fornite una documentazione pesante, un modello rigido, dei requisiti che sono stati così accuratamente rivisti e dei piani così rigorosamente organizzati, che cambiare anche l'aspetto più piccolo di un progetto può causare un effetto a cascata di imprevedibilità.

L'*agile* è invece una gestione guidata dal cambiamento, dove la collaborazione continua e la comunicazione aperta tra progettisti e stakeholder sono fortemente incoraggiate: insieme, infatti, arrivano a definire la direzione che prenderà il prodotto, acquisendo questa direzione durante l'intero ciclo di vita del progetto. (Gathercole, 2018)

I requisiti del prodotto non vengono definiti tutti in un'unica soluzione come era per il *waterfall*, ma ad ogni iterazione si lavora su pezzi di requisiti. Ciò rende possibile eseguire simultaneamente le attività di sviluppo e di effettuare più test durante l'intera vita del progetto. (Scannapieco, 2020)

Nella pratica, quindi, si vanno a rilasciare continue e rapide modifiche al progetto in piccole porzioni, con l'obiettivo di migliorare la soddisfazione dei clienti. Inoltre, per perseguire il principio del *kaizen* ripreso dal *lean thinking*, l'*agile* utilizza un approccio flessibile anche per il lavoro in team: i vari team, costituiti da pochi sviluppatori ciascuno, si organizzano in autonomia ed all'inizio di ogni iterazione pianificano e definiscono con il cliente, un elenco di deliverable da consegnare entro la durata della relativa iterazione. Se non è possibile completare tutto il lavoro pianificato, il lavoro viene ridistribuito e le informazioni vengono utilizzate per la pianificazione dell'iterazione successiva. In questo modo anche la stesura della documentazione di progetto risulta più leggera e diventa possibile integrare le modifiche in qualsiasi fase del ciclo di vita; come dicevamo, un approccio guidato dal cambiamento. (Red Hat, n.d.) (Scannapieco, 2020) (Gathercole, 2018)

L'agile più utilizzato: il metodo Scrum

Dalla sua definizione, nel 2001, *agile* è stato usato come termine “ombrello”, ovvero è stato utilizzato come espressione per indicare in modo generico una serie di metodi leggeri e flessibili; in realtà *agile* rappresenta prima di tutto un *mindset*, una filosofia, un paradigma organizzativo descritto da 4 linee guida che ne costituiscono la filosofia e definito da 12 principi. La filosofia si è poi concretizzata in metodi agili, i quali condividono alcuni principi e valori fondamentali, descritti nell'*Agile Manifesto*. I più diffusi tra i metodi agili sono l'*Extreme Programming* e lo *Scrum*.

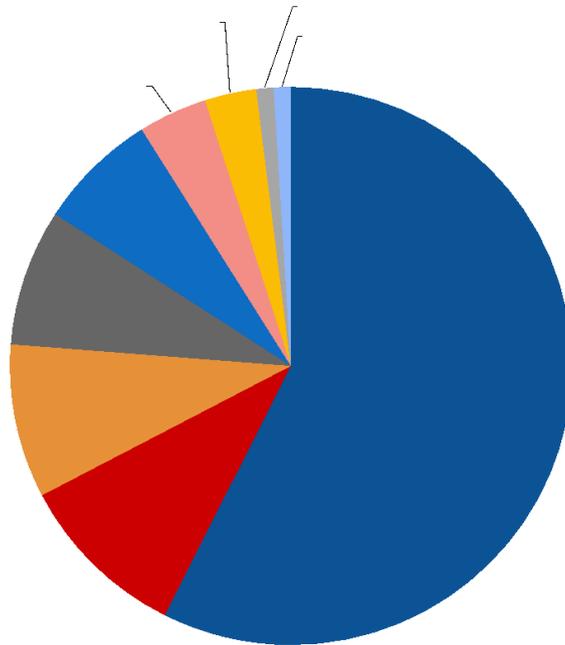


Figura 8 - Metodologie Agile utilizzate. Fonte: elaborazione su Digital.ai, 2020

Il termine *scrum* viene utilizzato per la prima volta in un articolo di Nonaka e Takeuchi, *The New Product Development Game*, pubblicato in Harvard Business Review nel 1986, in cui viene descritta una nuova modalità di lavoro per i team di sviluppo del prodotto: questa modalità ha una forte somiglianza con la “mischia del rugby”, *scrum* appunto, durante la quale non vi è nessun capo, nessun leader che prende le decisioni, non vi sono schemi ed è il team che in modo autonomo e auto-organizzato prende le decisioni sul momento. Il team di sviluppo deve lavorare dunque come una squadra che “cerca di raggiungere

l'obiettivo come unità, passando la palla avanti e indietro". La sua diffusione come metodologia strutturata si deve, invece, a Ken Schwaber e Jeff Sutherland, che hanno sviluppato e mantengono la "Scrum Guide", la guida alla metodologia scrum per lo sviluppo del software. Scrum consiste in una serie di pratiche che rendono il lavoro dei team altamente produttivo e funzionante nella consegna di valore per il cliente finale. (Lisca, 2019)

"Scrum è un framework all'interno del quale le persone possono affrontare complessi problemi di adattamento, fornendo al contempo in modo produttivo e creativo prodotti del valore più alto possibile." (Scrum.org, n.d.)

Scrum è il metodo più indicato per progetti complessi ed innovativi. Si tratta di un framework di processo utilizzato dai primi anni Novanta per gestire il lavoro su prodotti complessi. Scrum non è un processo, una tecnica o un metodo definitivo. Piuttosto, esso è un framework all'interno del quale è possibile utilizzare vari processi e tecniche. Scrum rende chiara l'efficacia relativa della propria gestione di prodotto e delle proprie tecniche di lavoro, così da poter migliorare continuamente il prodotto, il team e l'ambiente di lavoro. (Schwaber & Sutherland, 2017, 4)

Il framework è costituito da un particolare insieme di pratiche che rendono il lavoro dei team altamente produttivo e funzionante nella consegna di valore per il cliente finale. Il processo di gestione di un progetto è suddiviso in sprint, per poter coordinare il processo di sviluppo del prodotto con le esigenze del committente/cliente; il processo è iterativo e la durata degli sprint va da 1 a 4 settimane (hubstrat, n.d.).

La teoria alla base di questo metodo è quella del controllo empirico dei processi (empirismo), secondo la quale, da un lato, la conoscenza deriva dall'esperienza e, dall'altro lato, le decisioni si basano su ciò che si conosce. Per questo motivo si prevede un processo iterativo con un approccio incrementale

che ottimizza, passo dopo passo, la prevedibilità ed il controllo del rischio. (hubstrat, n.d.)

Secondo quanto riportato dalla *Scrum Guide*, inoltre, sono tre i pilastri che sostengono il controllo empirico di processo: la trasparenza, l'ispezione e l'adattamento.

- *Trasparenza*: significa che tutti coloro che partecipano ad un progetto sanno qual è lo scopo (trasparenza verticale) e che cosa fanno gli altri (trasparenza orizzontale). Dunque, gli aspetti significativi del processo devono essere visibili ai responsabili del risultato e perché ciò sia possibile, devono essere definiti degli standard comuni in modo che gli osservatori condividano una comune comprensione di ciò che viene visto. Un esempio di standard può essere un linguaggio comune di riferimento al processo, condiviso da tutti i partecipanti.
- *Ispezione*: chi utilizza Scrum deve ispezionare frequentemente gli artefatti (modi di documentare il lavoro) di Scrum e l'avanzamento verso l'obiettivo di ogni sprint, con lo scopo di rilevare deviazioni indesiderate. Le ispezioni non devono essere tanto frequenti da intralciare il lavoro stesso e sono più utili quando eseguite da chi ha l'abilità e la competenza necessarie ad effettuarle in un particolare stadio del lavoro.
- *Adattamento*: è la conseguenza dell'ispezione, in quanto, se chi ispeziona verifica che uno o più aspetti del processo sono al di fuori dei limiti accettabili, in base a questo risultato il team di sviluppo ripianifica il progetto per apportare maggior valore al cliente finale. L'adattamento deve essere portato a termine il più rapidamente possibile per ridurre al minimo un'ulteriore deviazione. Sempre nella guida Scrum vengono previste quattro occasioni formali all'interno dello Sprint per l'*ispezione* e l'*adattamento*: lo *Sprint Planning*, il *Daily Scrum*, la *Sprint Review* e la *Sprint Retrospective*.

Ogni parte del framework serve a uno specifico scopo ed è essenziale per il successo e l'utilizzo di Scrum. Scrum è costituito dagli Scrum Team e dai ruoli,

eventi, artefatti e regole a essi associati e le regole di Scrum governano le relazioni e le interazioni tra essi. (Schwaber & Sutherland, 2017, 4)

In pratica Scrum definisce un set di attività che consentono al team di offrire più valore ai clienti in meno tempo. Le attività, infatti, offrono ai clienti la possibilità di esaminare, guidare e influire sul lavoro del team durante lo sviluppo. L'approccio non tenta di definire tutto all'inizio di un progetto, come nel *waterfall* in cui la definizione degli obiettivi e la pianificazione vengono completati come prima attività; al contrario, il team lavora in brevi iterazioni, gli sprint, perfezionando il piano man mano che il lavoro procede.

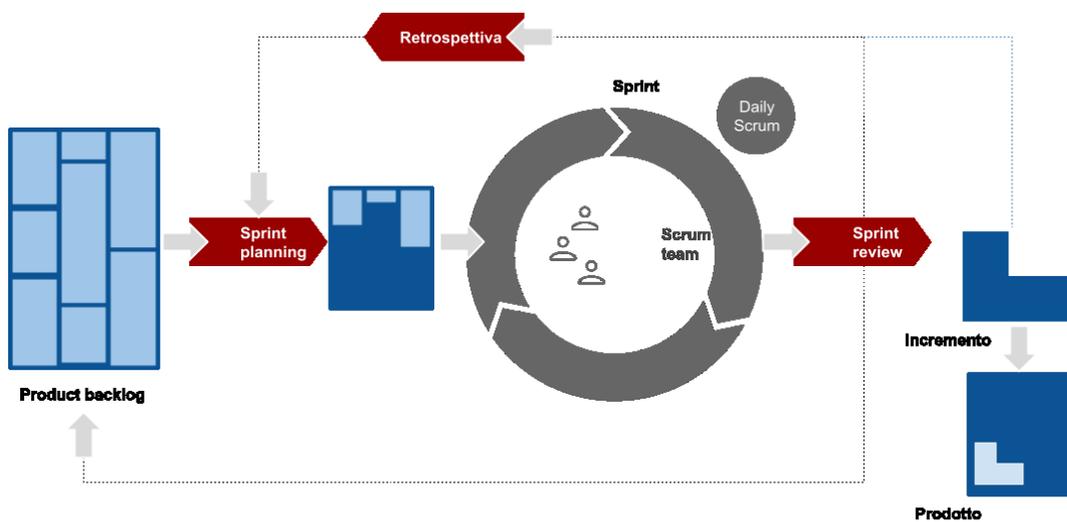


Figura 9 - Rappresentazione schematica della metodologia Scrum. Fonte: elaborazione su RareSeed, 2020

Le componenti principali di Scrum si dividono in: ruoli, artefatti ed eventi.

Lo Scrum Team

Le figure (ruoli) principali in un processo Scrum rappresentano lo Scrum Team e sono coloro che realizzano il prodotto. Il Team è costituito da un *Product Owner*, dal Team di Sviluppo e da uno Scrum Master. Secondo quanto riporta la Scrum Guide, gli Scrum Team devono essere auto-organizzati e cross-funzionali, ovvero, scelgono come meglio compiere il proprio lavoro invece di essere diretti

da altri al di fuori del team ed hanno tutte le competenze necessarie per realizzare il lavoro senza dover dipendere da nessuno al di fuori del team.

Gli Scrum Team rilasciano i prodotti in modo iterativo e incrementale, massimizzando le opportunità di feedback. I rilasci incrementali di prodotto garantiscono che vi sia sempre disponibile una versione potenzialmente utile del prodotto. Il modello di team in Scrum è progettato per ottimizzare la flessibilità, la creatività e la produttività. (Schwaber & Sutherland, 2017, 8).

Il Product Owner

Il *Product Owner* è colui che fa le veci degli *stakeholder*: li rappresenta e, conoscendo tutti i requisiti del prodotto, ne porta avanti gli interessi. Interpreta l'interfaccia tra il business, i clienti e i requisiti del prodotto da un lato e il team dall'altro ed ha quindi il compito di massimizzare il valore del prodotto e del lavoro svolto dal Team di Sviluppo.

Riprendendo la descrizione della Guida *Scrum*, il *Product Owner* è l'unica persona responsabile della gestione del *Product Backlog*. Tale gestione include:

- Esprimere chiaramente gli “*Item*” del *Product Backlog*;
- Ordinare gli “*Item*” del *Product Backlog* per meglio raggiungere gli obiettivi e le missioni;
- Ottimizzare il valore del lavoro del Team di Sviluppo;
- Assicurare che il *Product Backlog* sia visibile, trasparente e chiaro a tutti e mostri su cosa lo *Scrum Team* lavorerà in seguito;
- Assicurare che il Team di Sviluppo comprenda gli “*Item*” del *Product Backlog* al livello necessario.

Il *Product Owner* è un'unica persona, non un comitato. Può esprimere la volontà di un comitato nel *Product Backlog*, ma chiunque voglia cambiare l'ordine di un elemento deve rivolgersi al *Product Owner*; è lui il responsabile finale (*accountable*) e perché possa agire con successo, all'interno dell'organizzazione

tutti devono rispettare le sue decisioni. Le decisioni del *Product Owner* sono visibili nel contenuto e nell'ordine delle priorità del *Product Backlog*.

Il Team di Sviluppo

Il Team di Sviluppo è costituito da professionisti che lavorano per consegnare un incremento di prodotto potenzialmente rilasciabile alla fine di ogni Sprint. I Team di Sviluppo sono strutturati e autorizzati dall'organizzazione per gestire il proprio lavoro, in questo modo si crea una sinergia che ottimizza l'efficienza e l'efficacia complessive del Team.

I Team di Sviluppo hanno le seguenti caratteristiche:

- Sono auto-organizzati, per cui nessuno (neanche lo *Scrum Master*) dice al Team di Sviluppo come trasformare il *Product Backlog* in incrementi di funzionalità potenzialmente rilasciabili;
- Sono cross-funzionali, con tutte le competenze come team necessarie per creare un Incremento di prodotto;
- Non sono gerarchicamente organizzati. *Scrum* non riconosce alcun titolo ai membri del Team di Sviluppo, indipendentemente dal lavoro eseguito dalla persona;
- Non contengono sotto-team dedicati a particolari domini come il testing o la *Business Analysis*;
- I singoli membri possono avere competenze specialistiche e aree di specializzazione, ma è il Team di Sviluppo nel suo complesso ad avere la responsabilità finale.

Anche la dimensione del Team di Sviluppo è indicata nella guida a Scrum: il numero di professionisti non deve essere né troppo basso (si potrebbero incontrare limiti dovuti alla mancanza di skill durante lo Sprint, che impedirebbero così la consegna di un incremento potenzialmente rilasciabile), né troppo alto

(richiederebbe un eccessivo lavoro di coordinamento) e solitamente si mantiene da 3 a 9 componenti.

Lo Scrum Master

Lo Scrum Master è il responsabile del processo, colui che deve garantire che la metodologia Scrum venga compresa ed eseguita con successo. Deve accertarsi che il team lavori in maniera coerente con lo sviluppo del progetto, eliminare eventuali ostacoli che limitano la capacità del team di raggiungere l'obiettivo dello *Sprint* e i deliverable previsti ed organizzare e facilitare i meeting di confronto. (hubstrat, n.d.)

Sebbene sia un ruolo manageriale, lo Scrum Master non è il team leader, ma piuttosto colui che facilita una corretta esecuzione del processo. Lo Scrum Master aiuta coloro al di fuori dello *Scrum Team* a capire se le loro interazioni con esso sono utili per massimizzare il valore creato dal team, oppure no.

Gli artefatti

“Gli artefatti di Scrum rappresentano il lavoro e il valore al fine di fornire trasparenza e opportunità di ispezione e adattamento.” (Schwaber & Sutherland, 2017, 20)

Gli artefatti sono documenti che rendono trasparenti le informazioni su prodotto, modelli, parti di prodotto funzionante (RareSeed, 2020).

Compito dello Scrum Master è quello di lavorare con il Team di Scrum e con l'organizzazione per aumentare la trasparenza degli artefatti. Questo lavoro di solito comporta una fase di apprendimento, una di persuasione ed una di cambiamento (Lavecchia, n.d.).

Nella metodologia sono previsti tre artefatti:

- il *Product Backlog*;

- lo *Sprint Backlog*;
- l'incremento.

Il Product Backlog

“Il *Product Backlog* è il documento in cui sono specificate tutte le attività necessarie per la realizzazione del progetto ed è l'unica fonte di requisiti per le modifiche da apportare al prodotto” (Schwaber & Sutherland, 2017).

Si può dire, in maniera semplicistica, che si tratta di un elenco di attività ordinato per priorità e scadenze (Cistaro, 2020).

Come anticipato, il *Product Owner* è responsabile del suo contenuto, dell'assegnazione del livello di priorità alle attività in esso elencate e della sua disponibilità per il Team.

Un *Product Backlog* può nascere come una lunga lista oppure con pochi elementi: la sua prima stesura definisce solo i requisiti inizialmente conosciuti e meglio compresi. Non è mai completo, in quanto evolve come il prodotto al quale si riferisce: è dinamico e cambia continuamente per identificare ciò che serve al prodotto per essere appropriato, competitivo e utile. Nel *Product Backlog* vengono elencati le caratteristiche, le funzioni, i requisiti, le migliorie e le correzioni che costituiscono le modifiche da apportare al prodotto nelle future versioni. I suoi elementi hanno i seguenti attributi: descrizione, ordine, stima e valore. I requisiti non smettono mai di cambiare e perciò il *Product Backlog* è un “artefatto vivente” (Lavecchia, n.d.) (Schwaber & Sutherland, 2017, 13).

Le attività che compongono la lista vengono chiamate “*Item*” e a differenza di una tipica lista di requisiti del prodotto, il *Product Backlog* viene prioritizzato, cioè viene deciso con quale priorità svolgere i compiti; quest'ultima è una caratteristica di *Scrum* che lo differenzia dal classico metodo *Waterfall* di gestione dei progetti. Il team si concentra sulle attività più importanti e definisce le attività da svolgere per prime con una stima di tempo (Biumi, 2019).

Il raffinamento del *Product Backlog* (*Product Backlog refinement*) è l'atto di aggiungere dettagli, stime e ordine agli elementi del *Product Backlog*. Questo è un processo continuo in cui il *Product Owner* e il Team di Sviluppo collaborano

sui dettagli degli elementi del *Product Backlog*. Gli elementi ordinati più in alto sono più chiari e meglio dettagliati rispetto a quelli più in basso e le stime sono più precise e si basano su maggiore chiarezza e maggior numero di dettagli. Gli elementi del *Product Backlog* che impegneranno il Team di Sviluppo durante lo *Sprint* sono rifiniti a tal punto che possano essere “Fatti” nel periodo massimo di uno *Sprint* (Schwaber & Sutherland, 2017, 13-14).

Le stime contenute nel *Product Backlog* sono stime approssimative sia del valore di business che dello sforzo necessario a svilupparlo; le stime aiutano il *Product Owner* a calcolare la *timeline* e possono influenzare l'ordine degli “*Item*” (Lavecchia, n.d.).

Lo Sprint Backlog

Lo *Sprint Backlog* è il documento che definisce tutti i task (attività, operazioni, incarichi assegnati) da completare nei singoli Sprint. È una previsione fatta dal Team di Sviluppo in relazione alle priorità indicate nel *Product Backlog* e al lavoro necessario per raggiungere gli obiettivi del singolo *Sprint* (Cistaro, 2020).

Durante lo *Sprint Planning* viene deciso quali sono gli “*Item*” da inserire nello *Sprint Backlog*, che saranno oggetto di lavoro in un determinato Sprint; qui sono i membri del Team stesso a decidere quali “*Item*” includere e sui quali lavorare, il *Product Owner* non ha potere decisionale in questo, e il lavoro viene distribuito all'interno del Team su base volontaria (Biumi, 2019).

Lo *Sprint Backlog* consiste dunque in un piano con dettagli sufficienti affinché i cambiamenti in atto possano essere compresi e aggiornati quotidianamente nel *Daily Scrum* e rende visibile il lavoro che il Team di Sviluppo identifica come necessario per raggiungere lo *Sprint Goal*. Durante tutto lo *Sprint*, il piano può essere modificato da ogni membro del Team perché appeso e visibile su un tabellone per tutti i membri del Team stesso: per la comunicazione dello *Sprint Backlog* viene infatti solitamente utilizzata la metodologia “Kanban Board” (Biumi, 2019)

Le modifiche potrebbero essere necessarie se il Team di Sviluppo, operando attraverso il piano, dovesse venire a conoscenza di maggiori dettagli sul lavoro necessario a raggiungere lo Sprint Goal. Il Team avrà dunque la responsabilità di:

- aggiungere allo *Sprint Backlog* l'eventuale nuovo lavoro risultato necessario;
- aggiornare la stima relativa al lavoro rimanente, quando del lavoro viene eseguito o completato;
- rimuovere alcuni elementi del piano, qualora non dovessero più essere ritenuti utili (Schwaber & Sutherland, 2017, 15).

L'incremento

L'Incremento è la somma di tutti gli elementi del *Product Backlog* completati durante uno *Sprint* e durante tutti gli *Sprint* precedenti (Schwaber & Sutherland, 2017, 14).

Quando si tratta dell'elenco di tutti i *task* completati durante lo *Sprint*, questo può essere esattamente uguale a quanto espresso nello *Sprint Backlog* - ciò accade solitamente nei Team altamente performanti- o inferiore.

Alla fine di uno Sprint, il nuovo Incremento deve risultare "Fatto", il che significa che deve essere utilizzabile e deve incontrare la definizione di "Fatto" data dallo *Scrum Team*: rispondendo alla domanda "Quando un lavoro viene considerato completato?" i membri del Team decidono di comune accordo cosa vogliono considerare come un Incremento "Fatto" (Biumi, 2019) (Schwaber & Sutherland, 2017, 14)

L'Incremento deve essere utilizzabile indipendentemente dal fatto che il *Product Owner* decida di rilasciarlo realmente o meno: nell'Incremento, infatti, sono presenti gli elementi realizzati e completati esclusivamente in base a quanto concordato dal Team di Sviluppo per garantire la realizzazione del progetto. Nella metodologia *Agile*, la realizzazione del prodotto è lo scopo primario, quindi tutte le attività che non sono focalizzate su questo obiettivo vengono ridotte al minimo (Cistaro, 2020).

Gli eventi

Nel framework è prevista la suddivisione del lavoro in eventi: come indicato dalla *Scrum Guide*, infatti, gli eventi sono finalizzati a creare regolarità e ridurre al minimo la necessità di incontri non definiti da Scrum stesso. Ogni Sprint contiene al suo interno tutti gli eventi, i quali, oltre allo Sprint stesso, rappresentano un'occasione formale per ispezionare e adattare qualcosa: sono infatti specificamente progettati per garantire i principi di trasparenza ed ispezione ed una mancata inclusione di uno qualsiasi degli eventi comporta dunque una riduzione della trasparenza o un'occasione mancata per praticare l'ispezione e l'adattamento (Schwaber & Sutherland, 2017, 11).

Gli eventi hanno la caratteristica di essere “time-boxed”, ovvero di avere una durata temporale massima fissa:

- lo Sprint ha una durata prefissata, la quale non può essere né accorciata né allungata;
- gli altri eventi possono invece terminare quando viene raggiunto il loro scopo, assicurando che sia stato speso il tempo necessario e sufficiente per evitare sprechi all'interno del processo.

Lo Sprint

Lo Sprint rappresenta il fulcro di Scrum: uno Sprint è un breve periodo (da una a quattro settimane) in cui il team procede di iterazione in iterazione per creare il prodotto, fare modifiche e apportare miglioramenti. Nel momento in cui finisce uno Sprint ne inizia subito un altro e ogni Sprint ha come obiettivo quello di creare un incremento di prodotto potenzialmente rilasciabile, utilizzabile e “Fatto/Done”.

“Fatto” è quando un elemento del Product Backlog o un incremento è considerato un lavoro completo, rilasciabile in produzione. Ha principalmente tre caratteristiche:

- Tutti devono capire che cosa si intende per “Fatto”, che cosa comporta e che cosa significa, secondo il principio della trasparenza;
- La definizione di “Fatto” deve essere condivisa e chiara per tutti all’interno dello Scrum Team;
- La definizione di “Fatto” garantisce la trasparenza, evita le ambiguità, ed è usata per valutare quando il lavoro dell’incremento di prodotto è completo (Biumi, 2019).

Durante uno *Sprint* vi sono alcune regole da seguire, come l’obbligo di non apportare modifiche che mettano a rischio lo Sprint Goal o la necessità di assicurarsi che gli obiettivi relativi alla qualità non si degradino (Schwaber & Sutherland, 2017, 12).

Ogni *Sprint* può essere considerato un progetto, con un obiettivo di ciò che si va a costruire e un piano flessibile che guida la costruzione, il lavoro svolto e l’Incremento risultante con un orizzonte non più lungo di un mese. Infatti, se uno Sprint avesse un orizzonte temporale troppo lungo, la definizione di ciò che viene costruito potrebbe cambiare, la complessità potrebbe crescere e di conseguenza il rischio potrebbe aumentare. Invece gli Sprint rendono possibile la prevedibilità assicurando, almeno una volta al mese, l’ispezione e l’adattamento del progresso verso uno *Sprint Goal* ed eventualmente limitano il rischio ad un mese di costo (Schwaber & Sutherland, 2017, 12).

Uno Sprint può essere cancellato prima della scadenza del tempo massimo stabilito se lo Sprint Goal diventa obsoleto: può verificarsi, ad esempio, se l’organizzazione cambia direzione o se le condizioni di mercato o della tecnologia cambiano. Solo il *Product Owner* ha l’autorità di annullare lo *Sprint*, anche se può essere influenzato dagli *stakeholder*, dal Team di Sviluppo o dallo *Scrum Master*. Ad ogni modo, data la breve durata degli *Sprint* ed il fatto che un’eventuale cancellazione di uno *Sprint* consumerebbe risorse, poiché tutti

dovrebbero partecipare ad un nuovo *Sprint Planning* per poterne cominciare un altro, in realtà le cancellazioni avvengono molto raramente (Schwaber & Sutherland, 2017, 12).

Come già anticipato, gli Sprint sono costituiti da quattro eventi differenti: lo *Sprint Planning*, il *Daily Scrum*, lo *Sprint Review* e la *Sprint Retrospective*.

Lo Sprint Planning

All'inizio di ogni *Sprint*, viene svolto un meeting durante il quale lo Scrum Team si riunisce per decidere in modo collaborativo il lavoro da eseguire nello Sprint. Per uno Sprint di quattro settimane, lo Sprint Planning è limitato temporalmente ad un massimo di otto ore, mentre per Sprint più brevi, l'evento è di solito più breve. È compito dello *Scrum Master* assicurarsi che il meeting abbia luogo e che tutti ne comprendano la finalità (Schwaber & Sutherland, 2017, 13).

Gli input per questo incontro sono il *Product Backlog*, l'ultimo Incremento del prodotto (somma di tutti gli elementi del *Product Backlog* completati durante uno Sprint), la capacità prevista del Team di Sviluppo durante lo Sprint e le prestazioni registrate in passato del Team di Sviluppo.

Durante lo *Sprint Planning* il Team deve affrontare due temi principali:

1. “Cosa può essere fatto in questo *Sprint*?”
2. “Come si effettuerà il lavoro scelto?”

Per poter rispondere alla prima questione, viene richiesta l'intera collaborazione da parte dello Scrum Team: il Team di Sviluppo lavora per prevedere le funzionalità che saranno sviluppate durante lo Sprint e sarà anche l'unico componente del Team in grado di valutare cosa si potrà compiere durante lo *Sprint* successivo; il *Product Owner* discute l'obiettivo al quale lo Sprint dovrebbe aspirare e gli elementi del *Product Backlog* che, se completati durante lo *Sprint*, permetterebbero di raggiungere lo *Sprint Goal*.

Lo *Sprint Goal* è un obiettivo che sarà raggiunto all'interno dello *Sprint* attraverso l'implementazione del *Product Backlog* e fornisce indicazioni al Team di Sviluppo sulle motivazioni per le quali il Team di Sviluppo stesso sta costruendo l'Incremento (Schwaber & Sutherland, 2017, 13).

Una volta definito lo *Sprint Goal* (*The What?*), cioè quale risultato può essere raggiunto nello *Sprint*, e dopo aver di fatto suddiviso gli elementi del *Product Backlog* da considerare per raggiungerlo, si arriva a rispondere al secondo quesito: il Team di Sviluppo deve decidere come costruire, durante lo *Sprint*, le funzionalità in un Incremento "Fatto" del prodotto. Gli elementi del *Product Backlog* selezionati per lo *Sprint* più il piano per la consegna definiscono lo *Sprint Backlog* (*The How?*) (Biumi, 2019).

Tenendo bene a mente lo *Sprint Goal*, il Team di Sviluppo inizia con la progettazione del sistema e la definizione del lavoro necessario per convertire il *Product Backlog* in un Incremento del prodotto. Questo lavoro può essere di varia dimensione o impegno stimato e può capitare che durante lo *Sprint* il lavoro richiesto sia diverso da quanto il Team di Sviluppo aveva pianificato. In questo caso il Team di Sviluppo collaborerà con il *Product Owner* per determinare come rivedere al meglio il piano senza perdere di vista lo *Sprint Goal*. Se il Team di Sviluppo determina che c'è troppo o troppo poco lavoro, può rinegoziare gli elementi dello *Sprint Backlog* con il *Product Owner*. Il *Product Owner*, infatti, può aiutare a chiarire gli elementi selezionati dal *Product Backlog* e, di conseguenza, a fare dei compromessi.

Per concludere il meeting avendo uno *Sprint Goal* e uno *Sprint Backlog* compresi e assimilati da tutti, il Team fa domande, chiede chiarimenti e approfondimenti per avere informazioni il più possibile dettagliate sul lavoro che andrà a svolgere. Prima della fine dello *Sprint Planning*, il Team di Sviluppo dovrebbe essere in grado di spiegare al *Product Owner* e allo *Scrum Master* come intende intraprendere il lavoro contenuto nello *Sprint Backlog*, in quanto

team autogestito, al fine di raggiungere lo Sprint Goal e creare l'Incremento previsto (Biumi, 2019) (Schwaber & Sutherland, 2017, 13-14).

Il Daily Scrum

“Il Daily Scrum migliora la comunicazione, elimina altri incontri, identifica gli ostacoli allo sviluppo allo scopo di rimuoverli, evidenzia e promuove il rapido processo decisionale e migliora il livello di conoscenza del Team di Sviluppo. Esso rappresenta un incontro chiave d'ispezione e adattamento.” (Schwaber & Sutherland, 2017, 15)

Il Daily Scrum è un evento della durata massima di 15 minuti, durante il quale il Team di Sviluppo pianifica il lavoro per le successive 24 ore. Per ridurre la complessità, si svolge ogni giorno allo stesso orario e nello stesso luogo e prendono parte all'incontro sempre le stesse persone; è infatti un meeting interno al Team di Sviluppo e qualora siano presenti altre persone, lo Scrum Master si assicura che queste ultime non disturbino il meeting.

In generale lo *Scrum Master* garantisce che il Team di Sviluppo tenga la riunione e che la riunione si mantenga entro il tempo limite; sarà poi il Team di Sviluppo responsabile della conduzione del *Daily Scrum*.

Il *Daily Scrum* ottimizza la probabilità che il Team di Sviluppo raggiunga lo *Sprint Goal*: ogni giorno permette al Team di Sviluppo di ragionare su come ha intenzione di lavorare insieme, come team auto-organizzato, per raggiungere lo Sprint Goal e realizzare l'Incremento atteso entro la fine dello Sprint. La struttura del meeting è definita dal Team di Sviluppo e può essere condotta in vari modi purché questi ultimi siano finalizzati all'avanzamento verso lo Sprint Goal; alcuni Team usano domande, altri si basano maggiormente su discussioni (Schwaber & Sutherland, 2017, 15).

Ogni membro del team deve concentrarsi su tre punti principali:

- Che cosa ho fatto ieri?
- Che cosa faccio oggi?
- C'è qualche impedimento che mi ostacola nell'andare avanti?

Queste tre domande fanno sì che ogni membro del Team comprenda a che punto si trova lo svolgimento del lavoro e cosa manca ancora da completare. Inoltre, dichiarare cosa verrà fatto durante la giornata aumenta il senso di responsabilità del Team (Biumi, 2019).

Infine è importante comprendere che il *Daily Scrum* non deve concentrarsi solamente sulla risoluzione dei problemi, ma serve a comunicare gli aggiornamenti giornalieri a tutte le figure che ricoprono i tre ruoli dello *Scrum Team*, per fare in modo che siano sempre connessi tra loro.

La Sprint Review

La Sprint Review è un incontro che si tiene alla fine dello Sprint per ispezionare l'Incremento e adattare, se necessario, il *Product Backlog*. Durante la *Sprint Review* lo *Scrum Team*, il *Product Owner*, lo *Scrum Master* e gli *stakeholder* discutono su ciò che è stato realizzato durante lo Sprint. Si tratta di un incontro informale e la presentazione dell'Incremento ha lo scopo di raccogliere feedback, anche sulla base dello *Sprint Goal* deciso nello *Sprint Planning*; sulla base delle informazioni acquisite, il *Product Backlog* viene adattato e si condivide che cosa verrà fatto negli Sprint successivi.

Durante questo meeting è più importante presentare l'obiettivo generale che si voleva raggiungere, piuttosto che entrare nel dettaglio di ogni oggetto degli Sprint (Biumi, 2019).

La *Scrum Guide* stabilisce che la durata dell'incontro debba essere di quattro ore per uno Sprint che dura un mese (la durata sarà minore per Sprint più brevi) e che la review debba includere i seguenti elementi:

- Il *Product Owner* deve spiegare quali “*Item*” del *Product Backlog* sono stati “Fatti” e quali non sono stati “Fatti”;
- Il Team di Sviluppo deve discutere su cosa sia andato bene durante lo Sprint, quali problemi siano stati riscontrati ed eventualmente la loro risoluzione;
- Il Team di Sviluppo deve mostrare il lavoro che ha “Fatto” e rispondere alle domande relative all’Incremento;
- Il *Product Owner* deve discutere lo stato attuale del Product Backlog e fare una previsione delle possibili date di consegna in base alla misura dell’avanzamento attuale;
- L’intero gruppo deve stabilire cosa fare dopo; in questo modo la Sprint Review fornisce un prezioso contributo ai successivi Sprint Planning.
- Passare in rassegna come il marketplace o il potenziale utilizzo del prodotto possa avere cambiato l'elemento di maggior valore da implementare successivamente;
- Passare in rassegna la timeline, il budget, le funzionalità potenziali e il marketplace per il successivo rilascio previsto del prodotto.

La Sprint Retrospective

La Sprint Retrospective è l’evento che si tiene dopo la Sprint Review e prima del successivo Sprint Planning e consiste in una riunione della durata massima di tre ore per uno Sprint di un mese. Rappresenta un’occasione formale per lo Scrum Team di ispezionarsi e creare un piano di miglioramenti da attuare durante lo Sprint successivo.

Per la valutazione, vengono ispezionati diversi aspetti su come è andato lo Sprint: persone, relazioni, processi, strumenti, regole, definizione di fatto (Biumi, 2019).

Come descritto nella Guida Scrum, la Sprint Retrospective ha diversi obiettivi, ovvero:

- Ispezionare come l'ultimo Sprint è andato in merito a persone, relazioni, processi e strumenti;
- Identificare e ordinare gli elementi principali che sono andati bene e le migliorie potenziali;
- Creare un piano per attuare i miglioramenti al modo di lavorare dello Scrum Team.

Entro la fine della Sprint Retrospective, lo Scrum Team dovrebbe aver individuato i miglioramenti che implementerà nel seguente Sprint; attuare tali miglioramenti durante lo Sprint successivo è l'adattamento all'ispezione dello Scrum Team stesso. Anche se i miglioramenti possono essere implementati in ogni momento, la Sprint Retrospective fornisce un'opportunità formale per focalizzare l'ispezione e l'adattamento (Schwaber & Sutherland, 2017, 19).

Scrum e waterfall: quale scegliere

L'utilizzo delle metodologie agili per il project management continua ad aumentare nel tempo, sia in ambito IT sia in altri settori e dipartimenti. Tuttavia, è necessario tenere presente che la metodologia di project management scelta deve adattarsi alle caratteristiche del progetto e dell'organizzazione che lo implementa.

Una prima considerazione sull'opportunità di scegliere un approccio prevalentemente waterfall o prevalentemente agile può essere fatta analizzando come tali metodologie permettono di gestire il triangolo del triplice vincolo, anche noto come *iron triangle*.

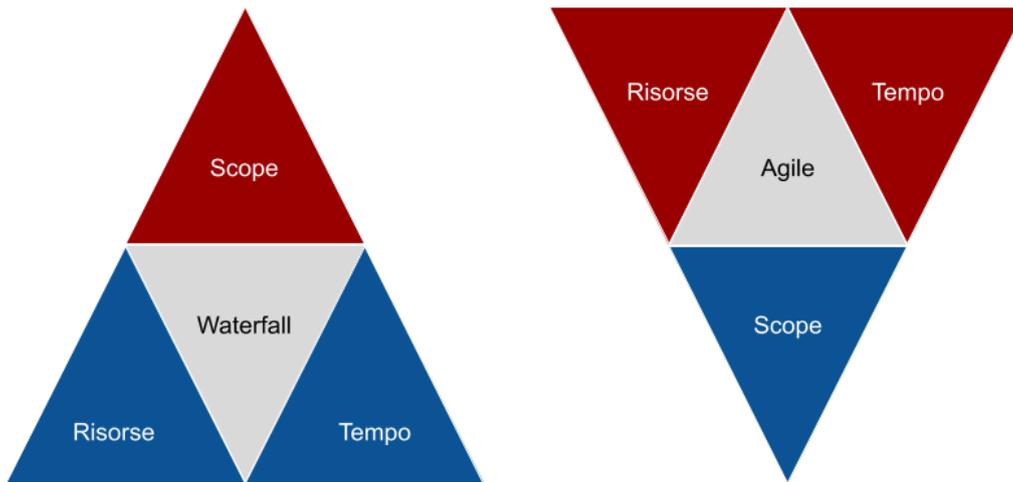
Il triangolo, introdotto da Barnes nel 1969, rappresenta una metafora dei tre vincoli fondamentali a cui è sottoposto un progetto - tempi, costi e *scope* - e della stretta correlazione tra essi. Non è infatti possibile cambiare uno dei vincoli senza ripercussioni sugli altri.

Il triangolo dei vincoli originale segue un approccio waterfall. Lo *scope* del progetto è fisso, tempi e costi sono variabili (Aljaber, n.d.). Nella pratica significa

che gli obiettivi e i requisiti del progetto sono chiari e fissi dall'inizio, mentre i tempi e le risorse possono cambiare e dipendono dallo *scope*.

Il rapporto tra i vincoli risulta invertito nel caso di Agile. In questo caso, costi e tempi sono variabili fisse mentre lo *scope* può cambiare nel corso del tempo e a seguito delle varie iterazioni.

Fissi



Variabili

Figura 10 - Il triangolo dei vincoli nell'approccio Waterfall e Agile. Fonte: elaborazione propria

Per quanto siano rappresentazioni semplificate, i diversi *iron triangle* aiutano a individuare i progetti per i quali ciascuna metodologia risulta più adeguata.

Ricorrendo ad un'ulteriore semplificazione, i progetti possono essere classificati considerando due caratteristiche - gli obiettivi e la soluzione - e assumendo per esse due valori: sono chiari o non chiari. Si ottiene così la matrice di seguito (Wysocki, 2019).

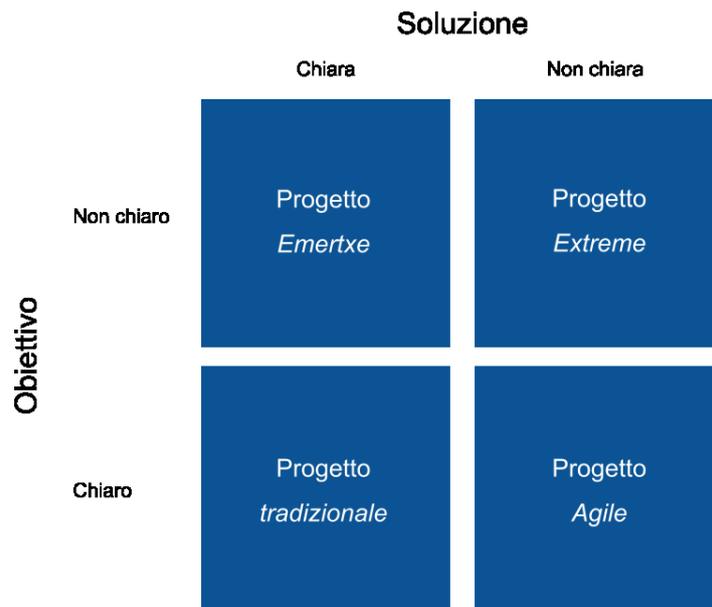


Figura 11 - Categorizzazione dei progetti in base alla conoscenza degli obiettivi perseguiti e delle soluzioni da applicare. Fonte: elaborazione su Wysocki, 2019

I progetti che ricadono nel terzo quadrante - progetti tradizionali - sono i più semplici. Si tratta spesso di progetti già noti all'organizzazione che deve svolgerli e/o che sono già stati realizzati in passato. Gli obiettivi sono chiari e il team di progetto sa come raggiungere i risultati prefissati. Non ci si aspettano particolari imprevisti o cambiamenti. In questo caso, la metodologia waterfall, che, come menzionato sopra, prevede che gli obiettivi e l'ambito di lavoro siano fissi, risulta la più adeguata (Wysocki, 2019).

I vantaggi delle metodologie di project management tradizionale

Quando applicabile, la metodologia *waterfall* ha il vantaggio di:

- una fase di pianificazione e design chiara e rapida, poiché è possibile concordare i *deliverable* del progetto;
- una migliore definizione dell'approccio complessivo;
- un ambito di lavoro definito;
- una stima dei costi più facile;
- ruoli definiti all'interno del team;
- risorse per lavorare parallelamente su attività specifiche (Delos Santos, 2020).

I progetti tradizionali sono, tuttavia, sempre meno frequenti in un contesto socioeconomico caratterizzato dal continuo cambiamento. Le organizzazioni e le società si trovano sempre più spesso a gestire progetti complessi, in cui l'obiettivo non è sempre definito in modo chiaro ma non si sa qual è la soluzione giusta per raggiungerlo. In questi casi la metodologia *waterfall* non è adatta, in quanto non si conoscono a priori le attività che verranno svolte e non è possibile pianificarle nelle prime fasi progettuali.

Per la gestione di progetti per i quali è noto l'obiettivo finale ma non si conoscono nel dettaglio le attività da implementare per raggiungerlo - i progetti rappresentati nel quarto quadrante della matrice - si suggerisce un approccio Agile (Wysocki, 2019). L'utilizzo di rilasci incrementali e il continuo confronto con il committente permettono infatti di concordare nel tempo la soluzione che verrà adottata, mitigando il rischio che questa non risponda alle aspettative iniziali. Tale rischio risulta invece massimo seguendo un approccio *waterfall*, in cui la condivisione avviene solo al termine della fase di sviluppo, durante la quale non è possibile tenere conto di eventuali cambiamenti relativi allo *scope* o agli obiettivi iniziali.

Valore vs rischio

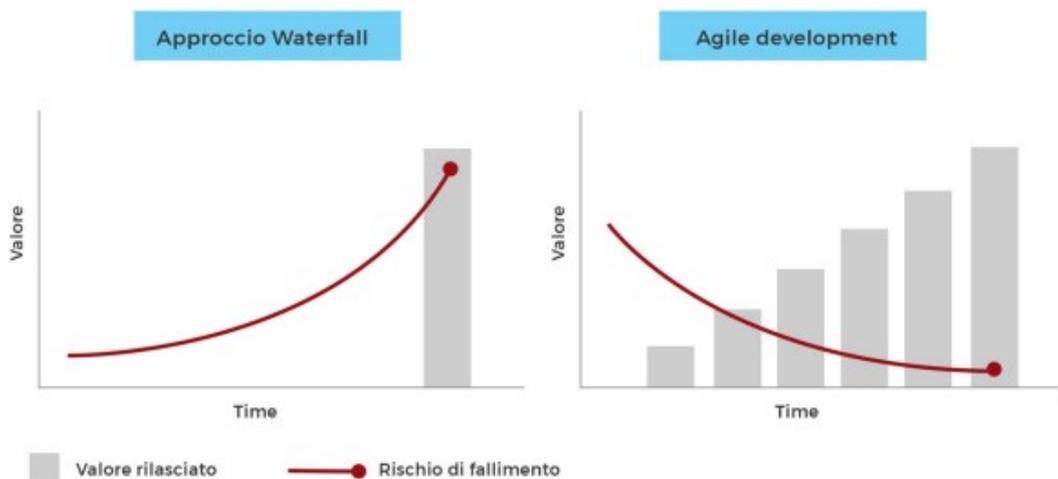


Figura 12 - Gestione di valore e rischio: waterfall vs agile. Fonte: (RareSeed, 2020)

Per fornire alcuni esempi pratici, si suggerisce di adottare:

- **un approccio orientato alla pianificazione** per progetti che presentano rischi significativi, e.g. rischi dovuti alla regolamentazione, rischi di perdite finanziarie o rischio che il progetto venga abortito. Un approccio predittivo permette di rafforzare la pianificazione, il controllo e il monitoraggio delle attività, mitigando tali rischi.
- **un approccio orientato al cambiamento** per progetti innovativi, quali ad esempio lo sviluppo di una nuova soluzione, per i quali non avrebbe senso creare tutta la documentazione prevista dalla metodologia waterfall. Un approccio iterativo e flessibile risulta più adatto in tali contesti caratterizzati da un rapido cambiamento e nei quali è possibile alcuni miglioramenti incrementali che generano valore per gli stakeholder.
- **un approccio orientato al tempo** per progetti rapidi o urgenti, in cui il tempo è la variabile determinante, o per progetti che vengono ripetuti su base regolare (e.g. gestione di eventi mensili, campagne pubblicitarie). In questo caso si considerano solo le attività essenziali di *project management* (Burgan & Burgan, 2014).

Tuttavia, i criteri da scegliere per classificare i progetti ed associare la metodologia di gestione sono molti. Si possono considerare la dimensione (in termini di persone coinvolte, *budget* e durata), la natura del progetto (progetto legato ad un'opportunità strategica, progetto di natura organizzativa, progetto richiesto da un cliente) e del settore in cui si inserisce (un progetto in ambito edilizio ha peculiarità diverse da uno in ambito di sviluppo *software*) (Burgan & Burgan, 2014).

Ad ogni modo, è sempre necessario tenere presente che quando si tratta di progetti e project management, non esiste una soluzione unica per tutti. Le caratteristiche e la natura del progetto così come l'esperienza del project manager e le peculiarità del team di lavoro dovrebbero essere tenute in considerazione quando si sceglie la modalità con cui il progetto sarà affrontato: più la metodologia di project management è tagliata sul progetto, maggiori sono le possibilità di successo dello stesso (Whitaker, 2014).

Conclusioni

Dalla fine del XX secolo è emersa gradualmente la consapevolezza che la metodologia tradizionale di project management - affermatasi nel secolo precedente - non si addice a tutte le tipologie di progetto.

La metodologia waterfall rimane valida per i progetti per i quali è possibile - e necessario - pianificare in anticipo attività e *deliverable*, consentendo di gestire in modo efficace gli elementi variabili del progetto, ovvero il tempo e le risorse.

Tale approccio mostra però i suoi limiti quando applicato in un contesto caratterizzato da cambiamenti continui - sempre più frequente nel mondo attuale.

In queste situazioni, è necessario adottare metodologie di *project management* flessibili che permettono di ridurre il rischio che il prodotto finale non sia adeguato alle aspettative, grazie alla possibilità di gestire, e non subire, le possibili evoluzioni che si verificano nel corso dello sviluppo. Il loro utilizzo è diffuso velocemente, partendo dal mondo del *software development* ed interessando oggi aziende che operano in ogni settore.

Rimane tuttavia valido il principio che non esiste una metodologia di gestione valida per tutti i progetti. Il project manager deve essere in grado di scegliere quella più adeguata considerando, innanzitutto, le caratteristiche del progetto, in termini di rischi, tipologia di attività, chiarezza degli obiettivi e della soluzione da perseguire e rilevanza dei fattori chiave (e.g. tempo).

Bibliografia

Aljaber, T. (n.d.). *The iron triangle of planning*.
<https://www.atlassian.com/agile/agile-at-scale/agile-iron-triangle>

AltexSoft. (2019). *Agile Project Management. Best practices and methodologies*.

Beck, K., Beedle, M., Arie, v. B., & Cockburn, A. (2001). *The Agile Manifesto*.
<https://agilemanifesto.org/iso/it/manifesto.html>

Biumi, I. (2019, giugno 17). *Guida ai 4 eventi formali utilizzati in Scrum*. Agile School. <https://www.agile-school.com/blog/guida-ai-4-eventi-formali-utilizzati-in-scrum>

Burgan, S. C., & Burgan, D. S. (2014). *One size does not fit all: Choosing the right project approach*. <https://www.pmi.org/learning/library/choosing-right-project-approach-9346>

Cassese, S. (2016). La rivoluzione del fordismo. *Il Sole 24 Ore*.
https://st.ilssole24ore.com/art/cultura/2016-08-06/la-rivoluzione-fordismo-171654.shtml?uuid=ADVb69y&refresh_ce=1

CodingJam. (2021). *Dal waterfall ai principi agili*. CodingJam.
<https://codingjam.it/dal-waterfall-ai-principi-agili/>

Delos Santos, J. M. (2020). *Agile vs. Waterfall: Differences in Software Development Methodologies?* <https://project-management.com/agile-vs-waterfall/>

Digital.ai. (2020). *14th Annual State of Agile report*. <https://stateofagile.com/#>

The Economist. (2018, luglio 5). The fashion for agile management is spreading. *The Economist*.

Fair, J. (2012). *Agile versus Waterfall approach is right for my ERP project?*
<https://www.pmi.org/learning/library/agile-versus-waterfall-approach-erp-project-6300>

Fair, J. (2012, Maggio 9). *Agile versus Waterfall: approach is right for my ERP project?* Project Management Institute.

<https://www.pmi.org/learning/library/agile-versus-waterfall-approach-erp-project-6300>

Gathercole, R. (2018, Settembre 13). *How to apply agile project management*. SPF Quality Professionals. <https://www.spf-consulting.ch/insights/how-to-apply-agile-project-management/>

hubstrat. (n.d.). *Metodo Agile E Scrum: I Vantaggi Di Applicarli All'interno Dell'azienda*. hubstrat. <https://hubstrat.it/metodo-agile-scrum-vantaggi-azienda/#:~:text=Scrum%20%C3%A8%20il%20metodo%20Agile,le%20esigenz e%20del%20committente%2Fcliente>.

humanware ProjectManagementcenter. (2021). *Metodologia Agile e PMBOK: waterfall e metodo iterativo*. humanware. <https://www.humanwareonline.com/project-management/center/pmbok-agile/>

IONOS. (2019, marzo 21). *Il modello a cascata*. IONOS Digital Guide. <https://www.ionos.it/digitalguide/siti-web/programmazione-del-sito-web/modello-a-cascata/>

Iqbal, M. (2019). *The Standard of Project Management and PMBOK 7*. <https://mudassiriqbal.net/the-standard-of-project-management-and-pmbok-7/>

Lavecchia, V. (n.d.). *Artefatti utilizzati di Scrum per il successo di un progetto*. Informatica e Ingegneria Online. <https://vitolavecchia.altervista.org/artefatti-utilizzati-di-scrum-per-il-successo-di-un-progetto/>

Lisca, F. (2019, giugno 26). *Agile e Scrum, qual è la differenza: facciamo chiarezza*. Agile school. <https://www.agile-school.com/blog/agile-e-scrum-qual-%C3%A8-la-differenza-facciamo-chiarezza>

PM Consulting. (n.d.). *Breve storia del Project Management*. <http://www.pmconsult.it/cenni-storici.html>

Project Management Institute. (2003). *Guida Al Project Management Body of Knowledge*. Project Management Institute.

RareSeed. (2020). *Storia di Agile da waterfall a scrum*. RareSeed Edizioni.

Red Hat. (n.d.). *What is agile methodology*. Red Hat. <https://www.redhat.com/it/devops/what-is-agile-methodology>

Rigby, D. K., Sutherland, J., & Noble, A. (2018, maggio). Agile at Scale. *Harvard Business Review, May-June*. <https://hbr.org/2018/05/agile-at-scale>

Scannapieco, G. (2020, maggio 12). *Metodi Waterfall e Agile*. Hub dell'innovazione digitale. <https://digitalinnovationhub.org/metodi-waterfall-e-agile-7c97dcfea54c>

Schwaber, K., & Sutherland, J. (2017). *Scrum Guide*. <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Italian.pdf>

Scrum.org. (n.d.). *Un modo migliore per costruire prodotti*. Scrum.org The Home of Scrum. <https://www.scrum.org/resources/what-is-scrum>

Simeoni, E., Serpelloni, P., & Rampazzo, L. (2002). *Project Management. La gestione per progetti*. Regione Veneto.

Tonchia, S., & Nonino, F. (2013). *La Guida del Sole 24 Ore al Project Management*. Gruppo24Ore.

Twproject Staff. (2020, Ottobre 19). *Il metodo Waterfall: cos'è e a cosa serve*. twproject. <https://twproject.com/it/blog/il-metodo-waterfall-cose-e-cosa-serve/>

Watt, A. (2014). *Project mana*. Victoria, B.C.: BCcampus. <https://opentextbc.ca/projectmanagement/>

Whitaker, S. (2014). *PMI White Paper. The Benefits of Tailoring: Making a Project Management Methodology Fit*. <https://www.pmi.org/learning/library/tailoring-benefits-project-management-methodology-11133>

Wysocki, R. K. (2019). *Effective Project Management. Traditional, Agile, Extreme, Hybrid* (6a edizione ed.). John Wiley & Sons, Inc. ISBN: 978-1-119-56278-8

Zanella, F. (2017, Novembre 8). *I cicli di vita del software: dal modello waterfall all'agile*. BP Sistema. <http://bpsistema.it/i-cicli-di-vita-del-software-dal-modello-waterfall-allagile/>

Ringraziamenti

Ritengo infine doveroso concludere questa tesi ringraziando tutte le persone che mi hanno sostenuto in questo percorso universitario.

Per primo ringrazio il mio relatore, il Professor Maurizio Bevilacqua, per avermi trasmesso la sua conoscenza e per avermi seguito nella stesura di questo elaborato.

Un grazie particolare devo dedicarlo alla mia famiglia: mia madre, mio padre, mio fratello maggiore e le mie sorelle, tutti i loro compagni e i miei magnifici nipoti. Loro mi hanno sostenuta nei momenti più difficili e hanno gioito con me per i miei successi.

Grazie alle mie amiche di sempre per aver rappresentato per me dei punti fermi, dei riferimenti ai quali aggrapparmi ogni volta che mi sentivo smarrita.

Infine grazie al mio compagno Francesco, per avermi sopportato nei giorni in cui io stessa non lo avrei fatto e supportato le volte in cui sarei stata la prima a buttarsi via.