

POLITECNICO DI TORINO

Corso di Laurea Magistrale

Ingegneria della Produzione Industriale e dell'Innovazione Tecnologica

Tesi di Laurea Magistrale

La metodologia Agile: l'uso di Scrum nei progetti

Accenture



**Politecnico
di Torino**

Relatrice

Professoressa Elisabetta Raguseo

Candidato

Gabriele Raimo

Ottobre 2022

Sommario

Ringraziamenti	6
1. Introduzione	8
2. La metodologia Agile	9
2.1. Storia.....	9
2.1.1. Pre 2001 – Pratiche e Metodi si sviluppano attraverso l’esperienza	9
2.1.2. Post 2001 – L’adozione diventa mainstream.....	10
2.2. Cos’è la metodologia Agile.....	12
2.3. Confronto tra Agile e metodologie tradizionali (Waterfall)	14
2.4. Introduzione ai framework Agile	18
2.4.1. Approcci Agile a livello di team.....	18
2.4.2 Approcci Agile su scala aziendale	21
3. Il framework Scrum.....	22
3.1. Cenni storici e fondamenti teorici	23
3.1.1. La teoria di Scrum e i 5 valori	25
3.2. Scrum Team	27
3.2.1. Il Product Owner.....	28
3.2.2. Il Team di Sviluppo.....	28
3.2.3. Lo Scrum Master	29
3.3. Le cerimonie di Scrum.....	30
3.4. Gli artefatti di Scrum	33
3.4.1. Estensione di artefatti.....	35
3.4.2. Trasparenza degli artefatti.....	36
4. Gestione di un progetto con Scrum	37
4.1. Lo Sprint	39
4.1.1. Sprint Planning	40

4.1.2. Daily Scrum.....	41
4.1.3. Sprint Review	42
La Sprint Review include i seguenti elementi:.....	43
4.1.4. Sprint Retrospective.....	43
4.1.5. Cancellare uno Sprint.....	44
4.1.6. Tenere traccia dello stato di avanzamento del rilascio	44
4.1.7. Concludere il rilascio	45
4.2. Strumenti di gestione per progetti Agile.....	45
4.2.1. Atlassian Jira.....	46
4.2.2. Pivotal Tracker.....	47
4.2.3. Nutcache.....	48
4.2.4. Scrumwise	49
4.2.5. Miro.....	50
5. Casi studio – Progetti Accenture.....	52
5.1. Chi è Accenture.....	52
5.1.1. Accenture in Italia.....	52
5.2. Caso studio 1 – Come il testing Agile ha portato efficienza in Swisscom	53
5.2.1. La Sfida	53
5.2.2. Cosa ha fatto Accenture	54
5.2.3 Persone e Cultura	55
5.2.4. Valore consegnato.....	55
5.3. Caso studio 2 – Sviluppo di un portale per l’e-commerce di Leonardo	56
5.3.1. Invito al cambiamento.....	56
5.3.2. Quando la tecnologia incontra l'ingegno umano	56
5.3.3 Una differenza preziosa.....	57
5.4. Caso studio 3 – Guidare una compagnia assicurativa Europea nella sua crescita	58
5.4.1. La sfida.....	58
5.4.2. Cosa ha fatto Accenture	58

5.4.3. Persone e cultura.....	59
5.4.4. Valore consegnato.....	59
6. Casi studio – Non solo software: la metodologia Agile al di fuori del mondo IT e nella vita reale	60
6.1. Caso studio 4 – Agile coaching per la scuola, una didattica a prova di futuro	60
6.2. Come l’Agile viene utilizzato nella vita di tutti i giorni.....	62
7. Conclusioni	64
Bibliografia e sitografia.....	65

Indice delle figure

Figura 1: Fasi salienti Agile.....	10
Figura 2: Interesse Agile nel corso degli anni	11
Figura 3: Interesse certificazioni Agile nel corso degli anni	11
Figura 4: Pro e Contro Agile.....	15
Figura 5: Pro e Contro Waterfall.....	16
Figura 6: Confronto Waterfall VS Agile	17
Figura 7: Kanban dashboard	20
Figura 8: Storia di Scrum.....	24
Figura 9: Metodologie tradizionali VS Scrum.....	25
Figura 10: Scum Team	27
Figura 11: Gli artefatti di Scrum	33
Figura 12: Grafico Burndown.....	35
Figura 13: Processo Scrum	39
Figura 14: Esempio finestra Backlog Jira.....	47
Figura 15: Esempio finestra user story Pivotal Tracker	48
Figura 16: Dashboard Nutcache.....	49
Figura 17: Dashboard Scrumwise	50
Figura 18: Miroboard	51

Ringraziamenti

A mio Padre, che mi ha mostrato il carattere per poter affrontare qualsiasi sfida, a Te saranno dedicati tutti i miei traguardi.

Alla mia famiglia, modello di unità e disponibilità, che mi ha sempre sostenuto.

Ai miei amici tutti, lontani e vicini: ognuno di voi mi ha dato un supporto importantissimo in questo percorso.

1. Introduzione

Il tema di questa tesi, ovvero la Metodologia Agile, nasce dalla mia esperienza personale di tirocinio avvenuta in Accenture. Infatti, durante la collaborazione con il team in cui sono stato inserito, ho avuto la possibilità di seguire da vicino l'avanzamento di un progetto gestito secondo la Metodologia Agile ed in particolare utilizzando il framework Scrum.

L'interesse verso la gestione dei progetti, nato grazie a questa esperienza, mi ha portato a ricercare, in maniera più approfondita, le caratteristiche essenziali dell'approccio Agile. Di fatti, questo elaborato prova ad analizzare tutte le fasi e i requisiti necessari per utilizzare tale metodologia e ne evidenzia le varie applicazioni.

Per aiutare il lettore nella comprensione dell'argomento, che per alcuni versi può sembrare astratto, la tesi si apre con una descrizione della metodologia Agile seguendo cronologicamente le fasi salienti della stesura del suo manifesto e delle precedenti e successive applicazioni. Inoltre, all'interno della tesi viene fornito un confronto tra la metodologia Agile e le metodologie tradizionali di gestione dei progetti, evidenziandone vantaggi e svantaggi.

Successivamente all'interno dell'elaborato, è presente un focus sul framework Scrum, di cui viene offerta un'analisi dettagliata di tutte le sue dinamiche e cerimonie.

Infine, vengono forniti dei casi studio reali legati all'attività di Accenture e non, per comprendere a pieno l'utilità e la finalità della metodologia Agile.

2. La metodologia Agile

Nel febbraio 2001, un gruppo di 17 sviluppatori di software si è riunito nella pittoresca località sciistica di Snowbird, nello Utah, per discutere degli approcci emergenti nel loro mestiere. Hanno elaborato alcune idee che ritenevano sufficientemente valide da essere messe su carta. Avevano certamente delle ambizioni. Ma non avevano idea della portata delle loro parole.

Era loro la convinzione che lo sviluppo del software dovesse essere più adattabile e reattivo al cambiamento - o "agile", come il gruppo decise di chiamarlo.

Il Manifesto Agile ha trasformato questo concetto centrale in un grido d'allarme per l'industria del software. Il gruppo ha quindi prodotto una dichiarazione sintetica di principi di supporto che enfatizza la collaborazione spontanea, la semplicità e la ricerca dell'eccellenza tecnica per soddisfare le richieste in evoluzione e raggiungere gli obiettivi.

2.1. Storia

2.1.1. Pre 2001 – Pratiche e Metodi si sviluppano attraverso l'esperienza

Molti attribuiscono l'inizio dello sviluppo software Agile, e in un certo senso dell'Agile in generale, all'incontro avvenuto nel 2001, quando è stato coniato il termine "Agile software development".

Tuttavia, le persone hanno iniziato a lavorare in modo Agile prima di quell'incontro del 2001. A partire dalla metà degli anni Novanta, c'erano diversi professionisti, sia che lavoravano all'interno di organizzazioni di sviluppo di prodotti software, sia consulenti che aiutavano le organizzazioni a costruire software, i quali ritenevano che le modalità di costruzione dei software non essendo funzionali, andassero cambiate.

Questi sviluppatori hanno iniziato a mescolare idee vecchie e nuove e, quando hanno trovato una combinazione funzionante, hanno creato una metodologia per il loro team e che si potesse adattare anche a progetti differenti.

Tali metodologie enfatizzavano la stretta collaborazione tra il team di sviluppo e gli stakeholder aziendali, la consegna frequente del valore aziendale, team affiatati e auto-organizzati che utilizzavano modi intelligenti di creare, confermare e consegnare il codice.

Gli ideatori di queste metodologie, ritenendo che altri potessero essere interessati a condividere parte dei vantaggi sperimentati, hanno creato dei framework per diffondere le idee ad altri team in altre organizzazioni e contesti. Iniziano così a comparire framework come Scrum, oggetto di trattazione di questa tesi, Extreme Programming, Feature-Driven Development (FDD) e Dynamic Systems Development Method (DSDM), tra gli altri.

La diffusione delle idee in questo periodo è stata elevata e i diversi approcci sono stati potenziati. Vari sviluppatori e project managers hanno preso in prestito i framework originali e li hanno modificati con pratiche diverse per adeguarli ai loro contesti.

Nella Figura 1 è possibile vedere, in maniera schematizzata, le fasi salienti di questo periodo.

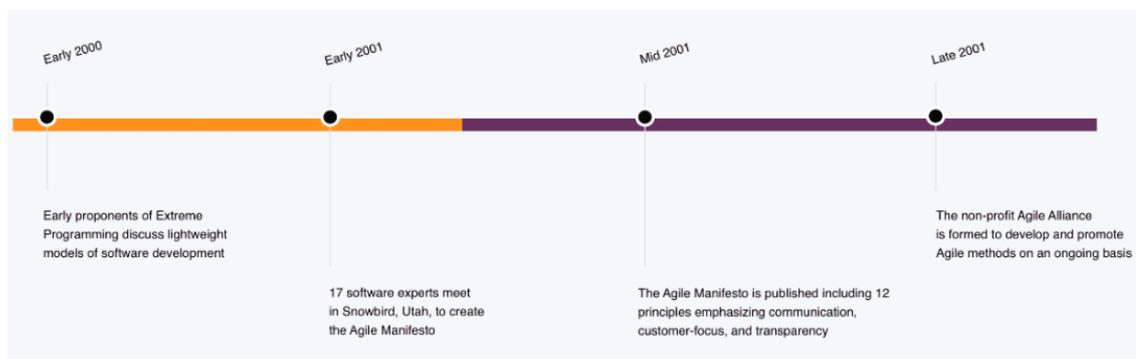


Figura 1: Fasi salienti Agile

Fonte: <https://www.thoughtworks.com/perspectives/edition1-agile>

2.1.2. Post 2001 – L'adozione diventa mainstream

Dopo l'avvenuta stesura, durante l'incontro del 2011 a Snowbird, della dichiarazione sulle metodologie Agile, Ward Cunningham pubblicò il Manifesto Agile e successivamente i 12 Principi online su AgileManifesto.org dove chiunque poteva firmare per dimostrare il proprio sostegno all'idea.

Agile Alliance è stata costituita ufficialmente alla fine del 2001 come luogo in cui, chi sviluppava software e chi aiutava gli altri a sviluppare software, poteva esplorare e condividere idee ed esperienze.

I team e le organizzazioni hanno iniziato ad adottare Agile, guidati principalmente dagli sviluppatori dei team. Gradualmente, gli stessi manager di questi team hanno iniziato a introdurre approcci Agile nelle loro organizzazioni, contribuendo a creare un ecosistema che condivide consulenze, formazione, framework e strumenti.

Con la crescita dell'ecosistema e la diffusione delle idee Agile, alcuni hanno perso di vista i valori e i principi enunciati nel manifesto e nei principi corrispondenti. Invece di seguire una mentalità "agile", hanno iniziato a insistere sulla rigidità della modalità di esecuzione di certe pratiche. Con questo approccio si ha difficoltà a lavorare in modo Agile che invece richiede una modalità che si adatti al contesto in cui è utilizzata.

Nella Figura 2 è possibile vedere graficamente il trend crescente dell'interesse riguardo all'Agile nel corso degli anni e allo stesso modo nella Figura 3 è presente quello delle certificazioni Agile.



Figura 2: Interesse Agile nel corso degli anni Fonte: www.thoughtworks.com/perspectives/edition1-agile

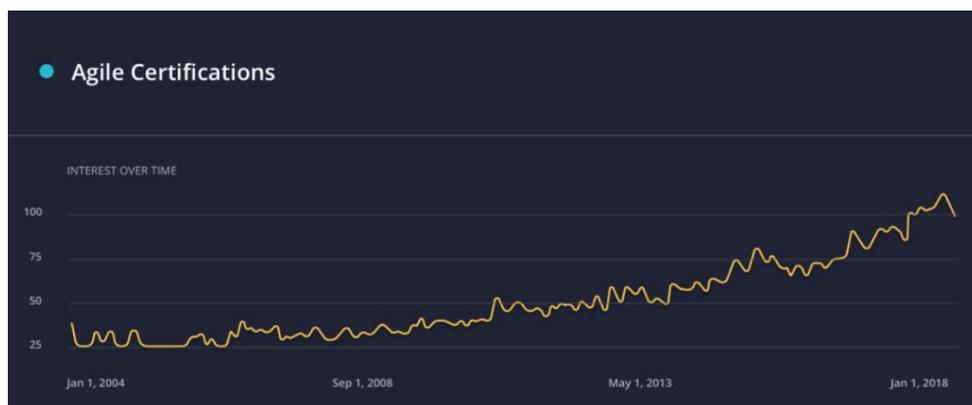


Figura 3: Interesse certificazioni Agile nel corso degli anni Fonte: www.thoughtworks.com/perspectives/edition1-agile

2.2. Cos'è la metodologia Agile

Sono trascorsi 20 anni dalla pubblicazione del Manifesto per lo sviluppo agile del software e ancora più tempo da quando è nata l'idea, eppure nel settore non c'è ancora una chiara comprensione di cosa sia veramente Agile.

"Troppi team che dichiarano di essere 'Agili' non lo sono. Mi è capitato di sentire alcuni che dichiarano di essere 'Agile' solo perché fanno qualche pratica Scrum e usano uno strumento di ticketing. Ci sono molti equivoci sull' Agile", ha dichiarato Andy Hunt, uno degli autori del manifesto e co-autore del libro "The Pragmatic Programmer".

Secondo Dave Thomas, altro co-autore di "The Pragmatic Programmer" e del Manifesto Agile, il modo in cui si parla oggi di Agile è sbagliato: Agile è un aggettivo, non un sostantivo, pertanto c'è una profonda differenza. "L'essenza del manifesto è che tutto cambia e che il cambiamento è inevitabile. Eppure, una volta che si inizia a parlare di 'Agile' come di una cosa, la si è congelata", ha dichiarato Thomas.

Un elemento non presente nel Manifesto Agile è una vera e propria definizione di Agile. In uno dei suoi libri, "Practices of an Agile Developer", Hunt definisce lo sviluppo Agile come un approccio che "utilizza feedback per apportare costanti modifiche in un ambiente altamente collaborativo".

"Credo che sia proprio così. Si tratta di feedback e aggiustamenti continui. Non si tratta di riunioni di standup, né di ticket, né di schede Kanban, né di story points", ha affermato Hunt.

Ma Thomas ritiene che ci sia una buona ragione per cui una definizione non è stata inclusa nel manifesto, e cioè che Agile è contestuale. "L'Agile deve essere personale per un particolare team, in un particolare ambiente, probabilmente su un particolare progetto, perché progetti diversi avranno modi diversi di lavorare", ha osservato. "Non si può andare a comprare un chilo di Agile da qualche parte. Non esiste, e nemmeno un team può andare a comprare un corso di formazione di due giorni su come essere Agile".

Dave West, CEO e product owner di Scrum.org, ritiene che il vero motivo per cui le persone non capiscono l'Agile sia da ricercare nei sistemi sociali, non nella pratica, nel lavoro vero e proprio o nei problemi che si vogliono risolvere. Questo perché i sistemi sociali sono ancora basati su scadenze e quantità di lavoro piuttosto che su elementi qualitativi e di contenuto.

Per tornare al vero significato di Agile, bisogna abbandonare i termini e tornare ai quattro principi fondamentali, secondo Danny Presten, fondatore della comunità Agile. Fornire valore incrementale, avere una buona visione del lavoro, essere in grado di stabilire le

priorità e migliorare i cicli è "ciò che rende veramente Agile. Non sono i termini. Più le persone si concentrano sui principi e meno si concentrano sui termini, migliore sarà Agile", ha dichiarato Presten.

Presten ha paragonato l'apprendimento di Agile all'apprendimento di uno strumento. "Quando si inizia, si leggono gli spartiti. Aiuta a dare slancio e chiarezza, ma se ci si ferma a leggere gli spartiti senza pensare e a fare le cose per bene, allora c'è un problema", ha proseguito.

Un buon modo di vedere la cosa è quello di osservare i feedback che si ricevono. Secondo Thomas "L'unico modo per essere agili è adattarsi costantemente all'ambiente. A volte questo può avvenire minuto per minuto, a volte giorno per giorno e a volte settimana per settimana, ma se non ricevete feedback il più spesso possibile, allora non state facendo Agile", ha affermato.

Cockburn ha spiegato che ci sono state tre ondate di Agile. La prima era a livello di team, poi l'Agile ha iniziato a spostarsi a livello di organizzazione, ora siamo nella terza ondata, che è a livello globale. La scala globale comprende dipartimenti finanziari, dipartimenti HR, dipartimenti legali, intere catene di fornitura, governi, progetti sociali, team distribuiti e persino diverse aree geografiche. "Non si tratta solo di team. Anzi, non si tratta solo di organizzazioni. Non è solo software. Non si tratta di prodotti. È un'adozione globale", ha detto Cockburn.

Ha inoltre spiegato che il motivo per cui si guarda all'Agile su scala globale è la VUCA: volatilità, incertezza, complessità e ambiguità. Ha detto che il mondo è "VUCA" e questo è diventato ancora più evidente con il COVID, le chiusure e le modalità di lavoro distribuite per ogni persona, team, settore, azienda e paese. Tutti devono avere la capacità di muoversi e cambiare direzione rapidamente e con facilità.

"Questo è il mondo nuovo e attuale. Sta accadendo. L'Agile ha smesso da tempo di riguardare solo il software; ora è completamente generale. Si possono considerare questi valori e principi ed estrapolarli a qualsiasi impresa", ha detto Cockburn.

Sebbene sia normale che le idee si volatilizzino nel tempo, Robert Martin, uno degli autori del Manifesto Agile e autore di "Clean Agile: Back to Basics", ritiene che il significato di Agile sia diventato più che diluito, perdendo la sua vera natura.

Ha spiegato che Agile è stato originariamente sviluppato da programmatori per programmatori, ma dopo un paio d'anni c'è stato uno spostamento per portare Agile alla gestione dei progetti.

"L'afflusso di project manager nel movimento Agile ne ha cambiato l'enfasi in modo piuttosto drammatico. Invece di essere un'idea di piccole squadre per realizzare progetti relativamente piccoli, Agile si è trasformato in una metodologia di gestione dei progetti", ha detto Martin. Ha spiegato che l'obiettivo originario dell'incontro di Snowbird, dove è nato il Manifesto Agile, era quello di colmare il divario tra business e tecnologia, ma il lato business ha preso il sopravvento sul movimento Agile e non ha coinvolto il lato tecnico.

Risulta quindi chiaro come ormai l'Agile abbia ampliato il suo campo d'azione trovando un ruolo particolarmente chiave nell'ambito del Project Management.

2.3. Confronto tra Agile e metodologie tradizionali (Waterfall)

Agile e Waterfall sono due note metodologie di gestione dei progetti. Entrambe sono popolari nell'ambito dello sviluppo di software, ma ciascuna si adatta meglio a diversi tipi di progetti. La differenza principale è che Waterfall è un sistema di lavoro lineare che richiede al team di completare ogni fase del progetto prima di passare a quella successiva, mentre Agile incoraggia il team a lavorare contemporaneamente su diverse fasi del progetto.

Metodologia Agile

- **Approccio:** Interazione frequente con gli stakeholder
- **Flessibilità:** Elevata
- **Richieste:** Spiccato spirito di squadra e scadenze a breve termine

La metodologia Agile è stata sviluppata come risposta alla struttura più rigida di Waterfall. Di conseguenza, è una forma di gestione dei progetti molto più fluida. Un progetto di sviluppo software può richiedere anni per essere completato e la tecnologia può cambiare in modo significativo durante questo periodo. Agile è stata sviluppata come una metodologia flessibile che accetta di rispondere reattivamente a cambiamenti di direzione anche in fase avanzata del processo, oltre a tenere conto dei feedback degli stakeholder durante tutto il processo.

In Agile, il team lavora su fasi del progetto in contemporanea, spesso con scadenze a breve termine. Inoltre, è il team, piuttosto che un project manager, a guidare la direzione del progetto. Questo può rendere la squadra più motivata e produttiva, ma richiede anche una buona capacità di autogestione (vedi Figura 4).

PRO	CONTRO
Le scadenze a breve termine favoriscono la produttività e l'efficienza	Poiché i membri del team lavorano su più fasi alla volta, è possibile che si verifichino sovrapposizioni o sforzi inutili nelle fasi successive se una fase iniziale deve essere modificata
C'è molta flessibilità per cambiare la direzione del progetto e sperimentare nuove soluzioni	Le consegne non sono un requisito per passare alla fase successiva. Può essere più difficile garantire che l'intero team sia sulla stessa lunghezza d'onda, soprattutto se si tratta di un team numeroso con diversi reparti. Ciò significa anche che il lavoro può andare perso o essere comunicato in modo errato tra i membri del team, soprattutto quando qualcuno lascia o si unisce al team nel bel mezzo del progetto
La metodologia è orientata al cliente, il che significa che il team condivide i progressi e incorpora i feedback nel processo	La tempistica del progetto è più difficile da determinare fin dall'inizio ed è anche più suscettibile a variazioni

Figura 4: Pro e Contro Agile

Metodologia Waterfall

- **Approccio:** Gli obiettivi e i risultati sono stabiliti fin dall'inizio
- **Flessibilità:** Bassa
- **Richieste:** Completare le consegne per passare alla fase successiva

La metodologia Waterfall è una forma lineare di gestione dei progetti, ideale per progetti in cui il risultato finale è chiaramente stabilito fin dall'inizio. Le aspettative per il progetto e i risultati di ogni fase devono essere chiari e sono necessari per passare alla fase successiva (vedi Figura 5).

PRO	CONTRO
Fornisce un piano concreto del progetto dall'inizio alla fine	Poiché ogni fase del progetto deve essere completata prima di passare alla fase successiva, il processo può richiedere più tempo
Il team stabilisce i requisiti del progetto fin dalle prime fasi, il che può far risparmiare tempo	Potreste accorgervi di un problema in una fase solo quando siete già passati a quella successiva. Ciò significherebbe ripercorrere ogni fase e verificare dove si è verificato l'errore o l'incongruenza, il che può essere un processo che richiede molto tempo
Ogni fase del progetto richiede una consegna per passare alla fase successiva, rendendo il flusso di lavoro più strutturato	La metodologia Waterfall richiede di delineare il progetto dall'inizio alla fine prima di iniziare. Questo non consente molta flessibilità o cambiamenti. Inoltre, può diventare problematico se le parti interessate non sono d'accordo sulla visione del progetto e lo scoprono solo durante l'esecuzione o in una fase successiva

Figura 5: Pro e Contro Waterfall

Agile e Waterfall a confronto

La metodologia Waterfall prevede una progressione lineare del progetto, quindi è più adatta a progetti con un obiettivo finale definito. Se il titolare di un progetto ha una visione chiara e specifica di un'applicazione, ad esempio, ed è sicuro che non cambierà durante lo sviluppo del progetto, le metodologie Waterfall potrebbero essere un buon sistema da seguire.

Invece, Agile lascia molto spazio per adattarsi e cambiare rotta durante lo sviluppo del progetto. È più adatto a progetti in cui il risultato può dipendere da ulteriori ricerche o test.

Il budget per i progetti che utilizzano le metodologie Waterfall tende a essere meno flessibile perché il progetto è tracciato fin dall'inizio. Con Agile, c'è più spazio per cambiare direzione durante lo sviluppo del progetto, quindi anche il budget è soggetto a modifiche. Allo stesso modo, la tempistica con Waterfall è stabilita fin dall'inizio, mentre con Agile è più flessibile e dipende dallo sviluppo del progetto.

Nella **Errore. L'origine riferimento non è stata trovata.** è possibile vedere quali sono le principali differenze tra le due metodologie in diversi ambiti di progetto.

	Waterfall	Agile
Timeline	Il Waterfall prevede una tempistica fissa. L'idea è che l'inizio e la fine del progetto siano già tracciati fin dal principio	Piuttosto che un calendario fisso, il programma si adatta all'avanzamento del progetto. Il Manifesto Agile afferma che i membri del team devono "consegnare software funzionante frequentemente, da un paio di settimane a un paio di mesi, con una preferenza per i tempi più brevi"
Coinvolgimento del cliente	Una volta stabilito l'obiettivo finale, il Waterfall non coinvolge il cliente o il proprietario del progetto durante il processo, a parte per specifici check-in o per le consegne	Una parte fondamentale di Agile è il coinvolgimento dei clienti nello sviluppo del progetto in ogni fase. Il Manifesto Agile afferma: "La nostra massima priorità è soddisfare il cliente attraverso la consegna tempestiva e continua di software di valore". Pertanto, i proprietari delle aziende devono essere coinvolti e fornire feedback al team di sviluppo del software durante le varie fasi del progetto.
Flessibilità	Waterfall non è flessibile come Agile, perché ogni fase deve essere completata prima di passare alla fase successiva. Inoltre, il progetto viene pianificato in anticipo, il che rende questo sistema di gestione ideale per i team che hanno una visione chiara di dove sono diretti dall'inizio alla fine	La flessibilità è parte integrante del metodo Agile. Agile dà valore a brevi periodi di lavoro. Il metodo è favorevole all'adattamento, incorporando nuove informazioni anche in una fase successiva del progetto
Budget	Fisso. Il budget per i progetti che utilizzano la metodologia Waterfall è generalmente fisso. Poiché il progetto è determinato dall'inizio alla fine, c'è meno spazio per modificare il budget a metà progetto.	Flessibile. Agile è aperto all'adattamento, incoraggia la sperimentazione e accoglie i cambiamenti, anche nelle fasi successive del progetto. Per questo motivo, il budget tende a essere più flessibile.

Figura 6: Confronto Waterfall VS Agile

Per concludere questa analisi, ecco elencati alcuni vantaggi della metodologia Agile rispetto alle tradizionali:

- **Meno incline all'errore**

Il Waterfall si basa molto sui requisiti iniziali. Tuttavia, se questi requisiti non sono documentati con precisione, o se c'è stato un fraintendimento sui dettagli di ciò che il cliente voleva, le cose diventano molto difficili. Non è così con Agile: i requisiti vengono controllati e confermati durante tutto il progetto.

- **Più flessibile**

Una volta completata una fase in Waterfall, è difficile tornare indietro e apportare modifiche. Al contrario, Agile costruisce una versione funzionante dell'intero progetto (un MVP), in modo che il cliente possa definire il modo in cui viene costruito. Vedere una versione funzionante fin dalle prime fasi del progetto permette al cliente di dire "mi piace questo, ma

non mi piace quello", e quindi di modellare il prodotto in base alle sue esigenze. Questo è più difficile da fare con Waterfall, perché il cliente deve delineare tutte le sue preferenze in anticipo, senza vedere una versione funzionante.

- **Prodotto finale più prevedibile**

Con il Waterfall, il prodotto viene testato principalmente alla fine del progetto. Se le esigenze del cliente non sono state colte bene all'inizio o sono cambiate dall'inizio del progetto, il collaudo può arrivare troppo tardi nel ciclo per apportare grandi modifiche. Il cliente deve quindi trovare un budget aggiuntivo per ottenere il prodotto di cui ha bisogno. Con Agile, il collaudo avviene regolarmente durante l'intero processo, in modo che il cliente possa verificare periodicamente che il prodotto sia quello che aveva immaginato. In questo modo è più probabile che il progetto venga portato a termine nei tempi e nei budget previsti.

- **Maggiore coinvolgimento del cliente**

Agile considera il cliente come parte del team di implementazione e lo coinvolge in ogni fase del processo. Al contrario, Waterfall tende a dedicare molto tempo al cliente all'inizio, cercando di documentare tutti i requisiti percepiti. Ma una volta fatto questo, il team di implementazione di solito prende il sopravvento.

2.4. Introduzione ai framework Agile

Un framework Agile è un approccio specifico alla pianificazione, alla gestione e all'esecuzione del lavoro. I framework Agile si dividono tipicamente in due categorie: framework progettati per i team e framework progettati per aiutare le organizzazioni a praticare Agile su scala, tra più team.

I framework Agile includono anche linee guida per funzioni tradizionalmente non-Agile, come la finanza e le risorse umane. Con la continua espansione del movimento Agile e l'emergere di nuove applicazioni, anche i framework Agile continuano a evolversi.

2.4.1. Approcci Agile a livello di team

SCRUM

Scrum, su cui si concentrerà questa tesi nei prossimi capitoli, è uno dei framework Agile più noti per i team. La Scrum Guide definisce Scrum come "... un framework di riferimento

all'interno del quale le persone possono affrontare problemi complessi e di adattamento, realizzando in modo produttivo e creativo prodotti del massimo valore possibile".

Per definizione, Scrum è pensato per essere leggero e facile da capire. Alcuni direbbero, tuttavia, che è difficile da padroneggiare. L'elemento più riconoscibile di Scrum è l'uso di iterazioni a tempo: I team Scrum operano in periodi di tempo prestabiliti, chiamati sprint, che di solito durano tra le due e le quattro settimane.

Operare in sprint consente ai team di fornire risultati in modo rapido e prevedibile, mantenendo al contempo l'agilità necessaria per modificare i progetti.

XP

L'Extreme Programming (o XP) è un altro framework Agile a livello di team che affonda le sue radici nello sviluppo del software. Come la maggior parte degli approcci Agile, XP consente rilasci frequenti in brevi periodi di sviluppo che incoraggiano il cambiamento quando necessario.

L'XP è meno regolato di molti framework Agile e segue un insieme di valori, piuttosto che di fasi. I valori fondamentali di XP sono:

- Semplicità
- Comunicazione
- Feedback coerenti
- Rispetto

L'Extreme Programming richiede agli sviluppatori di pianificare e comprendere innanzitutto le user stories dei clienti, ovvero le loro descrizioni informali delle caratteristiche e dei requisiti delle funzionalità.

KANBAN

Kanban deriva dalla parola giapponese che significa "segnale visivo" o "scheda". Sebbene le radici del framework Kanban siano nell'industria manifatturiera, questo framework Agile è stato applicato ampiamente allo sviluppo di software e ad altri tipi di lavoro non manuale.

La metodologia utilizza schede fisiche o digitali per rappresentare il processo unico di un team o di un'organizzazione. Gli elementi essenziali di Kanban sono i seguenti:

- Gli elementi di lavoro sono rappresentati da schede su una lavagna Kanban, come le note adesive su una lavagna.
- Ogni fase del processo è rappresentata da una corsia o colonna verticale. La versione più elementare di un processo è "Da fare", "In corso" e "Fatto".

- Le schede vengono poi spostate da sinistra a destra sulla lavagna per indicare la posizione di ciascun elemento del processo.
- Le schede possono contenere una serie di informazioni sul compito, tra cui lo stato, le date di scadenza, chi è assegnato alla scheda, ecc.

Grazie alla visibilità condivisa, Kanban consente ai team di collaborare in modo più efficace e di operare con maggiore agilità. Il Kanban è spesso utilizzato come strumento dai team Lean e Agile e può essere incorporato in altri framework Agile. Nella Figura 7 è possibile vedere l'esempio di una dashboard Kanban.

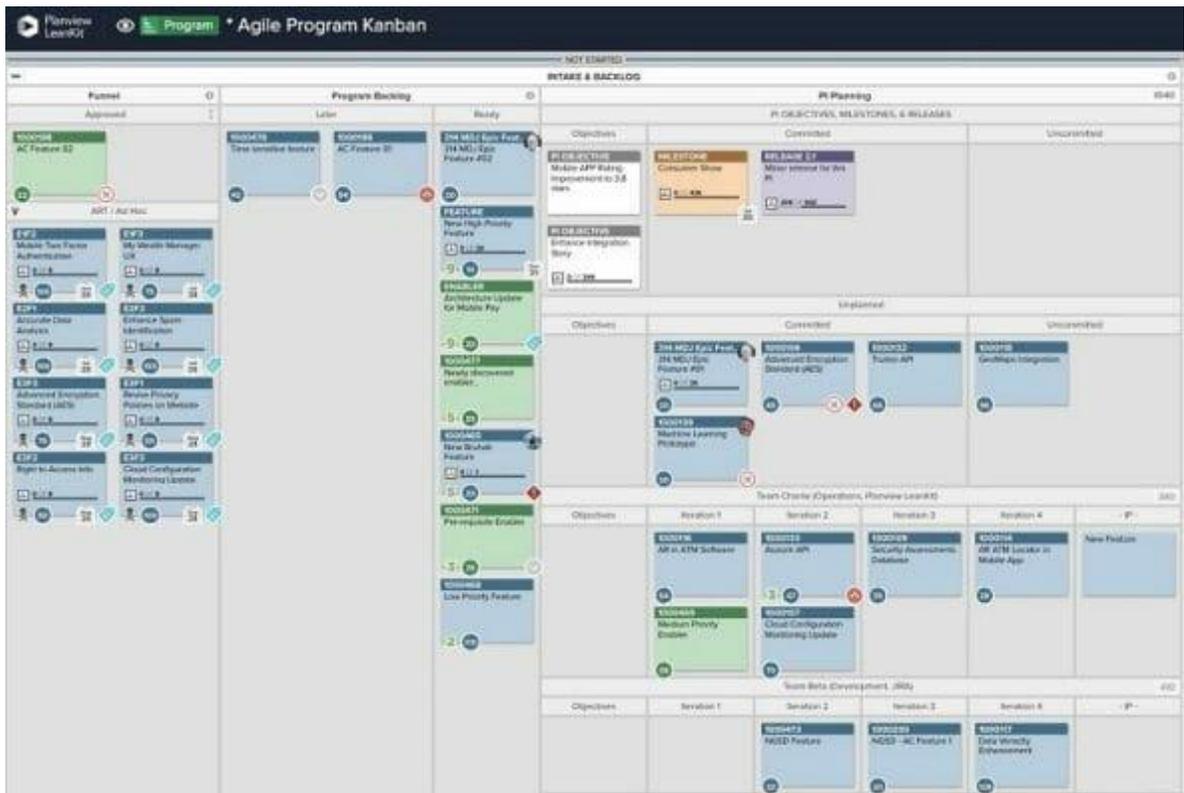


Figura 7: Kanban dashboard Fonte: <https://www.planview.com/resources/guide/what-is-agile-program-management/agile-frameworks>

SCRUMBAN

Scrumban è un framework Agile ibrido tra Scrum e Kanban, nato per soddisfare le esigenze dei team che volevano ridurre al minimo il frazionamento del lavoro e adottare un sistema pull-based. Combinando elementi di Scrum e Kanban, questo framework Agile offre ai team la flessibilità necessaria per adattarsi alle esigenze degli stakeholder e della produzione senza sentirsi sovraccaricati dalla metodologia di progetto.

Scrumban offre la struttura di Scrum con la flessibilità e la visualizzazione di Kanban, rendendolo un approccio estremamente versatile alla gestione dei flussi di lavoro.

Scrumban può anche essere utilizzato come anello di congiunzione per i team che desiderano passare da Scrum a Kanban.

2.4.2 Approcci Agile su scala aziendale

Sono molti i fattori che influenzano la capacità di un'organizzazione ad adottare Agile su scala aziendale, tra cui i cambiamenti culturali, tecnologici e di gestione del lavoro. Di conseguenza, i framework Agile aziendali sono diventati sempre più popolari per le organizzazioni che si trovano in questa fase di trasformazione nel loro percorso.

SAFe

SAFe è un acronimo che sta per: Scaled Agile Framework. È uno dei vari framework Agile progettati per guidare le aziende a espandere le attività Agile, al di là dei singoli team. SAFe prende in prestito concetti da molti degli approcci Agile a livello di team discussi in precedenza.

Più nel dettaglio, SAFe aiuta le grandi organizzazioni a superare le sfide associate all'implementazione, all'estensione e all'utilizzo di metodologie e pratiche Agile nei vari team e reparti all'interno di un'organizzazione.

SAFe è stato sviluppato da e per i professionisti del settore, sfruttando tre conoscenze principali:

- Sviluppo software agile
- Sviluppo snello del prodotto
- Pensiero orientato ai sistemi

SAFe adotta un approccio strutturato per consentire alle aziende di creare un allineamento, migliorare la collaborazione, massimizzare la flessibilità e aumentare la prevedibilità tra i team Agile.

DISCIPLINED AGILE (DA)

Disciplined Agile, o DA, è un framework Agile che fornisce una guida snella per aiutare i team a praticare Agile su scala. DA enfatizza un approccio "people-first" ed è tipicamente pensato come un ibrido di XP, Scrum, Kanban e altri framework Agile.

Disciplined Agile è più adatto alle organizzazioni che hanno già familiarità con i principi Agile, perché non offre molto in termini di insegnamento dei principi o delle pratiche Agile di base.

Esistono comunque molti altri modelli di approcci Agile su scala aziendale, ma per riassumere, tutti questi modelli sono modalità di utilizzo di framework Agile per team (spesso Scrum) adattati alle esigenze di progetto o dell'organizzazione e allargati a più team che comunque lavorano per lo stesso obiettivo. Famoso è il caso del modello Spotify, che con la sua applicazione e documentazione, ha posto le basi per lo sviluppo dello *Scrum@Scale*.

3. Il framework Scrum

Spesso si pensa che Scrum e Agile siano la stessa cosa poiché Scrum è incentrato su un principio fondamentale dell'Agile, il miglioramento continuo. Tuttavia, Scrum è un framework per portare a termine il lavoro, mentre Agile è una mentalità.

Il framework Scrum è euristico; si basa sull'apprendimento continuo e sull'adattamento a fattori fluttuanti. È consapevole che il team ha poca conoscenza all'inizio di un progetto e che si evolverà attraverso l'esperienza. Scrum è strutturato per aiutare i team ad adeguarsi alle mutevoli condizioni e alle richieste degli utenti, con la ridefinizione delle priorità incorporata nel processo e cicli di rilascio brevi in modo che il team possa imparare e migliorare costantemente.

Sebbene Scrum sia un framework strutturato, non è del tutto rigido. La sua esecuzione può essere adattata alle esigenze di qualsiasi organizzazione.

Ne “La Guida a Scrum - La guida definitiva a Scrum: Le regole del gioco” sviluppata e mantenuta dai creatori di Scrum: Ken Schwaber e Jeff Sutherland è presente la definizione ufficiale di Scrum.

Scrum (n): Un framework che consente alle persone di risolvere problemi complessi di tipo adattivo e, al tempo stesso, di creare e rilasciare prodotti in modo efficace e creativo del più alto valore possibile.

Successivamente la guida prosegue fornendo le caratteristiche principali di questo framework Agile.

Scrum è:

- Leggero

- Semplice da comprendere
- Difficile da padroneggiare

Scrum è 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; 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 tanto da poter migliorare continuamente il prodotto, il team e l'ambiente di lavoro.

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

Le regole di Scrum legano insieme i ruoli, gli eventi, gli artefatti governando le relazioni e le interazioni tra essi.

3.1. Cenni storici e fondamenti teorici

Scrum non è un acronimo, ma deriva dal concetto di mischia (inglese: Scrum) nel rugby. L'origine di Scrum come struttura risale a un articolo dell'Harvard Business Review del 1986. "The New New Product Development Game" di Takeuchi e Nonaka, che si rifaceva ad aziende come Honda, Canon e Fuji-Xerox, descriveva come i prodotti innovativi e di prima classe vengono sviluppati in team interfunzionali con un "all-simultaneous approach". L'articolo sottolineava l'importanza dei team auto-organizzati e descriveva anche il ruolo del management nel processo di sviluppo.

Takeuchi e Nonaka hanno utilizzato metafore tratte dal mondo dello sport per descrivere il processo di sviluppo del prodotto. Hanno paragonato il processo di sviluppo classico a una corsa a staffetta. Ogni corridore era da solo e, alla fine della sua corsa, passava il testimone al corridore successivo. Questo approccio non porta necessariamente alla massima velocità e soprattutto alla massima flessibilità. In alternativa, è stato presentato un approccio olistico o rugbistico, in cui la squadra copre l'intera distanza insieme, con la palla che passa avanti e indietro in modo flessibile. Nel mondo competitivo di oggi, questo approccio garantisce un successo maggiore.

Successivamente Jeff Sutherland e il suo team di Easel Corporation hanno adattato il processo Scrum allo sviluppo del software nel 1993. Combinarono le idee dell'articolo di Takeuchi e Nonaka con i concetti di programmazione orientata agli oggetti, controllo

empirico del processo, sviluppo iterativo, incrementale e risultati della ricerca sulla produttività.

Due anni dopo Ken Schwaber pubblicò il primo documento su Scrum, dopo accese discussioni con Sutherland. Da allora Sutherland e Schwaber hanno pubblicato diversi libri su Scrum e rivisitano regolarmente "The Scrum Guide", documento di 20 pagine che ne spiega le basi.

Inoltre, entrambi facevano parte dei 17 sviluppatori che hanno elaborato il Manifesto Agile nel 2001 e successivamente si sono impegnati nella diffusione di questo framework (vedi Figura 8) istituendo organizzazioni e ampliando le capacità di Scrum.



Figura 9: Storia di Scrum

Fonte: <https://www.atlassian.com/agile/scrum>

Oggi Scrum è utilizzato principalmente nello sviluppo di software. Tuttavia, molte aziende hanno imparato ad apprezzare i vantaggi del framework anche in altri settori, come lo sviluppo di hardware, il marketing, le vendite o le risorse umane facendone crescere l'utilizzo rispetto alle metodologie tradizionali (vedi Figura 9). Scrum è adatto a tutti i casi in cui i team lavorano su nuovi progetti, prodotti o servizi. Aiuta a coordinare meglio il lavoro del team, ad accelerare lo sviluppo del prodotto e a convalidare tempestivamente le ipotesi.



Figura 10: Metodologie tradizionali VS Scrum

Fonte: <https://thecrmteam.com/5-reasons-why-agile-is-better-than-waterfall/>

3.1.1. La teoria di Scrum e i 5 valori

Scrum si basa sulla teoria del controllo empirico dei processi, o empirismo. Secondo questa teoria, la conoscenza deriva dall'esperienza e le decisioni si basano su ciò che si conosce. Scrum utilizza un approccio iterativo ed incrementale per ottimizzare la prevedibilità ed il controllo del rischio.

Tre sono i pilastri che sostengono ogni implementazione del controllo empirico di processo:

- **Trasparenza**

Gli aspetti significativi del processo devono essere visibili ai responsabili del risultato. La trasparenza richiede che tali aspetti siano definiti da uno standard comune, in modo che gli

osservatori condividano una comune comprensione di ciò che viene visto. Ad esempio, un linguaggio comune di riferimento al processo deve essere condiviso da tutti i partecipanti; chi esegue il lavoro e chi ispeziona l'Incremento risultante deve condividere una definizione comune di "Fatto" (vedi paragrafo "[3.4.1. Estensione di artefatti](#)").

- **Ispezione**

Chi utilizza Scrum deve ispezionare frequentemente gli artefatti di Scrum e l'avanzamento verso uno Sprint Goal con lo scopo di rilevare deviazioni indesiderate. Tali ispezioni non dovrebbero essere tanto frequenti da intralciare il lavoro stesso. Le ispezioni sono più utili quando eseguite diligentemente da chi ha l'abilità e la competenza necessaria a effettuarle rispetto ad un particolare stadio del lavoro.

- **Adattamento**

Se chi ispeziona verifica che uno o più aspetti del processo sono oltre i limiti accettabili e che il prodotto finale non potrà essere accettato, in quel caso, il processo o il materiale processato devono essere adattati. L'adattamento deve essere portato a termine il più rapidamente possibile per ridurre al minimo un'ulteriore deviazione.

Scrum prescrive quattro occasioni formali all'interno dello Sprint per l'ispezione e l'adattamento, come descritto nel paragrafo "[3.3. Le cerimonie di Scrum](#)":

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Quando i valori di **impegno, coraggio, concentrazione, apertura e rispetto** sono incarnati e vissuti dallo Scrum Team, i pilastri Scrum di trasparenza, controllo e adattamento prendono vita e creano fiducia per tutti. I membri dello Scrum Team imparano ed esplorano tali valori durante il lavoro svolto attraverso gli eventi, i ruoli e gli artefatti di Scrum.

L'uso proficuo di Scrum dipende da quanto chi lo utilizza diventa più propenso a vivere questi cinque valori. I membri della squadra di lavoro sono personalmente dediti al raggiungimento degli obiettivi dello Scrum Team. Hanno il coraggio di fare la cosa giusta e di lavorare sui problemi difficili. Ognuno si concentra sul lavoro dello Sprint e sugli obiettivi dello Scrum Team. Lo Scrum Team e le parti interessate concordano nell'essere disponibili al lavoro e pronti alle sfide relative alla sua esecuzione. I membri dello Scrum Team si rispettano reciprocamente come persone capaci ed indipendenti.

3.2. Scrum Team

Lo Scrum Team è formato dal Product Owner, dal Team di Sviluppo e dallo Scrum Master (vedi Figura 10:). Lo Scrum Team è auto-organizzato e cross-funzionale. I Team auto-organizzati scelgono come compiere al meglio il proprio lavoro, anzichè essere diretti da altre figure esterne al team. I team cross-funzionali hanno tutte le competenze necessarie per realizzare il lavoro senza dover dipendere da nessuno al di fuori del team. Il modello di team in Scrum è progettato per ottimizzare la flessibilità, la creatività e la produttività.

Gli Scrum Team rilasciano i prodotti in modo iterativo e incrementale, massimizzando le opportunità di feedback. I rilasci incrementali di prodotto “Fatto” garantiscono che, una versione potenzialmente utile del prodotto funzionante, sia sempre disponibile.

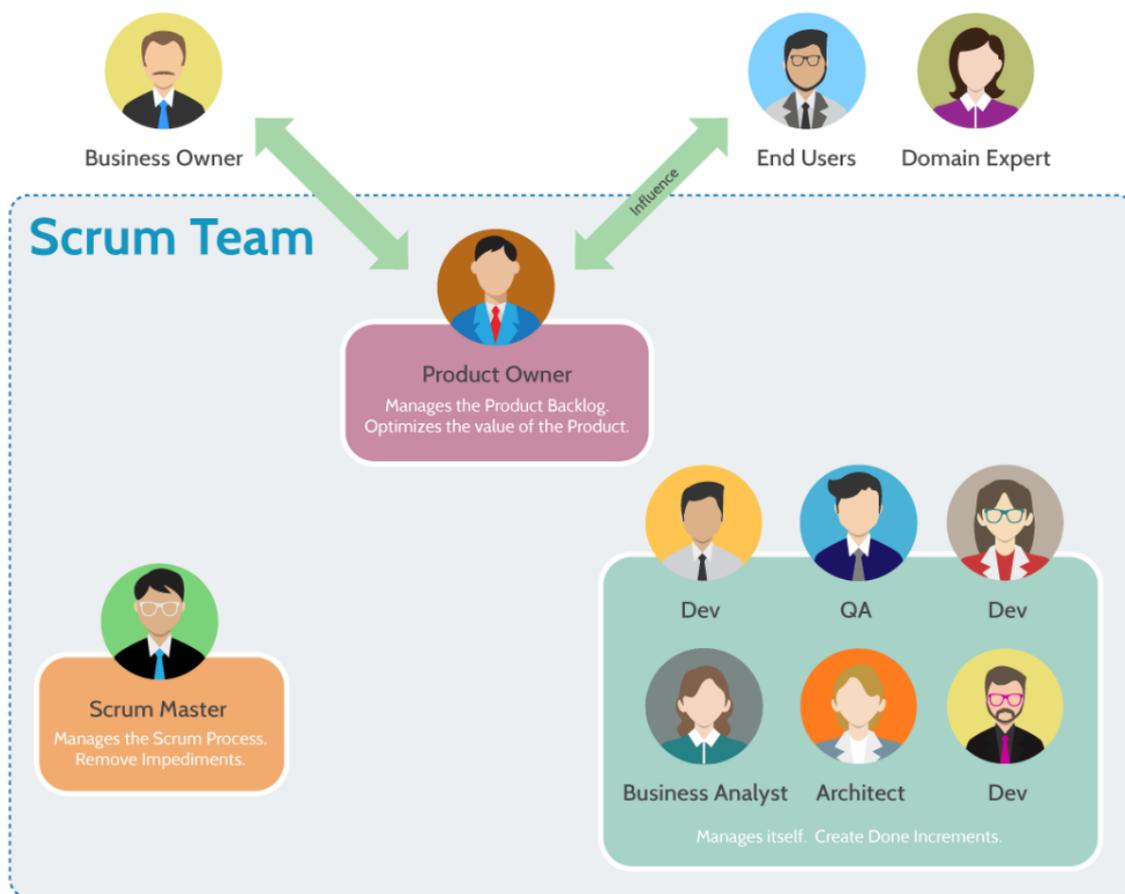


Figura 11: Scrum Team

Fonte: <https://www.visual-paradigm.com/scrum/what-is-scrum-team/>

3.2.1. Il Product Owner

Il Product Owner ha la responsabilità di massimizzare il valore del prodotto risultante dal lavoro svolto dal Team di Sviluppo. Le modalità di conseguimento di questo obiettivo variano di molto secondo l'organizzazione, gli Scrum Team e gli individui.

Il Product Owner è l'unica figura responsabile della gestione del Product Backlog, un elenco ordinato di ciò che è necessario fare per migliorare il prodotto (vedi paragrafo "[3.4. Gli artefatti di Scrum](#)"). Tale gestione include:

- Esprimere chiaramente gli elementi del Product Backlog;
- Ordinare gli elementi 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 elementi del Product Backlog al livello necessario.

Il lavoro sopra elencato può esser fatto dal Product Owner o dal Team di Sviluppo. Tuttavia, il Product Owner rimane il responsabile finale (accountable).

Il Product Owner è una singola 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.

Affinché il Product Owner possa agire con successo, all'interno dell'organizzazione tutti devono rispettare le sue decisioni. Le decisioni del Product Owner sono visibili nel contenuto e secondo l'ordine delle priorità del Product Backlog. Nessuno ha il permesso di dire al Team di Sviluppo di lavorare su un diverso insieme di requisiti, ed il Team di Sviluppo non ha il permesso di agire sulla base di ciò che dice chiunque altro.

Il Product Owner è l'unico che ha contatti all'esterno dello Scrum Team (clienti, utenti o consulenti); di conseguenza, è anche l'unico in grado di trasmettere informazioni ed obiettivi al Team di Sviluppo.

3.2.2. Il Team di Sviluppo

Il Team di Sviluppo è costituito da professionisti che lavorano per consegnare un Incremento "Fatto" di prodotto, potenzialmente rilasciabile alla fine di ogni Sprint. Alla Sprint Review

è richiesto un Incremento "Fatto" e soltanto i membri del Team di Sviluppo possono creare tale Incremento.

I Team di Sviluppo sono strutturati e autorizzati dall'organizzazione per coordinare e gestire il proprio lavoro. Ne deriva una sinergia che ottimizza l'efficienza e l'efficacia complessive del Team di Sviluppo.

I Team di Sviluppo hanno le seguenti caratteristiche:

- Sono auto-organizzati. Nessuno (neanche lo Scrum Master) dice al Team di Sviluppo come trasformare il Product Backlog in Incrementi di funzionalità potenzialmente rilasciabili;
- I Team di Sviluppo sono cross-funzionali, con tutte le competenze necessarie per creare un Incremento di prodotto;
- Scrum non riconosce alcun titolo ai membri del Team di Sviluppo, indipendentemente dal lavoro eseguito dalla persona;
- Scrum non riconosce alcun sotto-team nel Team di Sviluppo, indipendentemente dai particolari domini ai quali è necessario dedicarsi come test, architettura, operation o analisi di business;
- I singoli membri possono avere competenze specialistiche e aree di specializzazione, ma la responsabilità ricade sul Team di Sviluppo nel suo complesso

La dimensione ottimale del Team di Sviluppo deve essere abbastanza piccola da rimanere agile e abbastanza grande da completare un lavoro significativo all'interno dello Sprint. Avere meno di tre persone nel Team di Sviluppo riduce l'interazione; ciò comporta un minore guadagno in termini di produttività. Team di Sviluppo piccoli potrebbero avere dei limiti dovuti alla mancanza di skill durante lo Sprint, e impedire la consegna di un Incremento potenzialmente rilasciabile. Team di Sviluppo troppo grandi richiedono un eccessivo lavoro di coordinamento e generano elevata complessità rispetto a quella gestibile da un processo empirico. I ruoli del Product Owner e dello Scrum Master non sono inclusi nel conteggio, a meno che non stiano eseguendo il lavoro contenuto nello Sprint Backlog.

3.2.3. Lo Scrum Master

Lo Scrum Master è responsabile nel promuovere e sostenere Scrum, come definito nella Guida a Scrum. Gli Scrum Master sostengono chiunque a comprendere la teoria, le pratiche, le regole, ed i valori di Scrum.

Lo Scrum Master è un leader a servizio (servant-leader) dello Scrum Team e aiuta coloro al di fuori dello Scrum Team a discernere le interazioni utili e non, a modificarle per massimizzare il valore creato dallo Scrum Team.

Lo Scrum Master interagisce con ciascuna componente dello Scrum Team e, a seconda dell'unità a cui si rivolge fornisce determinati servizi di seguito elencati.

Per il *Product Owner*:

- Assicurare che obiettivi, portata e dominio del prodotto siano compresi nel miglior modo possibile da tutti nello Scrum Team;
- Trovare le tecniche per una gestione efficace del Product Backlog;
- Aiutare lo Scrum Team a comprendere la necessità di avere elementi del Product Backlog chiari e concisi;
- Comprendere la pianificazione del prodotto in un ambiente empirico;
- Assicurare che il Product Owner capisca come ordinare gli elementi del Product Backlog per massimizzare il valore;
- Comprendere e praticare l'Agile;
- Facilitare gli eventi Scrum come richiesto e necessario.

Per il *Team di Sviluppo*:

- Allenare il Team di Sviluppo all'auto-organizzazione e alla cross-funzionalità;
- Aiutare il Team di Sviluppo a creare prodotti di alto valore;
- Eliminare gli ostacoli all'avanzamento del Team di Sviluppo;
- Facilitare gli eventi Scrum come richiesto o necessario;
- Assistere il Team di Sviluppo in ambienti organizzativi in cui Scrum non è ancora pienamente adottato e compreso.

Per l'*Organizzazione*:

- Guidare ed assistere l'organizzazione all'adozione di Scrum;
- Pianificare le implementazioni di Scrum all'interno dell'organizzazione;
- Aiutare i dipendenti ed i portatori di interessi (stakeholder) a comprendere ed attuare Scrum e lo sviluppo del prodotto;
- Provocare il cambiamento che aumenta la produttività dello Scrum Team;
- Lavorare con altri Scrum Master per aumentare l'efficacia dell'applicazione di Scrum nell'organizzazione.

3.3. Le cerimonie di Scrum

Le cerimonie prescritte sono utilizzate per creare regolarità e ridurre necessità di incontri non definiti in Scrum. Tutti gli eventi sono limitati temporalmente (time-boxed), così da avere una durata massima fissa. Quando uno Sprint inizia, la sua durata è prefissata e non può essere né accorciata né allungata. Gli altri eventi possono invece terminare quando lo scopo è raggiunto, evitando sprechi di tempo.

Oltre allo Sprint in sé, che è un contenitore di tutti gli altri eventi, ogni cerimonia in Scrum è una occasione formale per ispezionare e adattare qualcosa. Questi eventi sono specificamente progettati per consentire trasparenza ed ispezione critiche. Un fallimento nell'includere una qualsiasi di queste cerimonie comporta una riduzione della trasparenza ed è un'occasione mancata per praticare l'ispezione e l'adattamento.

Di seguito viene fornita una breve descrizione delle cerimonie di Scrum, Si rimanda l'approfondimento al capitolo successivo ("[4. Gestione di un progetto con Scrum](#)").

Sprint Planning

Partecipanti: Team di Sviluppo, Scrum Master, Product Owner.

Quando: All'inizio di uno sprint.

Durata: Di solito circa un'ora per ogni settimana di iterazione. Uno sprint di due settimane inizia con una riunione di pianificazione di due ore.

Scopo: La pianificazione dello sprint prepara l'intero team al successo durante lo sprint. Durante la riunione, il Product Owner illustra un backlog prioritario del prodotto. Discute ogni elemento con il team di sviluppo che stima lo sforzo necessario. Il team di sviluppo fa quindi una previsione dello sprint; delinea la quantità e fissa la durata che il team necessita per completare il lavoro. Il tutto diventa quindi il backlog dello sprint.

La riunione di pianificazione dello sprint definisce i dettagli del lavoro da svolgere. Favorisce le discussioni e raccoglie il consenso sul piano d'azione. Una pianificazione efficace aumenta significativamente le possibilità di successo del team nel rispettare gli impegni dello sprint.

Daily Stand-up

Partecipanti: Team di Sviluppo, Scrum Master, Product Owner.

Quando: Una volta al giorno, in genere al mattino.

Durata: Non più di 15 minuti da trascorrere rigorosamente in piedi.

Scopo: Lo stand-up è pensato per informare rapidamente tutti su ciò che accade nel team. Il tono deve essere pacato, ma informativo. Ad ogni membro del team viene chiesto di rispondere alle seguenti domande:

- Cosa ho completato ieri?
- Su cosa lavorerò oggi?
- Sono bloccato da qualcosa?

Sprint review

Partecipanti: Team di Sviluppo, Scrum Master, Product Owner, Stakeholders.

Quando: Al termine dello sprint.

Durata: Di solito 60 minuti per ogni settimana di sprint. Per uno sprint di due settimane, due ore.

Scopo: La Sprint review è un momento di presentazione del lavoro del team. Può trattarsi di una riunione formale o informale. Questo è il momento in cui il team può celebrare i propri risultati, dimostrare il lavoro svolto durante lo sprint e ottenere un feedback immediato dagli stakeholder del progetto. Il lavoro deve essere completamente dimostrabile e soddisfare i requisiti di qualità del team per essere considerato completo e pronto per la revisione.

Retrospective

Partecipanti: Team di Sviluppo, Scrum Master, Product Owner, Stakeholders.

Quando: Al termine dello sprint.

Durata: Di solito 45 minuti per ogni settimana di sprint. Per uno sprint di due settimane, 90 minuti.

Scopo: Agile significa ottenere un feedback rapido per migliorare il prodotto e la mentalità di sviluppo. Le riunioni retrospective aiutano il team a riflettere e valutare i punti deboli e i punti di forza dell'operato, e trovare un piano di azione per problemi non risolti.

Queste riunioni non sono solo un momento di dibattito, ma anche di risoluzione. Il miglioramento continuo è ciò che sostiene e guida lo sviluppo di un team agile e le retrospective sono una parte fondamentale di questo processo.

3.4. Gli artefatti di Scrum

Il vocabolo artefatto è spesso associato a rovine archeologiche e antiche reliquie. Tuttavia, nello sviluppo del software, il termine si riferisce alle informazioni chiave necessarie durante lo sviluppo di un prodotto.

Gli artefatti di Scrum sono informazioni che lo scrum team e gli stakeholder utilizzano per definire in dettaglio il prodotto in fase di sviluppo, le azioni per produrlo e le azioni eseguite durante il progetto.

Questi artefatti forniscono parametri che permettono di capire l'andamento di uno sprint. Sono strumenti essenziali per ogni team scrum, in quanto permettono di conseguire gli attributi fondamentali di scrum: trasparenza, ispezione e adattamento.

SCRUM ARTIFACTS



Figura 12: Gli artefatti di Scrum

Fonte: atlassian.com/agile/scrum/artifacts

Come mostrato graficamente nella Figura 12 i principali artefatti di Scrum sono product backlog, sprint backlog e gli incrementi.

Product Backlog

Il product backlog è un elenco di nuove funzionalità, miglioramenti, correzioni di bug, attività o requisiti di lavoro necessari per costruire un prodotto. Viene compilato da fonti di

input come il servizio clienti, l'analisi della concorrenza, le richieste del mercato e l'analisi generale del business.

Il product backlog è un artefatto "vivo", in quanto viene aggiornato su richiesta quando sono disponibili nuove informazioni. È un registro trasversale al team che viene mantenuto e curato dal Product Owner durante tutta la durata del progetto. Contiene compiti che un tempo facevano parte di uno sprint attivo, ma che sono stati privati delle priorità e spostati nel backlog.

Sprint Backlog

Lo sprint backlog è un insieme di attività del product backlog che sono state promosse per essere sviluppate durante il prossimo sprint. Gli sprint backlog vengono creati dai team di sviluppo per pianificare le consegne per gli incrementi futuri e per definire nel dettaglio il lavoro necessario per creare l'incremento.

Gli sprint backlog vengono creati selezionando un'attività dal backlog di prodotto e suddividendola in elementi di sprint più piccoli e realizzabili. Consideriamo un'attività come "costruire la pagina del carrello di una piattaforma e-commerce", che richiede molte sotto-attività di progettazione e sviluppo. Il product backlog ospita l'attività principale, mentre le attività di supporto, come "creare un modello di design visivo del carrello" o "sviluppare le sessioni del carrello", sono ospitate nello sprint backlog.

Lo sprint backlog viene aggiornato durante la fase di pianificazione dello sprint di Scrum. I compiti più piccoli dello sprint vengono assegnati ai team interessati, come quelli di progettazione e sviluppo. Se un team non è in grado di consegnare tutti i compiti dello sprint, i compiti rimanenti resteranno in attesa nello sprint backlog per uno sprint successivo.

Incremento di Prodotto

Un incremento di prodotto è costituito dalla consegna del prodotto al cliente dopo aver completato le attività del product backlog durante uno sprint. Include anche gli incrementi di tutti gli sprint precedenti. C'è sempre un incremento per ogni sprint, il quale viene stabilito durante la fase di pianificazione. Un incremento ha luogo se il team decide di rilasciare il prodotto al cliente. Gli incrementi di prodotto sono incredibilmente utili e complementari per il monitoraggio delle versioni e, se necessario, per il loro ripristino.

3.4.1. Estensione di artefatti

Oltre agli artefatti ufficiali di Scrum precedentemente discussi, esistono alcuni artefatti aggiuntivi o meta-artefatti. Anche se non sono riconosciuti come ufficiali dalle linee guida di Scrum, questi artefatti aggiuntivi aumentano il valore e la comprensione di un ciclo di Scrum.

Burndown chart

Un grafico di burndown (o burnup) dello sprint non è un artefatto ufficiale di Scrum, ma molti team lo usano per comunicare e tracciare i progressi fatti verso l'obiettivo durante lo sprint. I grafici di burndown (vedi



Figura 13) sono grafici che visualizzano le attività completate durante la durata di uno sprint. Sono molto utili per aiutare a misurare la velocità di esecuzione attiva di un team, in modo da sapere se completerà ciò che è stato pianificato o se ha bisogno di ridefinire le priorità delle attività dello sprint.

Durante la pianificazione dello sprint, i team possono esaminare i grafici di burndown precedenti per farsi un'idea di quante attività possono realisticamente completare nello sprint successivo. I team possono esaminare i grafici di burndown in corso per identificare se sono in linea con l'obiettivo di completare con successo lo sprint. Durante la revisione dello sprint,

i team possono rivedere il grafico di burndown per individuare i punti in cui hanno raggiunto o mancato le aspettative.



Figura 13: Grafico Burndown

La definizione di "Fatto"

È importante che i team abbiano una chiara definizione di "Fatto". Questa definizione può essere un altro tipo di artefatto, che deve essere documentato e condiviso. Quando un elemento del product backlog o un Incremento è considerato "Fatto", tutti devono capire cosa si intende. Sebbene il significato di "Fatto" possa variare in maniera significativa da Scrum Team a Scrum Team, i membri devono avere una comprensione condivisa di ciò che si intende per lavoro completo, al fine di garantire la trasparenza. Un team senza una chiara definizione di "Fatto" si troverà spesso a chiedere, durante la revisione dello sprint, "questo è fatto?" quando si esaminano le attività di scrum aperte.

La stessa definizione guida il Team di Sviluppo a capire quanti elementi del Product Backlog è possibile selezionare durante lo Sprint Planning. Lo scopo di ogni Sprint è di consegnare incrementi di funzionalità potenzialmente rilasciabili che soddisfano la definizione attuale di "Fatto" dello Scrum Team.

La definizione di "Fatto" aiuta a definire i confini di un incremento. Gli incrementi devono essere consegnati in pacchetti completi e utilizzabili, che siano sommabili agli incrementi precedenti. Il termine fatto definisce anche quando i compiti sono completi e possono essere chiusi per il monitoraggio del burndown.

3.4.2. Trasparenza degli artefatti

Scrum è fondato sulla trasparenza. Le decisioni per ottimizzare il valore e controllare il rischio sono prese in base allo stato percepito degli artefatti. Nella misura in cui la trasparenza è completa, tali decisioni hanno una base solida. Nella misura in cui gli artefatti non siano completamente trasparenti, tali decisioni possono essere imperfette, il valore può diminuire e il rischio può aumentare.

Lo Scrum Master deve lavorare con il Product Owner, con il Team di sviluppo e con le altre parti coinvolte per capire se gli artefatti siano completamente trasparenti. Ci sono pratiche per gestire situazioni in cui la trasparenza non sia completa; in tali situazioni lo Scrum Master deve aiutare tutti ad applicare le pratiche più appropriate. Uno Scrum Master può rilevare che la trasparenza sia incompleta ispezionando gli artefatti, percependo i pattern, ascoltando attentamente ciò che viene detto e rilevando le differenze tra i risultati attesi e quelli reali.

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 l'apprendimento, la persuasione ed il cambiamento. La trasparenza non si ottiene da un giorno all'altro, ma piuttosto attraverso un percorso.

4. Gestione di un progetto con Scrum

Un progetto gestito con la metodologia Scrum si compone di cinque fasi (avvio, pianificazione, realizzazione, revisione e validazione, rilascio).

La fase di avvio si compone di 6 processi ed è particolarmente importante in quanto riguarda l'impostazione ed organizzazione complessiva del flusso di un progetto. È importante notare che i processi non sono necessariamente eseguiti in sequenza oppure separatamente. A volte, può essere più appropriato combinare alcuni processi, a seconda delle esigenze specifiche di ciascun progetto.

Nel seguito vengono pertanto descritti i processi da svolgere per avviare un progetto con la metodologia Scrum.

Costruire la Visione del Progetto

Una buona visione del progetto dovrebbe concentrarsi sui problemi che occorre affrontare piuttosto che sulla soluzione. La descrizione della visione del progetto non dovrebbe essere troppo specifica e dovrebbe lasciare spazio alla flessibilità. Il Product Owner è la persona responsabile del raggiungimento del massimo valore aziendale per il progetto e rappresenta anche il punto di vista della committenza nel processo di creazione della visione del progetto.

Identificare lo Scrum Master e gli Stakeholders

Lo Scrum Master e gli Stakeholder sono identificati utilizzando specifici criteri di selezione. Uno Scrum Master è un facilitatore che garantisce che lo Scrum Team lavori in un ambiente favorevole e comprenda bene gli obiettivi del lavoro e le modalità per completarlo. Le parti interessate (stakeholder come i clienti, utenti, sponsor e soggetti a vario titolo impattati dal progetto) spesso si interfacciano con lo Scrum Core Team e influenzano il progetto durante tutto il processo di implementazione dei deliverables previsti.

Costruire lo Scrum Team

Normalmente, il Product Owner ha la responsabilità primaria di selezionare i membri del team, ma spesso lo fa in collaborazione con lo Scrum Master. Lo Scrum Team è un gruppo di persone responsabili della comprensione dei requisiti aziendali specificati dal Product Owner, della stima delle User Story e della creazione dei deliverable del progetto. Il team decide la quantità di lavoro da impegnare in uno Sprint e determina il modo migliore per eseguire il lavoro.

Sviluppare le Epics

La Descrizione della Visione del Progetto funge da base per lo sviluppo delle Epics (ampie e non ancora raffinate User Stories nel Prioritized Product Backlog). Si possono tenere riunioni di gruppi di utenti per discutere le Epics appropriate. Le Epics vengono scritte nelle fasi iniziali del progetto, quando la maggior parte delle User Stories sono funzionalità di alto livello e le descrizioni dei prodotti e i requisiti sono definiti in modo ancora ampio. Una volta che queste Epics vengono inserite nel Product Backlog per il completamento in uno Sprint, vengono suddivise in User Stories più piccole e più granulari. Queste User Stories

più piccole sono generalmente semplici, brevi e facili da implementare per funzionalità o blocchi di attività che possono essere completati in uno Sprint.

Creare il Product Backlog Prioritizzato

Le Epics vengono ulteriormente dettagliate, elaborate e successivamente viene assegnata loro una priorità per creare un Product Backlog Prioritizzato. Questa attività viene svolta sotto il coordinamento del Product Owner. Il Product Backlog si basa su tre fattori primari: valore, rischio o incertezza e dipendenze. In questo processo vengono stabiliti anche i criteri ed i requisiti che devono essere soddisfatti come un insieme di regole applicabili a tutte le storie utente. Ciò al fine di rimuovere le possibili ambiguità dai requisiti e aiuta il team a rispettare gli standard di qualità. Una User Story è considerata completa e dettagliata quando è approvata dal Product Owner che la giudica sulla base dei criteri precedentemente definiti.

Creare un piano di rilasci

Lo Scrum Core Team esamina le User Stories presenti nel Product Backlog Prioritizzato per poi sviluppare un piano dei rilasci, che è essenzialmente un programma di implementazione a fasi che può essere condiviso con gli stakeholder del progetto. Il Product Owner e lo Scrum Team decidono la durata dello Sprint per il progetto, che spesso rimane uguale per tutto il progetto. Tuttavia, potrebbe cambiare se il Product Owner e il Team Scrum decidono che ciò è necessario o appropriato.

4.1. Lo Sprint

Una volta che il team ha sviluppato il backlog del prodotto e ha stabilito un piano di rilascio, può iniziare a lavorare in sprint. Come è possibile comprendere dalla Figura 14, lo Sprint è il cuore di Scrum, un periodo limite di un mese o meno durante il quale viene creato un Incremento di prodotto potenzialmente rilasciabile, utilizzabile e “Fatto”. Gli Sprint hanno una durata costante durante il lavoro di sviluppo e un nuovo Sprint si avvia immediatamente dopo la conclusione dello Sprint precedente.

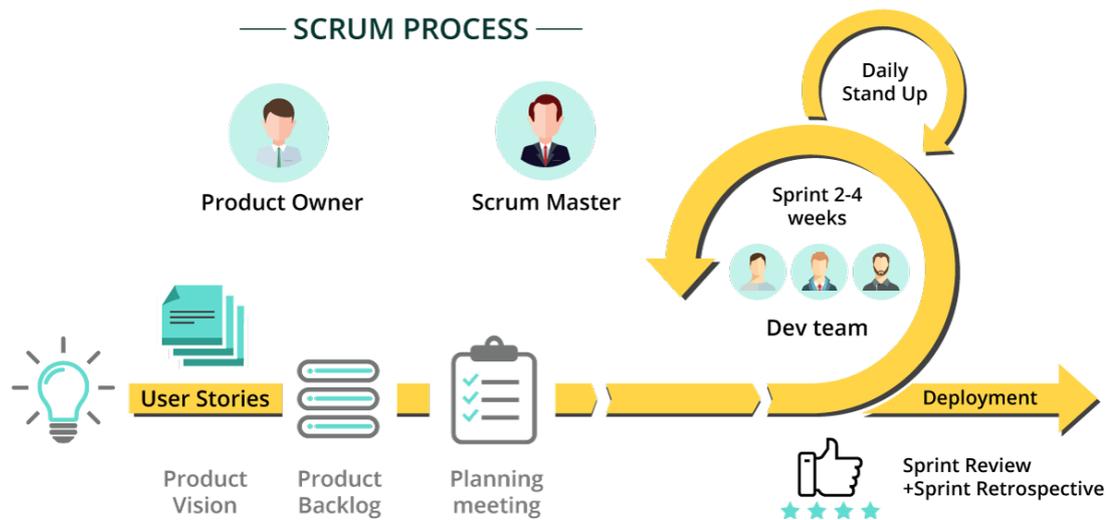


Figura 14: Processo Scrum

Gli Sprint contengono e sono costituiti da: lo Sprint Planning, il Daily Scrum, il lavoro di sviluppo, la Sprint Review e la Sprint Retrospective.

Durante lo Sprint:

- Non possono essere fatte modifiche che mettono a rischio lo Sprint Goal;
- Gli obiettivi relativi alla qualità non devono degradarsi;
- La visibilità può essere chiarita e rinegoziata tra il Product Owner e il Development Team, quando si è appreso di più.

Ogni Sprint può essere considerato un progetto con un orizzonte temporale non più lungo di un mese e ha un obiettivo di ciò che si va a costruire, un progetto e un piano flessibile che guideranno la costruzione, il lavoro svolto e l'Incremento risultante.

Gli Sprint sono limitati ad un mese di calendario. Quando uno Sprint ha un orizzonte temporale troppo lungo la definizione di ciò che viene costruito può cambiare, la complessità può crescere ed il rischio può aumentare. Gli Sprint rendono possibile la prevedibilità assicurando, almeno una volta al mese, l'ispezione e l'adattamento del progresso verso uno Sprint Goal. Gli Sprint inoltre limitano il rischio ad un mese di costo.

4.1.1. Sprint Planning

Il lavoro da eseguire nello Sprint è pianificato durante lo Sprint Planning. Tale pianificazione è creata grazie alla collaborazione dell'intero Scrum Team. Lo Sprint Planning è limitato temporalmente ad un massimo di sei ore per uno Sprint di un mese. Per Sprint più brevi,

l'evento è di solito più breve. Lo Scrum Master si assicura che l'evento abbia luogo e che i partecipanti ne comprendano la finalità. Lo Scrum Master insegna allo Scrum Team come svolgerlo nel tempo limite.

Lo Sprint Planning risponde alle seguenti domande:

- Qual è lo Sprint Goal?

Lo Sprint Goal è un obiettivo stabilito per lo Sprint che può essere raggiunto attraverso l'implementazione del Product Backlog. Esso fornisce una guida al Team di Sviluppo sul perché stia costruendo l'Incremento. Viene creato durante lo Sprint Planning e offre al Team di Sviluppo un certo grado di flessibilità.

Gli elementi del Product Backlog selezionati consegnano una funzionalità coerente, tale funzionalità è lo Sprint Goal che rimarrà fisso in mente al Team di Sviluppo durante tutto lo Sprint.

- Cosa può essere consegnato nell'Incremento risultante dallo Sprint imminente?

Il Team di Sviluppo lavora per prevedere le funzionalità che saranno sviluppate durante lo

Sprint. 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. L'intero Scrum Team collabora per comprendere il lavoro dello Sprint.

Gli input per questo incontro sono il Product Backlog, l'ultimo Incremento del prodotto, la

capacità prevista del Team di Sviluppo durante lo Sprint e le prestazioni registrate in passato del Team di Sviluppo. Il numero di elementi selezionati dal Product Backlog per lo Sprint è definito esclusivamente dal Team di Sviluppo che è il solo in grado di valutare cosa può compiere durante il prossimo Sprint.

- Come sarà messo in opera il lavoro necessario a consegnare l'Incremento?

Dopo aver creato lo Sprint Goal e selezionato gli elementi del Product Backlog per lo Sprint, il Team di Sviluppo decide come costruirà, durante lo Sprint, queste 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.

Il Team di Sviluppo di solito inizia con la progettazione del sistema e del lavoro necessario per convertire il Product Backlog in un Incremento funzionante del prodotto. Il lavoro può essere di varia dimensione o impegno stimato. Tuttavia, una quantità sufficiente di lavoro è pianificata durante lo Sprint Planning così da permettere al Team di Sviluppo di prevedere ciò che ritiene di poter fare nel prossimo Sprint. Alla fine di questo meeting, il lavoro che il Team di Sviluppo ha pianificato per i primi giorni dello Sprint è spesso suddiviso in unità della durata di un giorno o meno (user stories). Il Team di Sviluppo si autogestisce per intraprendere il lavoro contenuto nello Sprint Backlog.

Il Product Owner può aiutare a chiarire gli elementi selezionati dal Product Backlog e fare dei compromessi. 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 Team di Sviluppo può anche invitare altre persone a partecipare al meeting affinché forniscano una consulenza tecnica o di dominio.

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 lavorare in quanto team auto-organizzato, al fine di raggiungere l'Obiettivo di Sprint e creare l'Incremento previsto.

4.1.2. Daily Scrum

Il Daily Scrum è un evento limitato temporalmente a 15 minuti che serve al Team di Sviluppo per pianificare il lavoro per le prossime 24 ore. Ciò ottimizza la collaborazione e la prestazione ispezionando il lavoro svolto dall'ultimo Daily Scrum e prevedendo il lavoro in arrivo per i prossimi giorni dello Sprint. Il Daily Scrum si svolge ogni giorno allo stesso orario e nello stesso luogo per ridurre la complessità.

Il Daily Scrum ottimizza la probabilità che il Team di Sviluppo raggiunga lo Sprint Goal. Ogni giorno il Team di Sviluppo dovrebbe capire 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. Qui c'è un esempio di quali domande si potrebbero porre:

- Cosa ho fatto ieri che ha aiutato il Team di Sviluppo a procedere verso lo Sprint Goal?
- Cosa farò oggi per aiutare il Team di Sviluppo a procedere verso lo Sprint Goal?
- Vedo degli ostacoli che impediscono me o il Team di Sviluppo di procedere verso lo Sprint Goal?

Il Team di Sviluppo e le altre persone s'incontrano spesso subito dopo il Daily Scrum per discutere i dettagli, adattare o ri-pianificare il resto del lavoro dello Sprint.

Lo Scrum Master assicura che il Team di Sviluppo tenga la riunione ma il Team di Sviluppo è responsabile della conduzione del Daily Scrum. Lo Scrum Master insegna al Team di Sviluppo a mantenere il Daily Scrum entro il tempo limite di 15 minuti.

Il Daily Scrum è un meeting interno al Team di Sviluppo. lo Scrum Master si assicura, nel caso siano presenti altre persone, che queste ultime non disturbino il meeting.

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.

4.1.3. Sprint Review

Alla fine dello Sprint si tiene lo Sprint Review per ispezionare l'Incremento e adattare, se necessario, il Product Backlog. Durante lo Sprint Review lo Scrum Team e gli stackeolders discutono ciò che è stato fatto durante lo Sprint. A partire da questo e dai cambiamenti apportati al Product Backlog durante lo Sprint, i partecipanti collaborano a ciò che potrebbero essere fatto per ottimizzare il valore. Si tratta di un incontro informale e la presentazione dell'Incremento ha lo scopo di suscitare commenti e promuovere la collaborazione. È un incontro della durata massima di quattro ore per uno Sprint di un mese. Lo Scrum Master si assicura che l'evento abbia luogo e che i partecipanti ne comprendano la finalità. Lo Scrum Master insegna, a tutte le persone coinvolte, come mantenerlo nel tempo limite.

La Sprint Review include i seguenti elementi:

- I partecipanti sono lo Scrum Team e i principali portatori di interesse invitati dal Product Owner;
- Il Product Owner spiega quali elementi del Product Backlog sono stati "Fatti" e quali non sono stati "Fatti";
- Il Team di Sviluppo discute su cosa è andato bene durante lo Sprint, quali problemi si sono incontrati e come questi problemi sono stati risolti;
- Il Team di Sviluppo mostra il lavoro che ha "Fatto" e risponde alle domande relative all'Incremento;
- Il Product Owner discute lo stato attuale del Product Backlog. Fa una previsione delle possibili date obiettivo e di consegna in base alla misura dell'avanzamento attuale (se necessario);
- L'intero gruppo collabora su cosa fare dopo, così la Sprint Review fornisce un prezioso contributo ai successivi Sprint Planning.
- Passare in rassegna come il mercato o il potenziale utilizzo del prodotto possa avere cambiato la direzione del prossimo Incremento;
- Passare in rassegna la timeline, il budget, le funzionalità potenziali e il mercato per i prossimi previsti rilasci di funzionalità o di capacità del prodotto.

Il risultato dello Sprint Review è un Product Backlog revisionato che definisce gli elementi del Product Backlog probabilmente selezionati per il prossimo Sprint. Il Product Backlog può anche essere adattato globalmente per venire incontro a nuove opportunità.

4.1.4. Sprint Retrospective

La Sprint Retrospective è un'occasione per lo Scrum Team per ispezionare se stesso e creare un piano di miglioramento da attuare durante il prossimo Sprint. Si tiene dopo la Sprint Review e prima del successivo Sprint Planning. La Sprint Retrospective è una riunione della durata massima di tre ore per uno Sprint di un mese. Lo Scrum Master si assicura che l'evento abbia luogo e che i partecipanti ne comprendano la finalità, si assicura che il meeting sia positivo e produttivo e insegna come mantenerlo nel tempo limite (time-box).

Lo Scrum Master partecipa al meeting come elemento paritetico del team in relazione alla sua responsabilità sul processo di Scrum.

Lo scopo della Sprint Retrospective è di:

- 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.

Lo Scrum Master incoraggia lo Scrum Team a migliorare, all'interno del framework di processo Scrum, il proprio processo di sviluppo e le pratiche per renderlo più efficace e gradevole per il prossimo Sprint. Durante ogni Sprint Retrospective, lo Scrum Team pianifica i modi per aumentare la qualità del prodotto migliorando i processi di lavoro o adattando la definizione di "Fatto", se opportuno e non in conflitto con gli standard di prodotto o organizzativi.

Entro la fine della Sprint Retrospective, lo Scrum Team dovrebbe aver individuato i miglioramenti che implementerà nel prossimo Sprint. Attuare tali miglioramenti durante il prossimo Sprint è l'adattamento all'ispezione dello Scrum Team stesso. Anche se i miglioramenti possono essere implementati in ogni momento, la Sprint Retrospective fornisce una opportunità formale per focalizzare l'ispezione e l'adattamento.

4.1.5. Cancellare uno Sprint

Uno Sprint può essere cancellato prima che finisca il suo tempo limite. Solo il Product Owner ha l'autorità di annullare lo Sprint, sebbene possa farlo sotto l'influenza degli stakeholders, del Team di Sviluppo o dello Scrum Master.

Uno Sprint dovrebbe essere cancellato nel caso in cui lo Sprint Goal diventi obsoleto. Questo potrebbe verificarsi se l'organizzazione cambia direzione o se le condizioni di mercato o della tecnologia mutano.

Quando uno Sprint è annullato, ogni elemento del Product Backlog già completato e "Fatto" è esaminato. Se parte del lavoro è potenzialmente rilasciabile il Product Owner tipicamente la accetta. Tutti gli elementi incompleti del Product Backlog sono nuovamente stimati e reinseriti nel Product Backlog. Il lavoro svolto su di esse si deprezza rapidamente e deve essere frequentemente ristimato.

Le cancellazioni degli Sprint sottraggono risorse, poiché si rischia di sprecare il lavoro effettuato e di dover riprogrammare un nuovo Sprint partendo da zero.

4.1.6. Tenere traccia dello stato di avanzamento del rilascio

Con l'avanzamento del progetto sprint dopo sprint, il team tiene traccia dello stato di avanzamento complessivo verso il rilascio successivo. Il team terrà traccia dello stato di avanzamento anche per stimare e migliorare la velocità.

Man mano che il team tiene traccia dello stato di avanzamento deve tentare di rispondere a domande del seguente tipo:

- Stiamo lavorando sulle storie utente più appropriate? Il backlog del prodotto viene ridefinito con nuove storie utente con l'avanzamento del progetto. Tuttavia, se il numero totale di storie nel backlog non diminuisce, sebbene vengano completate storie a ogni sprint, il team deve esaminare la causa. Le storie completate potrebbero non rappresentare le scelte migliori. Il team deve prefissarsi una visione e un obiettivo per ogni rilascio e deve assicurarsi che le storie siano collegate direttamente a ciò che viene richiesto dal cliente.
- Siamo in presenza di un debito tecnico? Alcuni team considerano una storia utente finita anche se rimane ancora del lavoro da completare come, ad esempio, correggere alcuni bug. Tali team si assumono un debito tecnico che devono pagare in un secondo momento, generalmente a un costo maggiore.

4.1.7. Concludere il rilascio

Se il team non accumula alcun debito tecnico, può rilasciare il prodotto non appena ha completato gli sprint del rilascio, senza alcun lavoro aggiuntivo. Il team e il Product Owner tengono riunioni retrospettive per la revisione del cliente al fine di esaminare il rilascio nel suo complesso. Tuttavia, il debito tecnico è un problema difficile che i team non possono risolvere facilmente. Se il proprio team, come molti altri, continua ad accumulare debito tecnico, sarà necessario dedicare tempo allo svolgimento di tutto il lavoro rimanente per finire le storie utente prima di rilasciare il prodotto. Nella retrospettiva per il rilascio, considerare le azioni che il team deve eseguire negli sprint successivi al fine di evitare di assumere altro debito.

4.2. Strumenti di gestione per progetti Agile

Agile è il metodo più diffuso e utilizzato dai team di sviluppo software di tutto il mondo. Se questi team dovessero definirlo, probabilmente lo farebbero in termini di strumenti utilizzati e di diverse modalità di implementazione, come le dashboard kanban, gli standup giornalieri e le riunioni scrum. Sebbene sia vero che questi strumenti e tecniche sono utilizzati in agile, nessuno di essi può definire la metodologia nel suo complesso.

I team di sviluppo software hanno bisogno di strumenti agili per conseguire gli obiettivi dell'Agile; sono obbligati a dare spazio al feedback regolare dei clienti e degli utenti finali attraverso test abituali e iterazioni illimitate. Invece di fissare tutto in una rigida tabella di marcia, il lavoro viene pianificato in sprint.

In base alle esigenze di ogni organizzazione e team, la metodologia Agile può essere ulteriormente distinta in metodologie kanban e scrum.

Secondo lo State of Agile Report del 2019, il 72% degli intervistati ha dichiarato di utilizzare la metodologia scrum o almeno un ibrido che la include.

Un ambiente di lavoro agile ha un ritmo veloce, come suggerisce il nome. Per evitare che i progetti cadano nel caos, è necessario un solido strumento di gestione dei progetti agili. È il pilastro che sostiene un team agile.

Cosa cercare in uno strumento agile? Nella ricerca di quest'ultimo, bisognerebbe prendere in considerazione i seguenti aspetti:

- Che dimensioni ha il team?

- In che settore opera l'azienda?
- Su quali tipi di progetti e cliente lavora l'organizzazione?
- Che tipo di dati occorrono regolarmente? Quali tipi di report?
- Qual è il livello di adozione agile del team?
- Qual è il budget?

Ci sono comunque degli aspetti più importanti che bisogna cercare in ogni strumento di gestione agile dei progetti:

- Buona gestione del backlog
- Pianificazione e monitoraggio del progetto
- Roadmapping
- Pianificazione del rilascio
- Buone funzionalità di collaborazione

4.2.1. Atlassian Jira

Atlassian JIRA è probabilmente uno dei più famosi strumenti di gestione agile dei progetti.

Atlassian JIRA si rivolge soprattutto a professionisti IT, designer UI-UX e team di sviluppo. È dotato di schede kanban scrum completamente personalizzabili che si possono facilmente adattare al flusso di lavoro generale del team. Queste schede possono essere utilizzate in modo semplice per visualizzare tutte le attività di uno sprint.

Grafici di burnout, diagrammi di flusso cumulativi, diagrammi di velocità, rapporti sulle epiche e rapporti di sprint aiutano a fornire un quadro in tempo reale della situazione del team in un determinato sprint, oltre a un semplice sistema di gestione del backlog (vedi Figura 15). È anche possibile gestire filtri personalizzati utilizzando il linguaggio di interrogazione JIRA (JQL).

Inoltre, è possibile integrarlo con gli strumenti per gli sviluppatori e ci sono più di 1.000 add on diversi che si possono utilizzare in base alle esigenze del team.

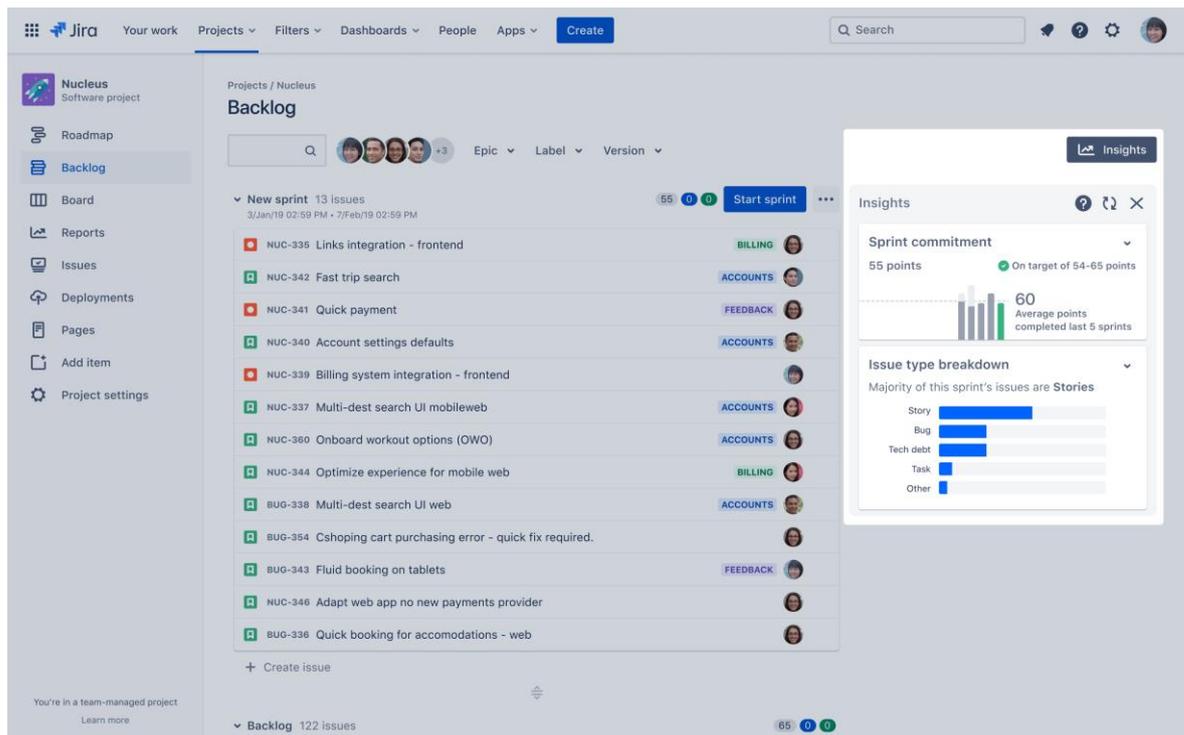


Figura 15: Esempio finestra Backlog Jira

Fonte: <https://www.justinmind.com/blog/agile-tools/>

4.2.2. Pivotal Tracker

Pivotal Tracker è uno strumento di gestione agile dei progetti di Pivotal Labs, un'azienda che fornisce molti strumenti software diversi per aiutare le aziende moderne.

Una delle principali caratteristiche di questo strumento agile è che utilizza le storie utente per indicare i diversi compiti, fornendo anche una pianificazione automatica e una mappatura delle storie utente. Inoltre, i membri del team possono classificare ogni attività in base alla sua complessità utilizzando i punti scrum (Vedi Figura 16). Pivotal Tracker conta quante attività e quanti punti vengono completati dal team ogni giorno.

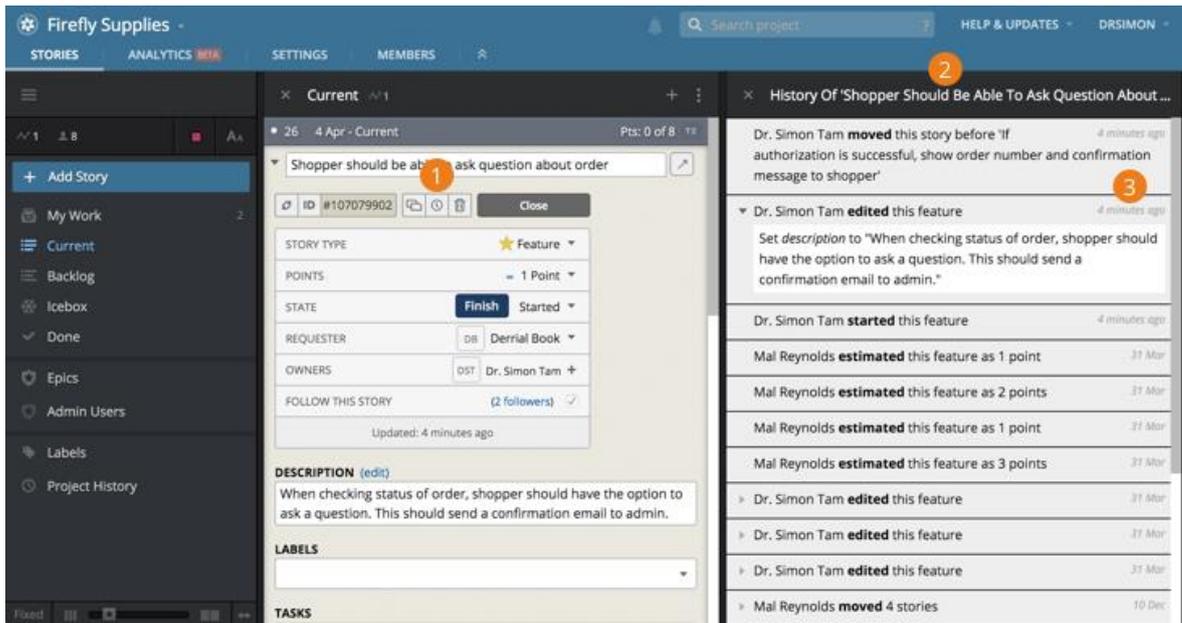


Figura 16: Esempio finestra user story Pivotal Tracker Fonte: <https://www.justinmind.com/blog/agile-tools/>

4.2.3. Nutcache

Nutcache si presenta come lo strumento agile "più affidabile al mondo". Consente al team il controllo di tutto, dal budget di progetto al monitoraggio del tempo, alla gestione delle spese e persino alla fatturazione dei prodotti. È ideale per tutti, dalle ditte individuali alle grandi aziende.

Nutcache ha anche una funzione che permette di monitorare l'avanzamento degli sprint e la redditività del prodotto utilizzando il framework Scrum e la dashboard del progetto (Vedi Figura 17). Ci sono altre caratteristiche che lo rendono più esclusivo, come la possibilità di codificare a colori le attività e i progetti.

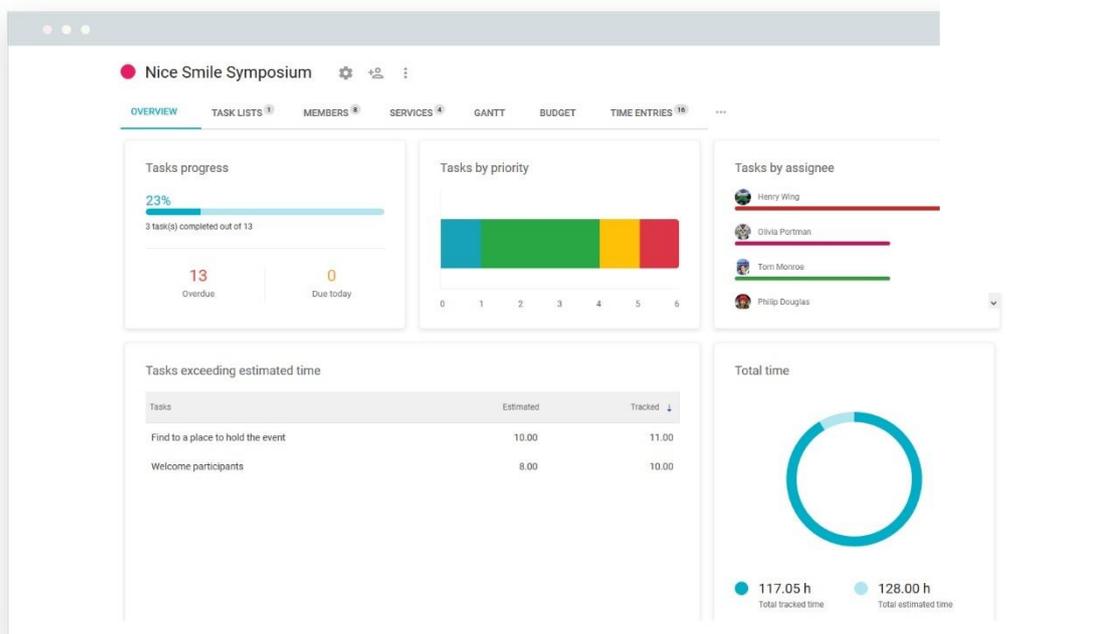


Figura 17: Dashboard Nutcache

Fonte: <https://www.justinmind.com/blog/agile-tools/>

4.2.4. Scrumwise

Scrumwise è uno strumento di gestione agile dei progetti che si concentra specificamente sui team che utilizzano l'approccio Scrum alla gestione agile.

L'obiettivo originario di Scrumwise era quello di creare lo strumento di scrum più amichevole sul mercato. Con questo strumento è possibile creare rapidamente elementi del backlog, categorizzarli con elenchi a scelta, applicare tag e utilizzare filtri in modo da potervi accedere rapidamente ogni volta che si, ha bisogno (Vedi Figura 18). Include anche grafici di burnup e burndown e funzionalità per i team, oltre al tracciamento nativo dei tempi.

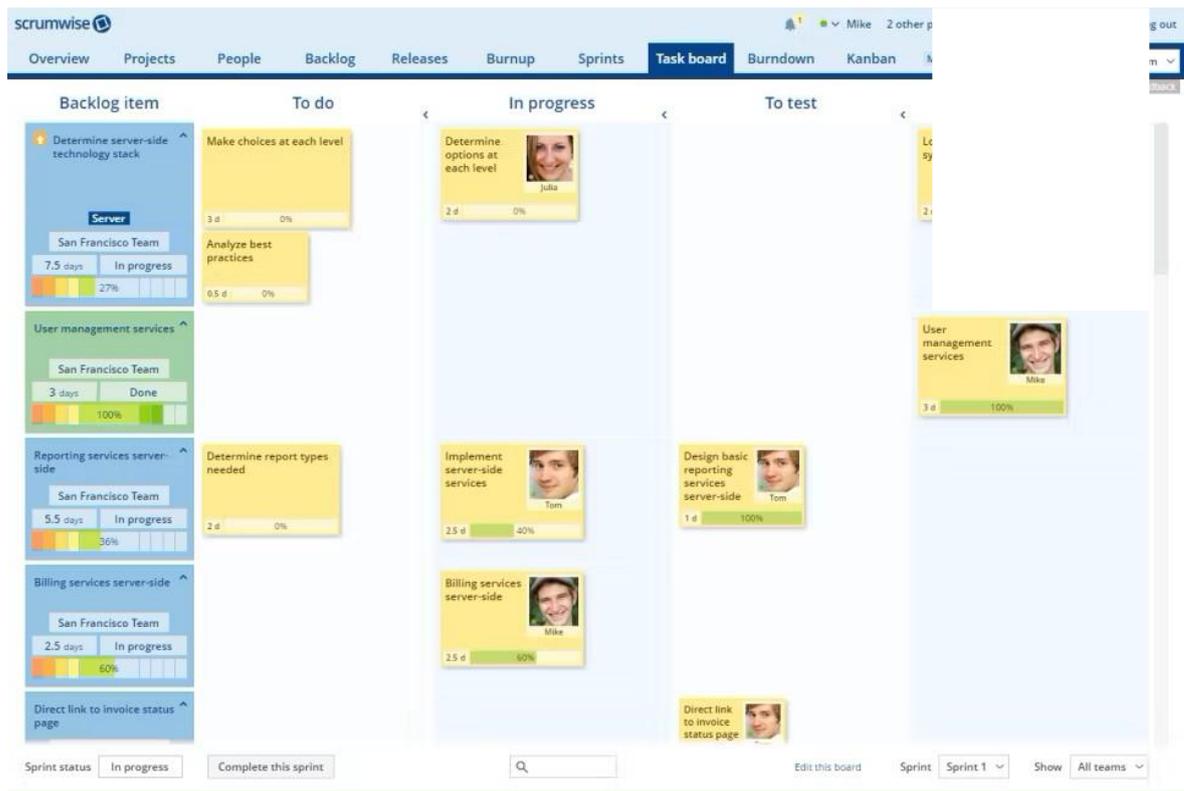


Figura 18: Dashboard Scrumwise

Fonte: <https://www.justinmind.com/blog/agile-tools/>

4.2.5. Miro

Miro (precedentemente noto come RealtimeBoard) si presenta come una lavagna digitale che facilita la collaborazione tra gli utenti. Il software consente di creare note e disegni, di spostare gli oggetti e di comunicare attraverso videochiamate o chat online (Vedi Figura 19). Lo strumento è dotato anche di una serie di modelli precostituiti che possono ispirare o servire come punto di partenza per il proprio progetto di lavoro.

Alcuni usi comuni di Miro includono:

- Creare un elenco di idee o argomenti su cui concentrarsi per un progetto importante;
- Sviluppo di una storia dell'utente o di una mappa del percorso del cliente;
- Creazione di un wireframe per un nuovo concetto o idea di prodotto;
- Definizione di una roadmap o di una strategia per un workshop o un programma di innovazione;
- Gestire una riunione più coinvolgente o innovativa per il team in una cultura agile.

Un'altra caratteristica importante da sottolineare sono le integrazioni e le connessioni con le app. Miro può collegarsi ad altri strumenti di lavoro virtuale comuni, come Google Drive, Slack, Asana, Trello e SurveyMonkey.

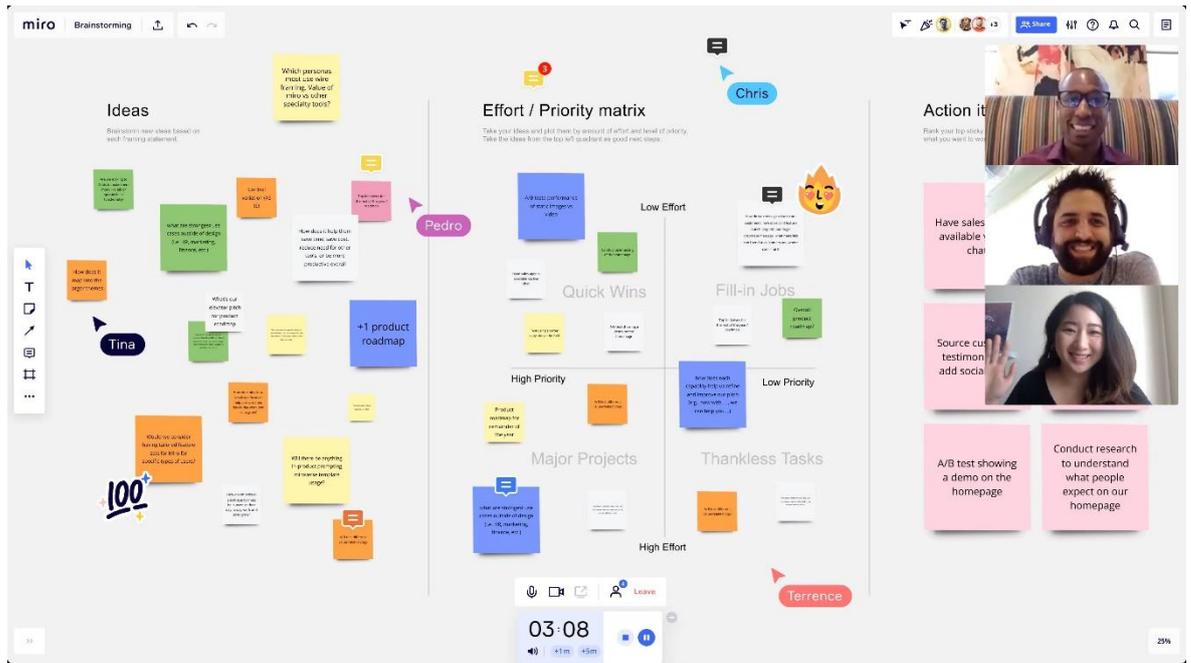


Figura 19: Miroboard

Fonte: <https://www.justinmind.com/blog/agile-tools/>

5. Casi studio – Progetti Accenture

5.1. Chi è Accenture

Accenture è un'azienda leader a livello globale nel settore dei servizi professionali, che fornisce una vasta gamma di servizi e soluzioni nei settori strategy, consulting, digital, technology e operations.

Combinando un'esperienza unica e competenze specialistiche in più di 40 settori industriali e in tutte le funzioni aziendali - sostenuta dalla più ampia rete di delivery center a livello mondiale - Accenture opera all'intersezione tra business e tecnologia per aiutare i clienti a migliorare le proprie performance e creare valore sostenibile per i loro stakeholder.

Con circa 435.000 professionisti impegnati a servire i suoi clienti in più di 120 paesi, favorisce l'innovazione per migliorare il modo in cui il mondo vive e lavora.

L'azienda può vantare un'ampia collaborazione con le più grandi aziende di tutti settori industriali e agenzie governative in tutto il mondo. In particolare, nell'ultimo anno annovera tra i suoi clienti 95 delle aziende Fortune Global 100 e oltre tre quarti delle Fortune Global 500. Tutti i principali clienti si affidano ad Accenture da almeno 5 anni, 98 sono clienti da 10 anni.

Al centro dell'azienda e del suo valore ci sono le persone e la loro continua crescita: per questo l'azienda sostiene la formazione e lo scorso anno fiscale a livello internazionale, ha investito 935 milioni di dollari in formazione.

I global delivery center integrano il lavoro svolto presso i clienti, attraverso un'ampia rete di centri tecnologici in tutto il mondo, a supporto costante dell'operatività e dell'efficienza dei servizi offerti. I Delivery Center sono più di 50 e offrono servizi in 39 lingue diverse, con un impiego di oltre 273.000 risorse qualificate.

5.1.1. Accenture in Italia

Presente in Italia dal 1957, Accenture conta oggi circa 13.000 dipendenti e negli ultimi quattro anni ha assunto con un ritmo medio di circa 1.500 persone, ogni anno. L'azienda ha sedi principali a Milano, Roma, Torino, Napoli e Cagliari oltre a diversi uffici sul territorio nazionale.

L'Italia è parte integrante del network internazionale di Accenture con:

- 5 centri di innovazione per lo sviluppo di soluzioni avanzate in settori specializzati quali: banda larga, telecomunicazioni, food, moda, automotive e impresa 4.0 (Accenture Automotive Industry Solution Center, a Torino; Accenture Customer Innovation Network, a Milano; Accenture Digital Acceleration Center e Accenture Digital Platform, a Roma; Industrial IoT Innovation Center Center a Modena).
- 2 Delivery Center a Napoli e Cagliari. Il centro di Napoli, aperto nel 2001 occupa oggi 1.300 persone e ha fatto da modello al centro di Cagliari che, inaugurato nel 2015 insieme ad Avanade, conta complessivamente 300 dipendenti. Entrambe queste realtà sono parte integrante della rete di oltre 50 centri di Delivery di Accenture presenti in tutto il mondo.

Negli ultimi anni Accenture ha continuato la propria strategia di consolidamento in Italia mediante l'acquisizione di New Energy Group (2016), azienda specializzata in soluzioni di Salesforce, e di i4C Analytics (2014), con l'obiettivo di consolidare il proprio ruolo nel campo digital e dei big data.

Accenture in Italia è attiva nelle principali associazioni di settore nel Paese e collabora con think thank e organizzazioni non profit.

5.2. Caso studio 1 – Come il testing Agile ha portato efficienza in Swisscom

5.2.1. La Sfida

La rapida commercializzazione del mercato delle telecomunicazioni in Europa esercita una forte pressione sui ricavi e sulle prospettive di crescita dei fornitori di servizi di comunicazione.

Per competere e prosperare in questo mercato spietato, i fornitori devono adattarsi sia a ciò che fanno, sia a come lo fanno, per una nuova era digital-first.

Nel 2016, Swisscom ha colto l'opportunità di passare a un'attività realmente digitale.

In qualità di fornitore leader in Svizzera di servizi di comunicazione, IT e intrattenimento, con una forza lavoro di oltre 19.000 persone, l'organizzazione aveva una ricca eredità di successi da cui partire. Avendo tra i suoi valori fondamentali l'innovazione e la sostenibilità, l'azienda ha deciso di digitalizzare sia l'attività che la funzione IT interna per semplificare le operazioni, aumentare la flessibilità e ridurre i costi IT.

Il testing del software è stato un componente centrale della trasformazione, con un'attenzione particolare ad Agile e DevSecOps. Per anni, Swisscom è stata dipendente da diversi fornitori di servizi e strumenti in licenza per questa parte critica del processo di sviluppo delle applicazioni. Tuttavia, l'azienda ha riconosciuto che la sua struttura tradizionale, che comprendeva un centro di eccellenza per il testing (TCoE) centralizzato con 300 persone, era diventata troppo rigida e doveva adottare metodi più agili. Inoltre, si stava rivelando sempre più costoso, con il 40% della spesa capex dedicata ai test, ben al di sopra dei parametri di riferimento del settore. Swisscom mirava ad acquisire una capacità di testing più agile che accelerasse il rilascio di software di alta qualità sul mercato, riducendo al contempo i costi.

5.2.2. Cosa ha fatto Accenture

Accenture ha lavorato fianco a fianco con Swisscom per una trasformazione quadriennale del suo approccio al testing delle applicazioni.

Per raggiungere il duplice obiettivo di una maggiore agilità organizzativa e di una riduzione dei costi complessivi, il team interfunzionale ha contribuito a incorporare i moderni principi di ingegneria della qualità e di continuous testing in tutta la pipeline di consegna del software.

Un'area chiave del programma è stata l'introduzione di un approccio moderno e agile "Shift Left". Ciò significava incoraggiare gli sviluppatori a testare il proprio codice molto prima nella pipeline di sviluppo. Tuttavia, affinché lo Shift Left funzioni nella pratica, gli sviluppatori dovevano essere in grado di vedere come funzionava il loro codice nel mondo reale. Accenture ha creato una funzionalità di virtualizzazione dei servizi in grado di simulare le prestazioni dell'applicazione in un ambiente di produzione complesso e interdipendente, aiutando gli sviluppatori a visualizzare il funzionamento dell'applicazione e l'interfaccia con gli altri componenti e consentendo loro di verificare l'impatto delle loro scelte di codifica molto prima e con maggiore efficacia.

Insieme a Swisscom, Accenture ha sviluppato un framework e strumenti di automazione dei test personalizzati, per aiutarla a cercare di contenere ingenti costi di licenza e a sviluppare capacità e standardizzazione interne. Ciò ha incluso l'approvvigionamento di un nuovo strumento per la gestione dei dati di test e la creazione di un ambiente di test più stabile con strumenti di sistema automatizzati. Un'altra importante spinta del programma è stata

l'automazione estensiva, compresa l'automatizzazione di decine di migliaia di casi di test manuali per consentire test rapidi e agili su scala. Il framework di automazione modulare su misura non solo ha stabilito una copertura del 100% dei processi aziendali, ma ha anche permesso l'esecuzione di test paralleli e continui 24 ore su 24, 7 giorni su 7, con un intervento umano ridotto.

5.2.3 Persone e Cultura

Lavorando a stretto contatto e accelerando l'adozione dei processi Agile, Swisscom e Accenture hanno apportato un vero e proprio cambiamento nel modo in cui la società pensa allo sviluppo del software.

L'idea del test continuo e il mantra della "qualità incorporata" sono stati saldamente incorporati nella funzione IT.

Per il team di testing stesso, questo programma ha trasformato la vita lavorativa quotidiana. Anziché dover eseguire laboriosi test manuali, una maggiore automazione combinata con test più precoci ha dato ai membri del team la possibilità di pensare in modo più strategico, sviluppare le proprie competenze e creare nuovi strumenti.

Allo stesso tempo, gli sviluppatori di Swisscom hanno ora le competenze, gli strumenti e la mentalità necessari per assumersi la responsabilità end-to-end della qualità del loro lavoro di sviluppo, anziché affidarsi ai test alla fine del ciclo di vita dello sviluppo. Tutto ciò è stato completato da una formazione completa e dall'adozione integrale di Agile.

5.2.4. Valore consegnato

La trasformazione della funzione di test e qualità delle applicazioni di Swisscom è stata fenomenale. I risultati sono eloquenti.

La quota di capex dedicata ai test è scesa da circa il 40% all'inizio dell'impegno a solo l'8% oggi, un obiettivo chiave, mentre l'automazione estesa consente di condurre i test di routine a qualsiasi ora del giorno con un intervento umano ridotto.

L'approccio complessivo ai test è passato da un approccio reattivo a uno più proattivo e predittivo, con il monitoraggio automatizzato dei server e l'auto-guarigione dei problemi dell'infrastruttura. La virtualizzazione dei servizi ha consentito un vero e proprio cambiamento di mentalità verso il testing continuo e precoce, contribuendo a ridurre lo

slittamento dei difetti del 35%. Inoltre, la centralizzazione dei test è stata ridotta al minimo, con un team molto più snello che fornisce solo test e consulenza su richiesta.

Anche i clienti di Swisscom sono destinati a beneficiare del nuovo approccio di testing, poiché una maggiore agilità di testing comporta una maggiore velocità di commercializzazione. Quando l'azienda identifica una correzione o una nuova funzionalità per migliorare l'esperienza del cliente, ha a disposizione un'organizzazione IT in grado di svilupparla, testarla e immetterla sul mercato molto più rapidamente. In effetti, i rilasci di software che prima richiedevano fino a 12 settimane, ora possono essere effettuati entro due settimane, con una riduzione dell'80% del time to market.

5.3. Caso studio 2 – Sviluppo di un portale per l'e-commerce di Leonardo

5.3.1. Invito al cambiamento

Leonardo, leader mondiale nel settore aerospaziale e della difesa con sede in Italia, ha visto la possibilità di sviluppare una piattaforma digitale all'avanguardia per i suoi prodotti.

L'azienda, produttrice di elicotteri, aerei, elettronica per la difesa e sistemi di sicurezza informatica, voleva snellire i processi, ottimizzare l'esecuzione degli ordini e trasformare le interazioni con i clienti in esperienze uniche. Ciò richiedeva lo sviluppo di una piattaforma completa e l'ideazione di un'esperienza utente solida e impressionante per gestire gli ordini, i servizi post-vendita, le relazioni con i clienti e i servizi di formazione. L'ambizione di Leonardo era quella di avere un'unica piattaforma, la migliore del settore, in grado di soddisfare le esigenze dei clienti in tutta l'azienda. Il piano prevedeva di iniziare il percorso nella divisione elicotteri e successivamente di estenderlo a tutta l'azienda.

5.3.2. Quando la tecnologia incontra l'ingegno umano

Grazie alla profonda conoscenza delle best practice del settore aerospaziale e della difesa, del panorama competitivo e dei processi, Accenture era il copilota ideale per aiutare l'azienda a percorrere l'intero processo di progettazione-implementazione. Il rapporto di lunga data di Accenture con Leonardo e le loro collaborazioni con le aziende tecnologiche leader SAP e Microsoft hanno conferito a questo progetto di trasformazione una forte attenzione alla

collaborazione con l'ecosistema. Di conseguenza, è stato possibile implementare contemporaneamente più soluzioni e servizi tecnologici. Il team di esperti aziendali e IT di Leonardo e i professionisti funzionali e tecnici di Accenture hanno intrapreso una ricerca approfondita per determinare le esigenze degli stakeholder di Leonardo e quelle dei suoi clienti. Le intuizioni raccolte sono state fondamentali per definire la visione strategica e per progettare una piattaforma che combinasse due soluzioni:

- Un portale clienti e un modulo CRM costruiti su Microsoft Dynamics 365, che offrono al cliente un unico punto di accesso a tutti i servizi post-vendita di Leonardo e l'accesso a ticket di assistenza, supporto tecnico e dashboard interattivi, tutto in un unico luogo.
- Una piattaforma di e-commerce, basata su SAP Commerce Cloud, che consente ai clienti che navigano nel Leonardo Store di accedere facilmente anche al catalogo completo dei servizi post-vendita, all'acquisto di parti di ricambio e molto altro ancora.

L'approccio agile allo sviluppo ha accelerato la consegna di questa piattaforma orientata all'esperienza, ideata per le esigenze dei clienti del settore aerospaziale e della difesa.

5.3.3 Una differenza preziosa

La piattaforma di e-commerce è diventata operativa dopo tre mesi dall'inizio dell'implementazione.

Ulteriori funzionalità sono state implementate nel corso dell'anno successivo, adottando l'approccio agile. Il nuovo portale per i clienti è stato realizzato pochi mesi dopo. Diverse tecnologie complesse lavorano ora simultaneamente dietro le quinte per offrire al cliente un'esperienza completamente compatta e apparentemente senza sforzo. Leonardo ha una visibilità in tempo reale su ordini e vendite, e i percorsi dei clienti possono ora essere tracciati dalla prima interazione all'acquisto. Il sistema supporta la collaborazione tra i reparti attraverso flussi di lavoro integrati, incrementando ulteriormente l'efficienza e riducendo i costi.

Dal lancio, i seguenti risultati dalla piattaforma a Leonardo:

- 1600 nuovi clienti acquisiti
- 11000 utenti attivi
- 5,000 accessi settimanali
- 2,500 pubblicazioni tecniche scaricate settimanalmente
- 700 richieste di informazioni da parte dei clienti create settimanalmente
- 1,000 ordini gestiti settimanalmente

I numeri riflettono il fenomenale percorso compiuto finora da Leonardo. I benefici sono evidenti in termini di nuovi servizi, efficienza, sostenibilità e riduzione dei costi interni. Sulla buona strada per raggiungere l'obiettivo finale di un'esperienza solida a livello aziendale che serva tutte le divisioni e i clienti dell'azienda, Leonardo dispone oggi di un unico punto di accesso per tutti i suoi servizi post-vendita e di un percorso cliente intuitivo e senza soluzione di continuità.

5.4. Caso studio 3 – Guidare una compagnia assicurativa Europea nella sua crescita

5.4.1. La sfida

Per anni, una compagnia assicurativa europea ha affidato ad Accenture la gestione delle vendite, dell'assistenza, dei sinistri e delle operazioni finanziarie e contabili. Quando il cliente ha intensificato la sua attenzione alla crescita del business, ha visto l'opportunità di portare i suoi servizi gestiti a un livello superiore. In particolare, passando a un modello operativo intelligente alimentato da dati, tecnologie intelligenti e talento, l'azienda riteneva di poter:

- Standardizzare i tassi di sovrapprezzo per i rinnovi delle polizze commerciali.
- Migliorare l'accuratezza dei rimborsi ai clienti: il 26% dei rimborsi non era corretto.
- Migliorare gli incassi dai coassicuratori, che dovrebbero rimborsare la compagnia per la loro parte di sinistri.
- Ridurre i debiti legati ai sinistri di vecchia data.

Queste iniziative non solo avrebbero migliorato l'esperienza dei clienti, ma avrebbero anche reso disponibile più risorse da reinvestire nel programma di crescita.

5.4.2. Cosa ha fatto Accenture

Prima di passare a un modello operativo intelligente, l'assicuratore ha ampliato il suo accordo di servizi gestiti con Accenture per includere le funzioni assicurative e finanziarie principali. Il team ha quindi analizzato e mappato il 100% dei processi end-to-end per identificare i colli di bottiglia e ridurre la frammentazione dei processi nelle linee di business personali e commerciali, nonché in tutte le funzioni di gestione dei sinistri e finanziarie. Allo stesso tempo, sono stati introdotti cambiamenti per migliorare la qualità del servizio e incrementare le prestazioni operative sfruttando l'automazione e l'analisi.

In particolare, questi sono gli obiettivi prefissati da Accenture:

Aumentare la reattività dei clienti

- Aumentare l'accesso dei clienti al canale di servizio Webchat dell'assicuratore collegandolo a pagine aggiuntive.
- Introdurre nuove funzionalità di self-service.
- Formare gli agenti su più prodotti.

Ridurre al minimo la perdita di ricavi da sovrapprezzi e rimborsi

- Identificare le polizze con supplementi errati e convertirle nella tariffa corretta.
- Applicare analisi e snellire i processi per identificare gli errori di rimborso e gli errori di sistema, riducendo così la perdita di ricavi.

Migliorare il recupero crediti

- Migliorare la riscossione dei crediti e chiudere sinistri per un valore di 10,2 milioni di dollari sfruttando le analisi.
- Ridurre i debiti di coassicurazione istituendo nuovi processi per assegnare priorità alle richieste di rimborso, migliorare il recupero dei crediti e risolvere i problemi.

5.4.3. Persone e cultura

Il gruppo assicurativo si impegna a creare una cultura incentrata sull'offerta di percorsi di carriera soddisfacenti. L'accento è posto sulla crescita lavorativa delle persone e sulla creazione di un ambiente collaborativo in cui tutti lavorano uniti per soddisfare al meglio i clienti. Il modello operativo intelligente supporta questo ambiente valorizzando le persone di talento, consente lo sviluppo di competenze e padronanza anche in altre aree, facilita la comunicazione sull'andamento dell'azienda con i leader, responsabili di assumere le decisioni migliori e più tempestive tali da garantire la sostenibilità del business.

5.4.4. Valore consegnato

La compagnia assicurativa ha fatto passi da gigante nella riduzione dei costi, nel miglioramento dei ricavi e dell'esperienza dei clienti. Grazie a un modello operativo intelligente, ha ottenuto:

- Un debito quasi dimezzato.
- Ridotto la perdita di ricavi di 78.000 dollari al mese.
- Miglioramento degli incassi a livelli record di 125 milioni di dollari, con un aumento dei profitti di 13 milioni di dollari.
- Riduzione del 63% dei debiti legati ai sinistri e del 55% dei debiti di coassicurazione.

- Raddoppiato il Net Promoter Score da +32% a +68%, ottenendo prestazioni ai massimi livelli nella soddisfazione dei clienti.
- Ha ridotto da 24 ore a un'ora i tempi di risposta via e-mail alle richieste di informazioni nel settore Personal Lines e ha ridotto di 21 giorni i tempi di recupero dei sinistri Motor.

Accenture sta ora lavorando con la compagnia per costruire una roadmap strategica che porterà a un'ulteriore innovazione e a una continua trasformazione, sfruttando SynOps, una piattaforma che orchestra e monitora la combinazione ottimale di talenti umani e macchine, grazie a intuizioni basate sui dati.

6. Casi studio – Non solo software: la metodologia Agile al di fuori del mondo IT e nella vita reale

Se è vero che Agile nasce come metodologia da adottare nello sviluppo software, è altrettanto vero che oggi viene applicato ai settori più disparati: dalla progettazione di macchine e aerei, alla reingegnerizzazione di processi, alla realizzazione di guide turistiche, alla costruzione di cartoni animati. In particolare, il framework che più si adatta ad essere applicato al di fuori di contesti di sviluppo software è Scrum, che viene utilizzato, per esempio, per produrre il cartone animato Southpark e declinato, con eduScrum, nel mondo scolastico per favorire il lavoro di gruppo degli studenti.

Ulteriori esempi che fanno comprendere bene la varietà di applicazione sono il prototipo di automobile progettata da un ingegnere usando principi e tecniche agile, le guide turistiche Lonely Planet, realizzate utilizzando il framework Scrum, il caccia Saab JAS 39 che ha avuto costi di progetto di 14 miliardi di dollari contro i 1500 miliardi dell’F-35 sviluppato secondo approcci tradizionali e molto altro ancora.

6.1. Caso studio 4 – Agile coaching per la scuola, una didattica a prova di futuro

Presso il liceo “Carlo Porta” di Monza si è tenuto un progetto extra-curriculare chiamato “Be Agile”, all’interno delle attività di orientamento in uscita degli studenti delle otto classi quinte. Un progetto che ha impegnato ciascuna classe quattro ore, durante le quali si è cercato di portare i ragazzi a una maggiore conoscenza di sé. Tra i più rilevanti “no technical skills”

richiesti nel mondo del lavoro odierno e di alcuni degli strumenti utilizzati nella metodologia Agile (la retrospettiva, il ciclo di Deming, il problem solving...).

Partendo con un bilancio delle competenze che potesse mettere in evidenza le capacità degli studenti, i loro interessi, le loro motivazioni e i loro valori, si sono toccate tematiche che vanno dall'uscita dalla zona di comfort fino alla retrospettiva.

Tra le diverse metodologie, quella Agile sembra rispecchiare tutte le caratteristiche di innovazione che la scuola di oggi necessita, con una forte componente umanistica, che diventa sempre più "conditio sine qua non" per una crescita completa delle nuove generazioni. Nell'Agile, infatti, ad essere messe in discussione non sono mai le persone ma i processi, contando su un principio per il quale si deve dare per scontato che i componenti del team abbiano fatto del loro meglio. Nessuno dice a nessuno cosa deve fare, ma si conta su un'autoregolazione del team, che presuppone una parità tra i membri che lo compongono.

Fra i principi cardine della metodologia Agile, passato agli studenti del Carlo Porta, c'è poi il principio, detto all'americana "Fail fast, fail often" (fallisci in fretta, fallisci spesso). Portare questo concetto in ambito scolastico, significa scardinare un presupposto più che mai intrinsecamente legato al classico concetto di apprendimento. La scuola e le famiglie aborriscono (rare eccezioni a parte) qualsiasi fallimento dei ragazzi.

Nel progetto "Trilogy Be Agile" si è cercato invece di far comprendere agli studenti come il fallimento sia assolutamente indispensabile per crescere, per migliorarsi, per imparare e non fuggire di fronte ad un ostacolo. Questo concetto ha sconvolto molto i ragazzi, facendo tirar loro un sospiro di sollievo, oltre che far nascere l'esigenza di comunicarlo e farlo comprendere ai propri genitori.

Partendo da una mera osservazione della realtà, in cui è sotto gli occhi di tutti un costante cambiamento del mondo del lavoro, il progetto "Be Agile" ha cercato di far comprendere ai ragazzi come sia necessaria una continua formazione, soprattutto in ambiti apparentemente non legati a una specifica professione. Ampliare la propria "cassetta degli strumenti", permetterà loro di restare costantemente inseriti nel mondo del lavoro, seppur in modalità, di volta in volta, diversa.

L'antico principio darwiniano secondo cui a sopravvivere è la specie che ha maggiori capacità di adattamento, questo può essere applicato anche in ambito lavorativo, oggi più che mai. Più conoscenze, più esperienze permetteranno a chiunque di essere maggiormente

al passo con i tempi e in grado, quindi, di reinventarsi e adattarsi agli scenari lavorativi che di volta in volta si configurano.

Un progetto come questo, ampio, trasversale, innovativo, ha permesso di pervenire a riflessioni insolite e inusuali in ambito scolastico. I ragazzi del progetto hanno riferito di avere avuto modo di sperimentare una dimensione di concetti, idee e sentimenti a cui non erano minimamente abituati. Si è cercato di parlare alle loro "pance", certi che in questo modo, avrebbero meglio appreso e sperimentato concetti fondamentali in un percorso umano che voglia sperimentare e vivere la felicità.

6.2. Come l'Agile viene utilizzato nella vita di tutti i giorni

Nel suo discorso a TED, Bruce Feiler ha definito le famiglie di oggi "fuori controllo" e ha raccontato come la sua famiglia abbia utilizzato la gestione agile dei progetti per migliorare la propria vita. Inizia col condividere i risultati di un sondaggio; è emerso che i ragazzi desiderano soprattutto vedere i loro genitori meno stanchi e meno stressati.

Feiler descrive poi come l'utilizzo di standup meeting familiari con checklist mattutine abbia ridotto drasticamente lo stress nella sua famiglia. "La settimana in cui abbiamo introdotto una checklist mattutina in casa nostra, le urla dei genitori si sono dimezzate", dice.

Utilizzando le retrospective e dando ai figli la possibilità di fornire input su ricompense, conseguenze e piani per la successiva "iterazione", la famiglia Feiler ha acquisito una visione dei pensieri più intimi dei figli e ne ha incoraggiato l'indipendenza.

Egli ritiene che le famiglie possano trovare la felicità adattando un "manifesto della famiglia agile" con i principi di adattamento e responsabilizzazione.

Sebbene molte delle tecniche e delle fasi delle metodologie agile siano incentrate sulla collaborazione, alcuni concetti possono essere utilizzati anche nel team di una sola persona. Lavorare a partire da un backlog di prodotto prioritario, suddividere il lavoro in iterazioni, limitare i lavori in corso e riflettere regolarmente sono solo alcuni dei concetti mutuati dall'agile che le persone possono utilizzare per gestire meglio la propria vita personale.

Nell'articolo di Lifehack "Scrum for One", Dustin Wax descrive come molti degli elementi di Scrum possano essere adattati alla produttività individuale. Ecco alcuni esempi:

- Iniziare prima di avere tutte le risposte e poi essere creativi nel trovare soluzioni agli ostacoli che si incontrano.

- Riflettere e adattarsi costantemente. Dedicare qualche minuto al giorno alla pianificazione, alla riflessione e alla domanda: "Cosa mi ostacola?".
- Lavorare per raggiungere obiettivi a breve termine chiaramente definiti. Suddividere i grandi obiettivi in obiettivi a breve termine, con scadenze precise, aiuta a creare un senso di realizzazione e a progredire.
- Sprint: rimanere concentrati sull'obiettivo. Stabilire dei momenti della giornata in cui poter lavorare senza distrazioni fino al raggiungimento dell'obiettivo.

7. Conclusioni

Si è visto come la metodologia Agile offra numerosi vantaggi e una routine ben precisa. Si riscontra purtroppo un basso grado di diffusione, soprattutto da parte del cliente, per motivi analizzati nella tesi, quali definizione di un budget, tempi precisi e requisiti chiari fin dall'inizio del progetto.

Tuttavia, la metodologia Agile è un concetto che viene da sempre applicato nella vita di tutti i giorni ed in alcune realtà professionali, anche prima della stesura del manifesto Agile del 2001. La flessibilità e la ricerca di collaborazione la rendono perfetta per il periodo che stiamo vivendo, soprattutto dopo la pandemia, e la candidano ad essere protagonista negli anni futuri.

Le crescenti esigenze di flessibilità, in un mercato di prodotti software sempre più competitivo, hanno portato Accenture ad adottare questa metodologia per la gestione delle commesse. La gestione dei progetti con questo metodo è ampia e completa; porta spesso il team a riflettere sul grado di utilizzo, cosa migliorare e in che modo. Creando un approccio "su misura", Accenture è riuscita a superare i soliti schemi predeterminati e generalizzati per la gestione del lavoro ed risultare più efficiente e organizzato.

Bibliografia e sitografia

Riferimenti a documenti pubblici

Thomas, D., & Hunt, A. (2019, September 13). *The Pragmatic Programmer: Your Journey To Mastery, 20th Anniversary Edition (2nd Edition)* (2nd ed.). Addison-Wesley Professional.

Apke, L. (2015, September 2). *Understanding the Agile Manifesto* (null). lulu.com.

Subramaniam, V. (n.d.). *Practices of an Agile Developer*. Shroff Publishers & Distributors Pvt Ltd.

Martin, R. (2019, October 17). *Clean Agile: Back to Basics (Robert C. Martin Series)* (1st ed.). Pearson.

Riferimenti a siti web

Agile | Perspectives. (n.d.). Thoughtworks. Retrieved September 7, 2022, from <https://www.thoughtworks.com/perspectives/edition1-agile>

Agile Alliance. (2022, May 26). *What is Agile? | Agile 101 | Agile Alliance*. Agile Alliance |. Retrieved September 7, 2022, from <https://www.agilealliance.org/agile101/>

Going from Waterfall to Agile. (n.d.). Thoughtworks. Retrieved September 7, 2022, from <https://www.thoughtworks.com/insights/blog/waterfall-agile>

Agile vs Waterfall | 5 Reasons why Agile is better. (2022, February 23). The CRM Team. Retrieved September 7, 2022, from <https://thecrmteam.com/5-reasons-why-agile-is-better-than-waterfall/>

Berman, J. (2022, April 21). *Agile Frameworks*. Planview. Retrieved September 7, 2022, from <https://www.planview.com/resources/guide/what-is-agile-program-management/agile-frameworks/#:%7E:text=An%20Agile%20framework%20is%20a,at%20scale%2C%20across%20many%20teams.>

Atlassian. (n.d.). *What is Scrum?* Retrieved September 7, 2022, from <https://www.atlassian.com/agile/scrum#:%7E:text=What%20is%20Scrum%3F,and%20loses%20to%20continuously%20improve.>

Academy, A. (n.d.). *The origins of Scrum*. Agile Academy. Retrieved September 7, 2022, from <https://www.agile-academy.com/en/foundations/the-origins-of-scrum/>

Scrum: A Brief History of a Long-Lived Hype. (2018, July 25). Scrum.org. Retrieved September 7, 2022, from <https://www.scrum.org/resources/blog/scrum-brief-history-long-lived-hype>

What is Scrum Team? - Scrum Guide. (n.d.). Retrieved September 7, 2022, from <https://www.visual-paradigm.com/scrum/what-is-scrum-team/>

Kukhnavets, P. (2021, February 17). *What are Scrum Artifacts and the Types of Scrum Ceremonies?* Hygger: Project Management Software & Tools for Companies. Retrieved September 7, 2022, from <https://hygger.io/blog/scrum-artifacts-and-scrum-ceremonies/>

Academy, A. (n.d.-a). *Events and Artifacts of Scrum*. Agile Academy. Retrieved September 7, 2022, from <https://www.agile-academy.com/en/foundations/events-and-artifacts-of-scrum/>

Atlassian. (n.d.-a). *Learn about Agile Scrum Artifacts*. Retrieved September 7, 2022, from <https://www.atlassian.com/agile/scrum/artifacts>

May, S. (2021, April 30). *Why is scrum the most popular agile methodology?* Zeolearn. Retrieved September 7, 2022, from <https://www.zeolearn.com/magazine/why-scrum>

L, D. (2021, December 31). *Why is Scrum the Most Popular Agile Framework?* Life Your Way. Retrieved September 7, 2022, from <https://lifeyourway.net/why-is-scrum-the-most-popular-agile-framework/>

20 best agile project management tools. (n.d.). Justinmind. Retrieved September 7, 2022, from <https://www.justinmind.com/blog/agile-tools/>

What is a Sprint in Scrum? (n.d.). Scrum.org. Retrieved September 7, 2022, from <https://www.scrum.org/resources/what-is-a-sprint-in-scrum>

Atlassian. (n.d.-b). *Scrum Sprints: Everything You Need to Know*. Retrieved September 7, 2022, from <https://www.atlassian.com/agile/scrum/sprints>

Accenture | Imprese associate | Assoconsult. (2019, October 13). Project Management Center. Retrieved September 7, 2022, from <https://www.assoconsult.org/imprese-associate/accenture/>

Agile Transformation Services for Business Agility | Accenture. (n.d.). Retrieved September 7, 2022, from <https://www.accenture.com/us-en/services/software-engineering/agile-transformation>

Accenture | (n.d.-d). Retrieved September 7, 2022, from <https://www.accenture.com/tw-en/case-studies/technology/how-agile-testing-brought-efficiency-swisscom>

Accenture | (n.d.-e). Retrieved September 7, 2022, from <https://www.accenture.com/tw-en/case-studies/aerospace-defense/leonardo-case-study>

Accenture | (n.d.). Retrieved September 7, 2022, from <https://accenture.com/tw-en/insights/strategy/mergers-acquisitions-integration>

Accenture | (n.d.-f). Retrieved September 7, 2022, from <https://www.accenture.com/tw-en/case-studies/operations/ensuring-excellence-agile-operations>

Accenture | (n.d.-b). Retrieved September 7, 2022, from <https://accenture.com/tw-en/insights/technology/learning-journeys-agile-leader>

Grilli, A. B. A. B. S. (2019, October 13). *Metodologia Agile: non solo software*. Tech Economy 2030. Retrieved September 7, 2022, from <https://www.techeconomy2030.it/2019/04/17/metodologia-agile-non-solo-software/>

Digite Inc. (n.d.). Digite. Retrieved September 7, 2022, from <https://www.digite.com/blog/agile-methodologies-for-non-software-teams/>

DBS Interactive. (n.d.). Retrieved September 7, 2022, from <https://www.dbswebsite.com/blog/adopting-agile-in-your-daily-life/>

Francino, Y. (2019, January 22). *How to use agile techniques to manage your personal life*. TechBeacon. Retrieved September 7, 2022, from <https://techbeacon.com/app-dev-testing/personal-scrum-using-agile-techniques-manage-your-life>

Agile Way. (2020, April 13). - *Stand-up, daily!* Retrieved September 7, 2022, from <https://www.agileway.it/>