



**Politecnico
di Torino**

Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria del Cinema e dei Mezzi di Comunicazione

A.A. 2023/2024

Sessione di Laurea Dicembre 2024

Pensare e agire Agile: i framework Scrum e SAFe® per i progetti IT nel settore bancario.

Il caso Santander Consumer Bank

Relatrice:

Nicoletta Gay

Tutor aziendale:

Barbara Torricelli

Candidato:

Davide Chessa

Indice

Introduzione	3
1. Pensare Agile	6
1.1 Lean thinking	6
1.2 Il Manifesto Agile.....	22
1.3 Una questione di mentalità: lean and agile mindset	30
2. Agire Agile.....	35
2.1 Il framework Scrum	38
2.1.1 I ruoli chiave	43
2.1.2 Gli Eventi	53
2.1.3 Gli Artefatti	59
2.2 SAFe®.....	63
2.3 Il modello Spotify	79
3. Lessons Learned: il caso Santander Consumer Bank	84
3.1 Banco Santander e Santander Consumer Bank	87
3.2 Il modello Waterfall messo alla prova.....	93
3.3 Sviluppare durante i test	102
3.4 Il ruolo del Product Owner.....	114
3.5 La gestione dei fornitori esterni	125
4. Actions.....	137
4.1 Agile Adoption.....	140
4.1.1 Roadmap	145
4.1.2 Evitare lo ScrumBut.....	151

4.2 Agile Transformation	157
4.2.1 Roadmap	163
4.2.2 Il “Big Bang” di Paypal	172
4.3 Dall’Adoption alla Transformation in Santander Consumer Bank.....	178
Conclusione	197
Bibliografia	201
Sitografia	203
Ringraziamenti	204

INTRODUZIONE

Adottare una metodologia Agile all'interno di un'azienda significa effettuare un vero e proprio cambiamento radicale della mentalità e delle strutture che la compongono. Non è sufficiente applicare strumenti seguendo determinate regole oppure semplicemente riorganizzare i gruppi di lavoro, ma risulta fondamentale promuovere la formazione dei dipendenti e la collaborazione in modo tale che il loro modo di pensare cambi e sia favorevole al cambiamento, focalizzandosi sul miglioramento continuo. Il cliente e il valore percepito dei prodotti e dei servizi assumono un ruolo centrale e i leader diventano i responsabili della creazione di un ambiente che renda favorevole lo sviluppo di una nuova cultura aziendale. L'introduzione di framework come Scrum e SAFe può sostenere e favorire una nuova gestione dell'intera organizzazione, dal singolo team ai livelli più alti, ma è solo tramite la definizione di un modello opportunamente modellato su una determinata realtà aziendale che si possono gettare le basi di una concreta trasformazione Agile.

Nel primo capitolo della tesi, ci si concentra sui principi del *lean thinking* e della mentalità Agile, due approcci che sono nati in contesti diversi, ma che condividono una visione di efficienza e di flessibilità organizzativa. Partendo dal contesto manifatturiero giapponese di inizio Novecento, in cui il Toyota Production System ha rappresentato un vero e proprio modello rivoluzionario di organizzazione del lavoro, si arriva ai primi anni Duemila con la pubblicazione del Manifesto Agile per lo sviluppo del software in modo flessibile e più efficiente. Si tratta di due modelli che, benché siano nati in due settori diversi, hanno avuto una vasta applicazione anche in altri ambiti, dimostrando che, prima ancora di essere strumenti, sono dei modelli culturali. In tal senso, si introduce il concetto di mentalità come pilastro fondante del cambiamento, facendo una distinzione tra una mentalità fissa e a una mentalità di crescita: il *lean and agile mindset*. Si vuole quindi evidenziare come *in primis* sia fondamentale pensare Agile.

Nel secondo capitolo, si presentano quelli che sono gli strumenti per agire Agile messi a disposizione delle aziende. Il primo framework esaminato è Scrum, ponendo un'attenzione particolare sui ruoli chiave e le responsabilità che introduce per garantire una

gestione collaborativa e costruita intorno al valore. Inoltre, si descrivono gli eventi e gli artefatti che sono elementi fondamentali per strutturare in modo trasparente ed efficace il flusso di lavoro svolto da un team. Dato che l'applicazione di strumenti che migliorano l'efficienza a livello di gruppo di lavoro potrebbe risultare insufficiente per le aziende più complesse e articolate, si introduce un altro framework, SAFe® - Scaled Agile Framework, in quanto permette di scalare i principi Agile all'intera organizzazione, garantendo che i benefici per i singoli team Scrum si estendano anche ai diversi livelli più articolati. Si analizzano quindi i principi su cui si basa e che ne consentono la sua implementazione corredati da opportuni esempi che aiutano a collegare la teoria alla pratica. In tal senso, a conclusione del capitolo, si presenta il cosiddetto *modello Spotify*, analizzando le modalità con cui l'azienda Spotify AB ha adottato con successo la metodologia Agile.

Nel terzo capitolo, ci si concentra sulle *lessons learned* dall'esperienza di tesi in azienda e di tirocinio come Product Owner presso Santander Consumer Bank, banca operante nel settore dei prodotti di credito al consumo parte del gruppo bancario internazionale Banco Santander. Il modello Waterfall utilizzato per la gestione della maggior parte dei progetti è messo in discussione analizzando un caso pratico che ha fatto emergere alcune criticità in un ambito come quello bancario in cui i ritardi possono avere degli impatti notevoli. Inoltre, si evidenzia l'importanza del testing per garantire che il software sviluppato mantenga elevati standard di qualità a conferma di alcuni dei principi presentati nei capitoli precedenti. In tal senso, si evidenziano quelli che sono stati alcuni degli errori e delle problematiche emerse con maggiore frequenza come punti di attenzione su cui migliorare. Si presenta, successivamente, l'importanza e il ruolo che ha la figura del Product Owner nelle diverse fasi di un progetto così come si è potuto notare durante la tesi in azienda e il tirocinio. Alcuni esempi pratici aiutano a comprendere meglio i diversi atteggiamenti che il Product Owner assume a seconda delle situazioni per facilitare la comunicazione, massimizzare il valore e garantire la qualità del prodotto, interagendo con gli *stakeholders* di un progetto e il team. Infine, si approfondisce la gestione dei fornitori terzi partendo dalle evidenze emerse durante il tirocinio e arrivando al tema cruciale della contrattualistica Agile.

Nel quarto capitolo, dalle *lessons learned* si passa alle *actions*, cioè si esplorano le strategie di *Agile Adoption* e di *Agile Transformation* ponendo particolare attenzione ai possibili ostacoli ed errori da evitare come per esempio quello che viene definito *ScrumBut*. Si presentano due roadmap distinte: una per l'Agile Adoption e l'implementazione di Scrum a livello di team; una per l'Agile Transformation verso l'implementazione di SAFe per una trasformazione completa dell'organizzazione. Per comprendere meglio la complessità e gli step da seguire per una trasformazione di successo si prende a titolo d'esempio il caso di PayPal. Il capitolo termina proponendo a Santander Consumer Bank Italia alcune azioni concrete sulla base delle *lessons learned* che si consiglia di implementare per un miglioramento nel breve termine. Inoltre, vengono proposti alcuni suggerimenti e alcune *best practices* per l'avvio di una vera e propria trasformazione Agile dell'intera organizzazione aziendale.

La tesi evidenzia quindi che, grazie all'adozione di una mentalità Agile e Lean e all'utilizzo di framework organizzativi come Scrum e SAFe, si possono ottenere miglioramenti significativi in termini di efficienza e di flessibilità aziendale, anche in un ambito complesso come quello bancario e dei prodotti di credito al consumo. Le azioni proposte e le raccomandazioni presentate permettono di agevolare l'implementazione delle pratiche Agile in un modo più accurato e opportunamente plasmato sulla realtà aziendale di Santander Consumer Bank Italia, ponendo le basi per una vera e propria trasformazione organizzativa in grado di rispondere adeguatamente alle sfide del mercato moderno e alla costante e rapida evoluzione tecnologica.

1. PENSARE AGILE

Applicare la metodologia Agile nell'ideazione e nella realizzazione di progetti e nella loro successiva manutenzione vuol dire prima di tutto avere una mentalità Agile. Non si tratta semplicemente di attenersi a una serie di regole fisse e ben definite da seguire minuziosamente, ma si tratta di un vero e proprio esercizio costante che implica un cambiamento nel modo di approcciare il lavoro svolto quotidianamente e una continua revisione di quanto si sta realizzando per mettere in atto azioni correttive che migliorano l'operato. Un progetto può essere definito come "Ideazione, piano, proposta per l'esecuzione di un lavoro o di una serie di lavori"¹. Di sua natura può quindi riguardare una molteplicità di lavori ognuno dei quali introduce una quantità elevata di variabili eterogenee in gioco. Pensare di poter disporre di una serie di norme a cui attenersi saldamente e affidare unicamente a esse il successo di ogni progetto non porta ai risultati desiderati, soprattutto a lungo termine. Inoltre, se si considera che all'interno di un'azienda i progetti sono il più delle volte dipendenti l'uno dall'altro, anche se di natura molto diversa, si comprende quanto sia fondamentale avere un approccio più elastico che ne consenta la coesistenza e la buona riuscita comune. Insomma, non esiste una regola che permetta di far lavorare un'azienda in modalità Agile, ma esiste un modo di porsi e di affrontare la propria attività in modo agile.

1.1 LEAN THINKING

Aumentare l'efficienza e ridurre gli sprechi è un obiettivo cruciale per tutte le aziende, indipendentemente dal settore in cui operano. Tuttavia, è nel settore manifatturiero, e in particolare quello automobilistico, che storicamente si sono sviluppate le tecniche avanzate di gestione del lavoro. Uno dei primi passi in questa direzione si ebbe all'inizio del Novecento, con l'introduzione dell'organizzazione scientifica del lavoro di Frederick Taylor, noto per aver sistematizzato e ottimizzato i processi lavorativi industriali

¹ Definizione Treccani: <https://www.treccani.it/vocabolario/progetto/> (ultimo accesso in data 28/09/2024).

attraverso una rigorosa analisi del lavoro umano e l'introduzione di tecniche di gestione scientifica. Il contributo di Taylor, seguito dall'adozione della produzione di massa da parte di Henry Ford con la sua catena di montaggio, ha segnato una svolta importante nella produzione industriale. Ford riuscì a standardizzare il processo di assemblaggio delle automobili, riducendo drasticamente i tempi di produzione e abbattendo i costi. Tuttavia, questo modello presentava delle significative limitazioni, poiché era focalizzato esclusivamente sulla produzione in massa di grandi quantità di prodotti, spesso non corrispondenti alla reale domanda di mercato. L'industria automobilistica americana era quindi a dir poco enorme messa in confronto con quella di altri Stati come per esempio il Giappone. In effetti, in quest'ultimo si pensava che non fosse possibile avviare una produzione automobilistica perché non si trovavano né tecnici specializzati né capitali lontanamente paragonabili a quelli dell'industria automobilistica americana². Effettivamente, il Giappone non avrebbe avuto la possibilità di importare il sistema americano e di applicarlo senza adattamenti nella propria industria. In questo contesto, Kiichirō Toyoda iniziò a interrogarsi su come realizzare un sistema di produzione che si adattasse alla realtà del suo Stato. È curioso notare che si rese subito conto di un aspetto fondamentale che ritornerà più volte all'interno di questa tesi, cioè di come non sia possibile applicare alla propria azienda senza modifiche un modello preesistente già adottato in altre realtà anche se rivelatosi vincente.

Sicuramente la situazione economica del Giappone, ben diversa da quella americana a seguito della guerra, influenzò notevolmente tale pensiero, suggerendo quindi la necessità di elaborare un sistema produttivo adattato alle specifiche esigenze del contesto giapponese. Per poter sostenere la produzione, era fondamentale ridurre al minimo gli sprechi e ottimizzare ogni fase del processo produttivo. Questo obiettivo non solo permetteva di rispondere più efficacemente alle fluttuazioni della domanda, ma contribuiva anche a migliorare la qualità dei prodotti. I vari tentativi di Kiichirō Toyoda di applicazione di un modello giapponese basato sulla riduzione degli sprechi furono rallentati dalla Seconda guerra mondiale e non riuscì a portare a termine la realizzazione di un sistema di

² Wakamatsu Y., *Toyota Way. Il valore delle persone nel Toyota Production System*, Franco Angeli, Milano, 2008, p. 21.

produzione giapponese in quanto morì nel 1952. Tuttavia, il suo pensiero venne ereditato dal cugino Eiji Toyoda e fu alla base del Toyota Production System (TPS), sviluppato insieme a Taiichi Ōno, che vede come pilastri fondamentali il concetto di *Just in time* e di *jidōka*. Quest'ultimo è stato tramandato da Sakichi Toyoda, padre di Kiichirō, e può avere diversi significati, ma attenendoci a quanto indicato sul sito ufficiale di Toyota possiamo definirlo come:

“The principle of designing equipment to stop automatically and to detect and call attention to problems immediately whenever they occur. [...] operators are equipped with the means to stop the production flow whenever they note anything suspicious (human jidoka), thereby preventing the waste that would result from producing a series of defective items.”³

In questo modo è possibile ridurre gli sprechi, ricercare le cause delle problematiche e apportare un miglioramento al processo produttivo. Perfezionare costantemente i processi produttivi è quindi un aspetto fondante del TPS e l'importanza della componente umana che analizza le possibili problematiche per portare un miglioramento permette di introdurre un altro termine fondamentale, il *kaizen*. Per comprenderne il significato possiamo affidarci alle parole di Ōno che lo definisce come “il miglioramento grazie all'impiego dell'intelletto”⁴. Intelletto che però non si limita a quello di manager o di figure che ricoprono ruoli più elevati nell'azienda, bensì l'intelletto di tutti i lavoratori coinvolti in un determinato processo di produzione. Si tratta quindi di coinvolgere gli operai non solo da un punto di vista fisico per l'esecuzione delle loro attività, ma anche da un punto di vista mentale invitandoli a proporre dei miglioramenti. In effetti, gli operai, lavorando in prima linea, possono accorgersi di problematiche ricorrenti, di rischi e di inefficienze stimolando la propria mente a produrre suggerimenti di azioni da intraprendere che potrebbero apportare un vantaggio all'intero processo. Taiichi Ōno, durante le sue

³ Toyota Blog, *Jidoka – Toyota Production System guide*, 31 maggio 2016, disponibile online all'indirizzo <https://mag.toyota.co.uk/jidoka-toyota-production-system/> (ultimo accesso in data 29/09/2024).

⁴ Wakamatsu Y., *Op. cit.*, 2008, p. 28.

visite agli stabilimenti americani, notò una caratteristica peculiare della produzione di massa: quando si verificavano errori, come a esempio un componente difettoso, la risposta immediata era quella di risolvere il problema a valle, cercando di correggere il singolo pezzo difettoso. Tuttavia, questo approccio si concentrava esclusivamente sulla risoluzione del sintomo, senza un'analisi approfondita delle cause alla radice del problema. Gli errori venivano spesso considerati come eventi isolati e casuali, con la speranza che non si ripetessero, piuttosto che essere visti come opportunità di miglioramento dell'intero sistema produttivo⁵. Ciò rischiava di produrre una sostanziale quantità di scarti di produzione che non potevano essere accettati nel sistema di produzione giapponese.

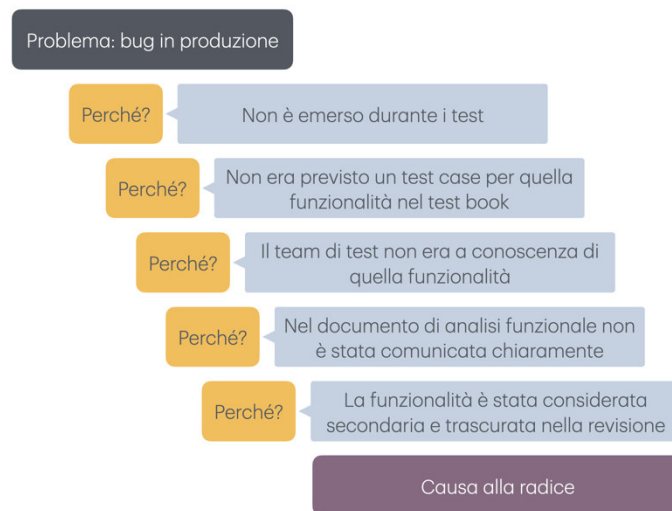


Figura 1 - Esempio della tecnica dei cinque perché applicata per trovare la causa radice a seguito di un bug emerso in produzione dopo il rilascio della nuova versione di un software.

Per questo motivo, Ōno invitò tutti gli operai ad approfondire gli errori riscontrati identificandone le cause alla radice utilizzando il metodo dei cinque perché. Questo metodo permette, interrogandosi ripetutamente sulle motivazioni che hanno portato a un problema, di risalire man mano alla causa alla radice. Non si tratta di identificare e di riuscire ad attribuire la colpa a un determinato responsabile, bensì di scomporre e analizzare più attentamente i diversi fatti che in catena hanno prodotto la problematica riscontrata. In

⁵ Womack J.P., Jones D.T., Roos D., *The Machine That Changed the World: The Story of Lean Production*, Harper Perennial, New York, 1990, p. 56.

questo modo è possibile apprendere dagli sbagli passati per migliorare le azioni future. Si pensi a titolo d'esempio al rilascio in produzione di una nuova versione di un software aziendale. In Figura 1 è possibile visualizzare l'analisi dei cinque perché applicata a un bug emerso dopo il rilascio denotando possibili criticità nel modo in cui sono stati raccolti e definiti i requisiti funzionali dell'aggiornamento appena rilasciato, suggerendo quindi una maggiore attenzione nella fase iniziale di progetto, favorendo una più precisa e puntuale identificazione dei possibili impatti che l'aggiunta di nuove funzionalità in un software può avere anche su aspetti secondari che in questo esempio erano stati trascurati durante la revisione dei requisiti funzionali.

Un altro concetto fondante del Toyota Production System, sempre atto all'eliminazione degli sprechi, è il *Just in time*. In poche parole, si tratta di produrre in base alla reale domanda evitando quindi di avere in uscita alla produzione dei prodotti in eccesso, i quali necessiterebbero di un magazzino per lo stoccaggio. Lo stato ideale a cui tendere, spiega Ōno, è quello di rendere disponibili le componenti necessarie per l'assemblaggio nel momento in cui ve ne è il reale bisogno e nella quantità opportuna⁶. Quindi non si tratta solamente di produrre beni seguendo l'andamento della domanda, eliminando gli eccessi di produzione, ma questo concetto mira a ridurre anche a monte la quantità di materie prime inutilizzate. Quest'ultime non devono quindi essere accumulate prima ancora della messa in produzione del prodotto finale, ma l'obiettivo è quello di far raggiungere il materiale utile per la catena produttiva solo nel momento del bisogno. Ciò ha un notevole impatto sul modo in cui l'azienda si relaziona con i propri fornitori con i quali è fondamentale instaurare un rapporto elastico e ben orchestrato in modo tale che possano essere in grado di fornire tempestivamente il materiale necessario in base all'andamento del processo produttivo. In questo modo, è possibile ridurre ritardi di produzione disponendo di un flusso costante e sincronizzato, guidato da una costante analisi dell'andamento del processo di produzione. Ogni step della catena prende in ingresso quanto prodotto dallo step precedente e produce componenti che saranno utilizzate in quello successivo. Per evitare sprechi è fondamentale disporre di un meccanismo che renda

⁶ Ōno T., *Toyota Production System: Oltre la produzione di massa*, Milano, Franco Angeli, 1993, p. 4.

possibile la segnalazione a uno step precedente della necessità di nuove componenti in ingresso solo quando queste risultano necessarie. Servono quindi dei segnali che lungo tutta la catena di produzione favoriscano un flusso costante. È qui che entra in gioco il termine *kanban* che possiamo definire come:

“A system that conveys information between processes and automatically orders parts as they are used up. Every item or box of items that flows through the production process carries its own kanban. Kanbans come off items that have been used or transported and go back to the preceding processes as orders for additional items.”⁷

Quindi ogni step, restituendo un segnale allo step precedente, è in grado di indicare la necessità di ricevere in ingresso nuovi componenti per far sì che la produzione prosegua nella sua fluida esecuzione. Mettere in pratica un tale meccanismo è tutt’altro che semplice soprattutto se si considera il fatto che non esistono inventari e magazzini da cui attingere, ma i componenti sono prodotti nel momento in cui vi è una reale richiesta lungo la catena. Si pensi, per esempio, ai possibili impatti a cascata che potrebbe avere un pezzo difettoso prodotto da uno step precedente verso quelli successivi. Risulta inevitabilmente importante segnalare immediatamente il rilevamento dell’errore con un conseguente e rovinoso blocco totale della produzione. Tuttavia, come visto in precedenza, nel TPS è fondamentale apprendere dagli errori e la possibilità di fermare la catena di produzione è vista come un’opportunità. Infatti, una completa analisi di ciò che ha causato l’errore permetterà di mettere in atto in futuro delle correzioni che ridurranno drasticamente la probabilità che una tale situazione possa verificarsi nuovamente. Si tratta quindi di un sistema in cui un flusso di produzione controllato è ottenuto grazie a una minuziosa sincronizzazione di ogni step della catena di montaggio mediante l’uso di segnali, riducendo al minimo le possibilità di errore grazie a un miglioramento costante.

⁷ Toyota Blog, *Kanban – Toyota Production System guide*, 31 maggio 2013, disponibile online all’indirizzo <https://mag.toyota.co.uk/kanban-toyota-production-system/> (ultimo accesso in data 30/09/2024).

Nel corso della seconda metà del XX secolo, i principi descritti finora sono stati progressivamente applicati a settori diversi dall'automotive. Per esempio, in ambito sanitario, il concetto di *Just in time* è essenziale per garantire che farmaci e strumenti medici siano disponibili esattamente quando richiesti, riducendo così i costi di stoccaggio e minimizzando gli sprechi. Per quanto riguarda l'IT, l'adozione di un approccio orientato al miglioramento continuo consente di mantenere le soluzioni software costantemente aggiornate, aspetto fondamentale in un contesto tecnologico in rapido cambiamento. Quindi, più in generale, i principi alla base del TPS possono essere applicati a qualunque settore produttivo o dei servizi ove vi sia la necessità di una migliore gestione delle risorse e di una maggiore efficienza dei processi. Nel 1988, in un articolo pubblicato sulla rivista *Sloan Management Review*, John Krafcik, ingegnere del Massachusetts Institute of Technology (MIT), introduce il termine *lean* per descrivere il sistema di produzione adottato da Toyota in contrapposizione al termine *buffered* utilizzato per descrivere il sistema di produzione occidentale. Krafcik evidenzia che un sistema buffered si basa sull'utilizzo di ampie scorte e di magazzini per poter compensare le inefficienze e le variazioni nella produzione, mentre in un sistema lean le scorte sono ridotte al minimo, i problemi sulla qualità possono essere identificati più rapidamente e ci si concentra sull'efficienza. È interessante l'analogia proposta con il mondo della finanza: la produzione lean presenta un rischio maggiore (si pensi al discorso fatto precedentemente sulle conseguenze bloccanti per l'intera catena di produzione in presenza di un pezzo difettoso) ma il guadagno potenziale è grande, quindi una sorta di investimento ad alto rischio con alto rendimento; nella produzione buffered, invece, il rischio a breve termine è basso, ma lo è anche il potenziale di guadagno di performance a lungo termine. Krafcik afferma che, tutto sommato, "lean plants tend to perform much better than buffered plants"⁸.

In seguito, nel 1990, James P. Womack, Daniel T. Jones e Daniel Roos pubblicarono il libro *The Machine That Changed the World*, basato su un approfondito studio condotto dal Massachusetts Institute of Technology (MIT) sul settore automobilistico globale. Grazie

⁸ Krafcik J.F., "Triumph of the Lean Production System", in *Sloan Management Review*, vol. 30, n. 1, pp. 41-52, 1988.

a questa pubblicazione si diffusero maggiormente i concetti fondanti della *lean production* presentandone i vantaggi in termini di efficienza, di riduzione degli sprechi e di miglioramento della qualità. Non solo il sistema adottato da Toyota permetteva di ottenere performance superiori dei modelli occidentali, ma era anche in grado di ridurre i costi operativi senza compromettere la qualità del prodotto finale, adattandosi rapidamente alle variazioni del mercato. Gli autori sono quindi molto entusiasti di presentare gli aspetti vincenti della produzione snella (termine italiano con cui possiamo tradurre lean) sostenendo ripetutamente che possa essere applicata in qualsiasi industria nel mondo, sia automobilistica, sia di altro tipo, e che ciò porterebbe a un grande cambiamento⁹. Tuttavia, la pubblicazione del libro oltre ad aver portato molto entusiasmo nei lettori, ha fatto accrescere in ognuno di loro un quesito ricorrente: come è possibile diventare lean? Ed effettivamente a questo punto del discorso dovrebbe essere una domanda più che lecita. James P. Womack e Daniel T. Jones nel 1996 pubblicano un nuovo libro, intitolato *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, ovvero, nella sua versione italiana, *Lean Thinking. Come creare valore e bandire gli sprechi*. L'obiettivo non è quello di fornire delle tecniche particolari o degli strumenti specifici per effettuare il passaggio dalla produzione di massa a quella snella, ma il loro desiderio è quello di

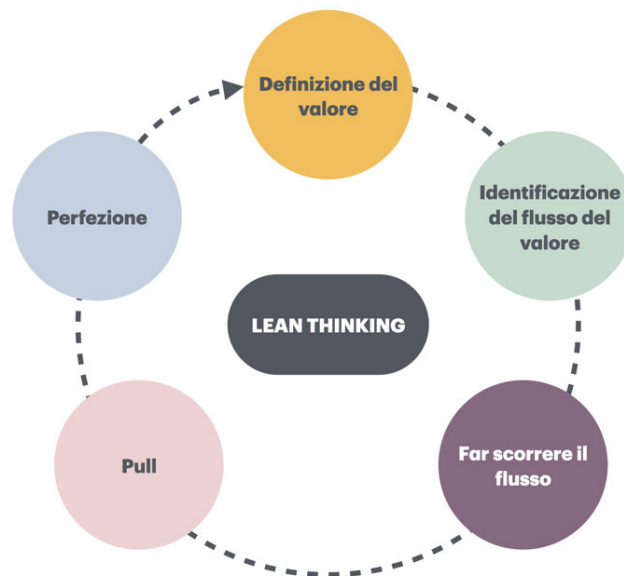


Figura 2 - I 5 principi del lean thinking

⁹ Womack J.P., Jones D.T., Roos D., *Op. cit.*, 1990, pp. 8 e 278.

definire dei principi che possano guidare le azioni di chi vuole abbracciare una metodologia lean per la produzione¹⁰.

Si inizia quindi a parlare di *lean thinking*, una vera e propria filosofia gestionale, basata su cinque principi così come indicati in Figura 2.

- 1. Definizione del valore:** come primo passo del pensiero snello, la definizione del valore assume un ruolo molto importante. Solo i clienti finali possono definire il valore di un determinato prodotto e risulta quindi fondamentale mettersi nei loro panni per comprendere al meglio quali sono le caratteristiche e le funzionalità che portano una reale aggiunta di valore. Il valore assume significato solo nel momento in cui lo si esprime di un determinato prodotto in grado di soddisfare le esigenze del cliente a un dato prezzo e in un dato momento¹¹. È quindi errato pensare che il produttore possa imporre la propria visione di un prodotto alla clientela senza prima aver adeguatamente compreso e mediato le esigenze dei clienti stessi. Si pensi, per esempio, in ambito IT, di voler implementare una nuova funzionalità sul sito istituzionale di un'azienda per migliorare il servizio clienti. Il team potrebbe decidere di aggiungere un complesso sistema di intelligenza artificiale (per esempio un chatbot) per rispondere automaticamente alle domande dei clienti, pensando che questo possa migliorare l'esperienza utente. Tuttavia, se i clienti preferiscono un'interazione più semplice e diretta con un operatore umano o necessitano più semplicemente di un'interfaccia che sia più intuitiva per trovare le informazioni di cui hanno bisogno in autonomia, senza dover contattare ogni volta il servizio clienti, questa funzionalità avanzata non solo risulterà inutile, ma potrebbe addirittura peggiorare l'esperienza d'uso. In questo caso, la decisione del produttore di aggiungere una funzionalità che non risponde alle reali esigenze degli utenti si pone in netto contrasto con questo principio. Nonostante le buone intenzioni del team e la scelta di utilizzare una tecnologia

¹⁰ Womack J.P., Jones D.T., *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, Simon & Schuster, New York, 1996, pp. 9-12.

¹¹ Womack J.P., Jones D.T., *Op. cit.*, 1996, p.16.

avanzata e di ultima generazione, il mancato allineamento con le reali aspettative del cliente potrebbe comportare un aumento degli sprechi – si pensi ai costi di sviluppo e di manutenzione – e una diminuzione della soddisfazione dell'utente finale, vanificando così gli sforzi fatti per migliorare il servizio. Sarebbe stato più opportuno raccogliere e analizzare i feedback dei clienti prima ancora di implementare la funzionalità, andando a identificare appunto l'effettivo valore per il cliente e definendo quindi le azioni da intraprendere per ottimizzare il servizio clienti partendo da delle necessità espresse dai clienti stessi.

2. Identificazione del flusso di valore: è costituito dall'insieme delle azioni richieste per condurre un dato prodotto attraverso i tre compiti critici del management: la risoluzione di problemi in tutte le fasi (ideazione, creazione, lancio del prodotto), la gestione delle informazioni dal ricevimento dell'ordine alla consegna e la trasformazione fisica della materia prima in un prodotto finito in mano al cliente¹². Identificare l'intero flusso del valore, come secondo step del pensiero snello, permette di identificare a sua volta grandi quantità di *muda*, cioè di spreco. Generalmente si possono identificare tre tipologie di attività all'interno del flusso:

- a. Attività che creano valore: per esempio l'aggiunta di una pagina FAQ su un sito web istituzionale crea valore al cliente finale perché fornisce informazioni utili e permette di ridurre il tempo necessario per risolvere problemi comuni. Migliora l'esperienza utente e riduce il numero di chiamate o richieste al servizio clienti, semplificando l'interazione tra il cliente e l'azienda.
- b. Attività che non creano valore, ma che non possono essere rimosse vista la tecnologia e gli strumenti attualmente utilizzati (*muda* di Tipo Uno): per esempio disporre di un servizio di backup che entri in funzione qualora il

¹² Womack J.P., Jones D.T., Op. cit., 1996, p.19.

servizio principale non dovesse essere raggiungibile non aggiunge valore diretto al cliente finale per il quale risulta trasparente dal punto di vista funzionale. Tuttavia, è di notevole importanza per garantire la continuità del servizio e prevenire interruzioni. Si tratta quindi di una attività che non può essere rimossa e che con le attuali tecnologie è necessaria per garantire l'integrità.

- c. Attività che non creano valore e possono essere immediatamente eliminate (*muda* di Tipo Due): per esempio, la scrittura di codice duplicato invece di riutilizzare moduli e librerie già esistenti è un'attività inefficiente e può essere eliminata favorendo l'utilizzo di strumenti di gestione del codice sorgente che permettano la condivisione e il riutilizzo di componenti.

È bene notare che l'identificazione del flusso di valore non deve limitarsi alla sola azienda, ma è fondamentale considerare tutti gli attori coinvolti e impattati dalla creazione di un prodotto. Si pensi, per esempio, a un'azienda che dispone di molti fornitori esterni: se non vi è un'opportuna comunicazione tra di loro il rischio di creare grandi quantità di *muda* è molto alto. Si immaginino due fornitori che lavorano rispettivamente sul front-end e sul back-end di un'applicazione web e si prenda come esempio un modulo per il cambio password. Il primo fornitore, responsabile del front-end, sviluppa una funzione di validazione del campo "nuova password", che verifica se il formato della password rispetta la seguente policy aziendale: la password deve avere almeno otto caratteri, includere un carattere maiuscolo, uno minuscolo, un numero e un carattere speciale. Se la password inserita dal cliente non rispetta questi requisiti, un messaggio di errore lo avvisa in tempo reale. Il secondo fornitore, che gestisce il back-end, effettua una seconda validazione della password quando questa viene inviata, prima di salvarla nel database. Tuttavia, in questo caso, la policy di validazione del back-end è diversa da quella implementata nel front-end: la password deve essere lunga almeno dieci caratteri e contenere almeno due numeri. La mancata comunicazione

dei due fornitori ha creato una tipica situazione di *muda* di Tipo Due. Infatti, entrambi i fornitori stanno eseguendo una validazione della password, ma con due policy diverse. Si tratta di uno spreco eliminabile proprio perché sarebbe meglio che la validazione avvenisse in un solo sistema evitando così possibili incoerenze e ridondanze. In più, i team di entrambi i fornitori spendono una determinata quantità di tempo per sviluppare logiche di validazione separate e ridondanti con policy diverse quando sarebbe più efficiente disporre di un'unica policy alla quale si attengano entrambi. In altre parole, in un'ottica di pensiero snello sarebbe stato importante che i due fornitori si fossero parlati chiaramente sin da subito, così da concordare una policy di validazione unica condivisa tra il front-end e il back-end. In questo modo il back-end sarebbe stato responsabile della validazione vera a propria, mentre il front-end, una volta ottenuta la policy concordata dal back-end, si sarebbe occupato solo di fornire un gentile avviso in tempo reale al cliente.

- 3. Far scorrere il flusso:** dopo aver identificato il valore reale di un prodotto e aver eliminato le attività superflue, il passo successivo è garantire che il flusso di lavoro si sviluppi senza interruzioni. L'obiettivo è quello di creare un flusso continuo, evitando di lavorare a comparti stagni, dove ogni fase del processo è isolata dalle altre. È bene che le diverse fasi di produzione siano strettamente collegate e integrate così che il lavoro possa svolgersi senza interruzioni o rallentamenti. È quindi fondamentale che l'azienda adatti le proprie risorse e i propri processi al tipo di prodotto o servizio che sta realizzando invece di seguire rigidamente uno schema di produzione frammentato. Lavorare in un flusso continuo consente di ridurre i tempi di attesa, minimizzare gli sprechi e migliorare la qualità del prodotto finito. Per esempio, un processo produttivo in cui i materiali fluiscono senza dover attendere la disponibilità delle fasi successive riduce il rischio di errori, migliora l'efficienza e aumenta la soddisfazione del cliente finale.

Creare un flusso continuo richiede anche che tutti i dipartimenti e le funzioni aziendali siano allineati e collaborino strettamente, eliminando le barriere tra i vari stadi del processo. Questo tipo di integrazione permette un'esecuzione più

accurata e rapida, garantendo che le attività siano svolte al momento giusto e con le risorse adeguate, facilitando così la consegna del prodotto o servizio nel minor tempo possibile e con la massima qualità. Un flusso produttivo efficiente consente, inoltre, di identificare immediatamente eventuali problemi o colli di bottiglia, permettendo di intervenire prontamente per risolverli e di migliorare ulteriormente il processo. Si pensi, per esempio, all'importanza della fase di testing nello sviluppo di un software. Se si attende che il reparto di sviluppo completi l'intero processo di implementazione prima di procedere ai test, si rischia di dover fare i conti con ritardi significativi. Questi ritardi possono diventare particolarmente gravi nel caso in cui emergano problematiche che richiedano ulteriori interventi correttivi una volta che il processo è terminato. Diversamente, se si adottasse un approccio in cui sviluppo e test procedono in parallelo, creando un flusso costante e integrato, si potrebbero identificare e risolvere gli errori più rapidamente, man mano che le nuove funzionalità vengono implementate. Ciò consente di ridurre il rischio di dover ripetere interi cicli di sviluppo, poiché i difetti vengono corretti immediatamente durante la fase di costruzione del prodotto, anziché alla fine.

- 4. Pull:** una volta creato un flusso continuo più efficiente e meno incline agli errori, si ottiene inevitabilmente anche una notevole riduzione del tempo impiegato per realizzare un determinato prodotto. Questo è un aspetto molto interessante perché permette di non dover disporre di grandi quantità di prodotti pronti per essere venduti a clienti che potrebbero non essere interessati all'acquisto in quel determinato momento. L'azienda rischierebbe di ritrovarsi con magazzini pieni di prodotti finiti e invenduti con costi notevoli di stoccaggio senza un reale ritorno economico. Infatti, l'applicazione di un approccio snello permette di implementare il principio del *pull*: si tratta di far "tirare" dal cliente il prodotto piuttosto che vederselo "spinto" (*push*) dall'azienda che, disponendo di una quantità elevata di prodotti finiti, necessita in tutti i modi di venderne il più possibile, onde evitare di dover smaltire scorte invendute. Ridurre i tempi di produzione e di consegna

dall'ordine di un prodotto fa sì che le aziende possano organizzare al meglio la produzione seguendo più precisamente la reale domanda del mercato. Essendo in grado di allineare la produzione alla domanda del mercato in tempo reale, l'azienda può evitare sovrapproduzione e rispondere tempestivamente a cambiamenti nelle preferenze dei clienti, mantenendo una capacità produttiva agile e adattabile. In un settore come quello del credito al consumo, si può pensare, per esempio, a un'azienda che collabora con retailer online e fisici per offrire piani di finanziamento per l'acquisto di beni come elettrodomestici o dispositivi tecnologici. In un approccio *push* predisporre di un catalogo fisso di offerte di finanziamento con termini e condizioni prestabiliti che rischiano di non adattarsi alle esigenze finanziarie del cliente. Invece, con un approccio *pull*, quando il cliente seleziona un prodotto e richiede informazioni su un possibile finanziamento, l'azienda esegue una valutazione in tempo reale del suo profilo finanziario e genera un piano di pagamento personalizzato (importo del prestito, tasso d'interesse, durata) adattato alle esigenze del cliente e al costo del prodotto. È così possibile personalizzare a misura di cliente le offerte favorendo quindi l'acquisto dei beni con i piani di finanziamento flessibili dell'azienda.

- 5. Perfezione:** i primi quattro principi interagiscono tra di loro in un circolo virtuoso. Non c'è quindi fine al processo di identificazione del *muda* e all'eliminazione di attività che non apportano valore per ridurre sforzi, tempi, costi ed errori per fornire un prodotto di una qualità sempre maggiore e sempre più in linea con quanto desiderato dai clienti¹³. Quindi è bene che si crei un processo di continuo miglioramento, ricercando costantemente di perseguire la perfezione, ottimizzando ogni attività e rimuovendo gli sprechi.

Il lean thinking non si pone quindi come obiettivo quello di fornire degli strumenti pre-stabiliti e specifici per cambiare il lavoro pratico, ma fornisce delle linee guida che sono

¹³ Womack J.P., Jones D.T., Op. cit., 1996, p. 25.

principalmente dei concetti che trovano la loro applicazione nei più diversi modi possibili a seconda del contesto in cui vengono applicati. Risulta molto importante un cambiamento quindi prima di tutto culturale all'interno delle aziende e quindi un cambiamento nel modo in cui si approccia la produzione da cima a fondo. Ruolo fondamentale è quello del leader che non deve limitarsi a impartire ordini e direttive dall'alto, ma deve sposare e plasmare sulla propria realtà aziendale i principi snelli orientando l'operato di tutte le risorse coinvolte, interne ed esterne, verso una nuova cultura del processo di produzione snello. I leader devono essere in grado di mettere in discussione quanto fatto finora e scovare i possibili sprechi e tutti quei punti critici che nei diversi processi necessitano di un'ottimizzazione. Come si è visto nel TPS è molto importante l'apporto dato dai lavoratori stessi che hanno un ruolo cruciale nel rilevare possibili problematiche e nell'elaborare soluzioni anche basate sui suddetti principi. Va da sé che anni di lavoro condotto secondo certi principi e modalità radicate nel tessuto aziendale sono sicuramente un grande ostacolo verso l'adozione di una nuova mentalità. Per questo risulta fondamentale che i leader non si concentrino unicamente sui risultati a breve termine, ma investano sul potenziale di tutti i dipendenti prevedendo corsi di formazione atti allo sviluppo di competenze necessarie per affrontare i problemi in modo autonomo e costruttivo. In questo modo all'interno di ogni gruppo di lavoro ogni membro sarà in grado di dare un apporto qualitativo tendendo a un miglioramento costante in un percorso di crescita personale e dell'intera azienda. Quella che potremmo definire una *leadership lean* pone, inoltre, una grande enfasi sulla collaborazione e sulla comunicazione, in linea con i principi snelli. Per questo risulta necessaria la creazione di un ambiente in cui ogni singolo membro si senta libero di condividere apertamente le proprie idee, preoccupazioni e suggerimenti, incoraggiando una cultura aziendale basata sul rispetto e sulla trasparenza. Inoltre, un leader lean non deve perdere mai di vista il cliente finale ricordando quanto sia di fondamentale importanza concentrarsi su ciò che è realmente di valore per lui. Le scelte e le decisioni vanno quindi prese in un'ottica che pone al centro il cliente con il fine ultimo di dare un impatto positivo sul valore finale del prodotto o del servizio. Mediante la raccolta costante di feedback e mettendo in atto un ascolto continuo del mercato si può disporre di una grande quantità di dati da analizzare per poi tradurli in

decisioni più consapevoli che permetteranno di ridurre gli sprechi, di migliorare la qualità e l'efficienza dei processi. È quindi grazie alla formazione costante del personale, alla responsabilizzazione di tutti gli attori coinvolti nella produzione, alla promozione del lavoro di gruppo e alla costruzione di una cultura di miglioramento continuo che un'azienda può iniziare il proprio percorso verso l'applicazione dei principi lean ottenendo i benefici fin qui descritti. In questo contesto i leader non sono più dei gestori di processi, ma diventano delle vere e proprie guide nel percorso di trasformazione aziendale.

1.2 IL MANIFESTO AGILE

Quanto visto sul lean thinking trova le sue origini nel settore dell'automotive, ma i suoi principi hanno gettato le basi per molti approcci moderni alla gestione dei processi e dei progetti in molteplici ambiti, anche ben lontani dall'industria manifatturiera. Gli esempi proposti finora sui principi snelli riguardano spesso l'ambito IT proprio a rimarcare come l'ottimizzazione delle risorse e la creazione di valore per il cliente siano a tutti gli effetti molto legati anche al mondo dello sviluppo software. In tal senso, il Manifesto Agile, pubblicato nel 2001, rappresenta un punto di svolta per la gestione dei progetti software, portando una nuova visione basata su flessibilità, collaborazione e capacità di rispondere rapidamente ai cambiamenti. Questo manifesto fu redatto da diciassette esperti di sviluppo software, tra cui Kent Beck, Martin Fowler, Robert C. Martin e Alistair Cockburn, che si incontrarono presso il resort sciistico Snowbird, nello Utah, per trovare un terreno comune tra i vari approcci leggeri già esistenti, come Extreme Programming (XP), SCRUM, Crystal, ecc. L'incontro nacque dalla necessità di trovare un'alternativa ai modelli tradizionali di sviluppo software, considerati troppo rigidi e basati su processi pesanti e una documentazione eccessiva. Durante i due giorni di discussione, vennero definiti quattro valori fondamentali, che mettevano al centro la collaborazione tra individui, il software funzionante, la collaborazione con i clienti e la flessibilità nel rispondere ai cambiamenti, valori che cambiarono radicalmente il modo di sviluppare software.

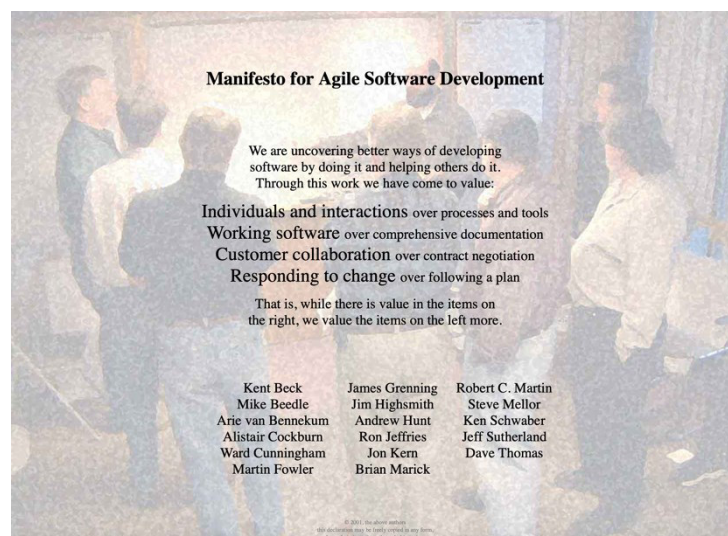


Figura 3 - Il Manifesto Agile. Fonte: <https://agilemanifesto.org/> (ultimo accesso: 30/10/2024)

Come emerse dalle parole di Alistair Cockburn, inizialmente era difficile pensare che un gruppo così eterogeneo potesse trovare un accordo su qualcosa di sostanziale. Tuttavia, la sintonia che si creò tra i partecipanti portò alla formulazione del Manifesto, che sancì un nuovo approccio allo sviluppo software, fondato su fiducia, rispetto reciproco e modelli organizzativi incentrati sulle persone e sulla collaborazione. Questo incontro segnò la nascita dell'Agile Alliance, un gruppo di pensatori indipendenti e, talvolta, concorrenti, uniti dalla convinzione che lo sviluppo software potesse essere gestito in modo più flessibile ed efficace rispetto ai modelli tradizionali¹⁴.

Il Manifesto Agile presenta quattro valori fondamentali che sono alla base della metodologia e della filosofia Agile. Viene messo in discussione il tradizionale approccio allo sviluppo software considerato troppo rigido e formale, in favore di un approccio che prenda in considerazione ciò che è realmente importante per la buona riuscita di un progetto e per la soddisfazione del cliente. Questi valori sono descritti qui di seguito.

- 1. Individui e interazioni più che processi e strumenti:** strumenti e processi sono utili, però sono le persone che con la loro collaborazione e comunicazione possono fare la differenza nell'analisi e nella risoluzione di problemi per permettere uno sviluppo che sia efficace e possa portare valore ai clienti. Viene quindi considerato molto importante il lavoro di squadra e la comunicazione all'interno del team di lavoro piuttosto che l'attenersi a processi rigidi e strumenti limitanti.
- 2. Software funzionante più che documentazione esaustiva:** produrre un software che sia ben funzionante e che non necessiti di documentazione dettagliata e voluminosa è indicazione di aver svolto un ottimo lavoro. Rispondere meglio ai bisogni immediati dei clienti fornendo un software che sia funzionante e che non

¹⁴ Highsmith J., "History: The Agile Manifesto", in *The Agile Manifesto*, Agile Alliance, 2001, disponibile online all'indirizzo <https://agilemanifesto.org/history.html> (ultimo accesso in data 04/10/2024).

richieda di consultare pagine e pagine di manuali per poterci interagire va preferito alla documentazione, riducendo al minimo le formalità non necessarie.

- 3. Collaborazione con il cliente più che negoziazione dei contratti:** è importante favorire una collaborazione costante con il cliente piuttosto che limitarsi alla negoziazione di un contratto a inizio progetto. Il coinvolgimento del cliente in tutto il ciclo di produzione risulta molto importante perché si ha la possibilità di raccogliere feedback e i requisiti possono essere costantemente adeguati al cambiamento delle esigenze. Attraverso la promozione di una collaborazione continua e aperta con il cliente, non limitata da rigidi vincoli contrattuali, si possono raggiungere risultati maggiormente in linea con le sue aspettative e bisogni reali.

- 4. Rispondere al cambiamento più che seguire un piano:** l'ambito dello sviluppo software è molto dinamico e riuscire a stare al passo con la tecnologia in costante evoluzione implica una molteplicità di cambiamenti costanti. Diventa quindi difficile pensare di poter seguire un piano iniziale che, una volta definito, non accetta modifiche in corso d'opera. In tal senso, il suggerimento è quello di essere pronti ad accogliere e a rispondere ai cambiamenti facendo riferimento a un piano base che sia però flessibile per potersi adattare velocemente alle nuove esigenze o ai cambiamenti necessari per poter restare al passo con lo sviluppo tecnologico.

Questi quattro valori principali presenti nel Manifesto Agile trovano al di sotto ben dodici principi che hanno guidato i diciassette firmatari nella loro formulazione. Di seguito vengono presentati i dodici principi con una breve spiegazione di ognuno¹⁵.

¹⁵ Beck K. et al., "I principi sottostanti al Manifesto Agile", in *The Agile Manifesto*, Agile Alliance, 2001, disponibile online all'indirizzo <https://agilemanifesto.org/iso/it/principles.html> (ultimo accesso in data 04/10/2024).

- 1. La nostra massima priorità è soddisfare il cliente rilasciando software di valore, fin da subito e in maniera continua:** quindi l'obiettivo primario è quello di consegnare software funzionante e che sia di valore per il cliente, garantendo dei rilasci che siano rapidi e continui.
- 2. Accogliamo i cambiamenti nei requisiti, anche a stadi avanzati dello sviluppo. I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente:** qui sta il cuore della flessibilità delle pianificazioni di cui si parlava prima. Essere in grado di accogliere cambiamenti in corso d'opera permette di fornire soluzioni che siano sempre aggiornate e pertinenti, in linea con i cambiamenti dei requisiti del cliente.
- 3. Consegniamo frequentemente software funzionante, con cadenza variabile da un paio di settimane a un paio di mesi, preferendo i periodi brevi:** un rilascio frequente e in brevi periodi permette di ottenere sin da subito feedback preziosi dal cliente che possono portare a delle piccole correzioni e variazioni gestibili molto più comodamente rispetto a una lista molto maggiore di segnalazioni a sviluppo concluso dell'intero software.
- 4. Committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto:** il team di sviluppo può avere dei dubbi sui requisiti del cliente, può incontrare problematiche non ipotizzate in fase di analisi, insomma può avere la necessità di chiarirsi con il committente. Piuttosto che bloccare l'avanzamento del progetto per ogni dubbio che può presentarsi, è bene promuovere un dialogo costante per l'intera durata del progetto tra il cliente e gli sviluppatori, garantendo un avanzamento fluido.
- 5. Fondiamo i progetti su individui motivati. Diamo loro l'ambiente e il supporto di cui hanno bisogno e confidiamo nella loro capacità di portare il lavoro a termine:** è importante che i team siano composti da persone motivate e che allo

stesso tempo dispongano di tutti gli strumenti necessari per compiere al meglio il loro lavoro. Creare un ambiente adatto al lavoro che deve essere svolto, a sua volta, favorisce la motivazione stessa degli individui oltre che la buona riuscita di un progetto.

- 6. Una conversazione faccia a faccia è il modo più efficiente e più efficace per comunicare con il team e all'interno del team:** questo principio è molto attuale, in un'epoca di crescente utilizzo di strumenti digitali per la comunicazione, come e-mail o piattaforme come Microsoft Teams. Infatti, sottolinea l'importanza della comunicazione diretta, dove il tono di voce, le espressioni facciali e la risposta immediata evitano malintesi e ritardi, problemi che possono sorgere con messaggi testuali non letti, fraintesi, o peggio ancora mai ricevuti. La comunicazione faccia a faccia, anche virtuale, permette maggiore chiarezza e comprensione immediata.
- 7. Il software funzionante è il principale metro di misura di progresso:** consegnare del software funzionante indica un reale avanzamento nel progetto, al posto di una grande quantità di documentazione che ne riporti lo stato di avanzamento. Per il cliente disporre di nuovo software funzionante significa avere del valore aggiunto.
- 8. I processi agili promuovono uno sviluppo sostenibile. Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante:** è molto importante monitorare il carico di lavoro, avendo cura di evitare sovraccarichi che metterebbero in difficoltà il team, riducendo drasticamente la qualità del lavoro svolto a lungo termine.
- 9. La continua attenzione all'eccellenza tecnica e alla buona progettazione esaltano l'agilità:** essere agili nel rispondere ai cambiamenti futuri implica di disporre di una progettazione tecnica e dello sviluppo di codice di qualità, essendo più

semplice da adattare e da modificare perché più opportunamente organizzato e pulito.

10. La semplicità - l'arte di massimizzare la quantità di lavoro non svolto - è essenziale: riuscire a ridurre al minimo lo spreco è fondamentale, sarà quindi opportuno semplificare al massimo il lavoro per concentrarsi al meglio solo su quanto è in grado di apportare un reale valore al cliente.

11. Le architetture, i requisiti e la progettazione migliori emergono da team che si auto-organizzano: quando i team hanno la possibilità di prendere decisioni collettive sul modo di lavorare risultano spesso più produttivi e innovativi. L'auto-organizzazione consente ai membri del team di collaborare liberamente, di adattarsi alle sfide e di trovare soluzioni in modo collaborativo. È essenziale che i manager garantiscano un'adeguata autonomia e fiducia, creando un ambiente di lavoro che favorisca l'iniziativa e la responsabilità condivisa.

12. A intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza: quest'ultimo principio è fondamentale per promuovere il miglioramento continuo. Periodicamente, il team riflette sull'andamento del lavoro, analizzando cosa ha funzionato bene e cosa invece può essere migliorato. Questi momenti di retrospettiva permettono al team di evidenziare i punti di forza e di debolezza delle metodologie adottate, consentendo di affinare il proprio processo di lavoro. Imparando dagli errori e ottimizzando le buone pratiche, il team è in grado di adattarsi continuamente e migliorare l'efficacia del proprio operato. Questa pratica contribuisce a creare un ciclo virtuoso di miglioramento e di crescita professionale.

È interessante notare che, come per i principi lean, anche i principi Agile non si sono limitati all'ambito dello sviluppo software, ma vengono adottati in ambiti diversi da quello in cui sono nati. Ne è un esempio, quello che viene definito *l'Agile Marketing*.

Nasce nel 2012 dalla necessità di migliorare l'approccio del marketing tradizionale in un mondo in costante cambiamento¹⁶. La comunicazione e promozione di un brand si basa su una strategia che è opportunamente realizzata con un'attenta analisi del mercato, dei competitors e dei potenziali clienti, basandosi sugli obiettivi che si desidera raggiungere, con una visione a lungo termine. Tale strategia è bene che sia una base di riferimento, ma allo stesso tempo deve garantire flessibilità e adattabilità nel corso della sua applicazione. Come nello sviluppo software, anche nell'ambito del marketing, il cliente finale assume un ruolo molto importante. Si deve comprendere come la comunicazione di un determinato brand possa essere vista come un valore aggiunto per il potenziale cliente. In caso contrario, tanto come per un software che non rispetta i requisiti, la comunicazione risulterebbe completamente inutile e quindi un vero e proprio spreco di risorse. Una strategia include anche una pianificazione dettagliata dei canali di comunicazione che si desidera utilizzare oltre che alla programmazione dei contenuti da pubblicare. Si pensi, per esempio, a un post su un canale social con cadenza settimanale. Tuttavia, non è detto che tutte le uscite previste nella programmazione siano per forza di successo. Applicare un approccio Agile in tal senso vuol dire dotarsi dell'opportuna flessibilità e rapidità di intervento che permettono di correggere il tiro in corso d'opera, basandosi su dati reali e analizzando gli indicatori chiave di prestazione (KPI). In generale quindi si applica un approccio iterativo che permette di migliorare la strategia man mano che la si applica. In questo modo, diventa possibile rispondere in modo rapido ed efficace ai cambiamenti degli interessi dei consumatori o realizzare contenuti legati a eventi di attualità che non erano stati previsti inizialmente nella pianificazione, insomma di essere più dinamici in un ambiente dinamico. In più, il lavoro proattivo e orientato al risultato, con un costante desiderio di miglioramento da parte dell'intero team di marketing, permette di raggiungere più facilmente risultati via via sempre migliori, massimizzando l'impatto della comunicazione del brand.

Il fatto che i principi fondamentali della metodologia Agile possano trovare applicazione in molteplici ambiti è un chiaro segnale di quanto Agile non sia una serie di strumenti o

¹⁶ Errico P., *Agile Marketing: Fai felici i tuoi clienti in metà tempo, con risorse limitate, con team più allineati e motivati*, Edizioni LSWR, Milano, 2020.

di metodologie da applicare tout court. Prima di poter essere applicate, queste idee richiedono un vero e proprio cambiamento nel modo di pensare da parte dell'intera azienda. Si parla di una vera e propria trasformazione culturale che non può essere messa in atto con una semplice adozione di nuove tecniche o di metodologie. Riuscire a superare i metodi rigidi applicati in passato favorendo un approccio più flessibile e che ponga maggiore importanza alla comunicazione e alla collaborazione all'interno del team è tutto fuor che di semplice raggiungimento. Entrare nell'ottica di un miglioramento costante grazie a una continua revisione del lavoro svolto implica una completa riorganizzazione delle attività e anche la necessità di creare dei momenti *ad hoc* per favorire la riflessione da parte di tutto il personale dell'azienda. Inoltre, non è sufficiente concentrarsi solo internamente, ma nella maggior parte dei casi in cui le aziende dispongono di un gran numero di fornitori esterni è opportuno che questi siano a loro volta coinvolti nel cambiamento del modo di pensare e di approcciare la produzione. Diventa molto più importante il lavoro di squadra, la responsabilità e l'opinione di ogni singola persona piuttosto che concentrarsi esclusivamente sul lavoro in quanto tale. In più, anche l'ambiente lavorativo necessita di un cambiamento, in modo tale da essere adattato per accogliere al meglio i team che sono in costante confronto, coinvolti in modo attivo in ogni fase di un progetto, e che dialogano maggiormente con tutti gli attori a cui si richiede una presenza costante nello svolgimento dei lavori. Vi è quindi la necessità di creare degli spazi adeguati e dedicati che non chiudano in reparti stagni e in uffici divisi tutte le persone appartenenti a un team e impegnate in un determinato progetto. Anche i ruoli cambiano e i manager non sono più unicamente responsabili del controllo, ma diventano coloro che hanno l'obiettivo di veicolare e di facilitare il lavoro dei team favorendone la crescita, permettendo loro di prendere decisioni e di adottare misure correttive in completa autonomia. È quindi molto chiaro come tutto questo non possa essere applicato concretamente se prima non si comprenda a fondo il cambio di mentalità. Non è possibile agire Agile se prima non si pensa Agile.

1.3 UNA QUESTIONE DI MENTALITÀ: LEAN AND AGILE MINDSET

Il modo di pensare e gli atteggiamenti di una persona o di un gruppo di persone dipendono dalla mentalità. Il modo in cui ci si pone di fronte alle sfide quotidiane, la capacità di prendere delle decisioni autonomamente e più in generale il nostro modo di interpretare il mondo dipende da una serie di convinzioni profonde insite in ogni essere umano. Va da sé che vi è un forte legame tra la mentalità e la capacità di gestire problemi complessi, di lavorare con altre persone che hanno a loro volta una mentalità che non sempre è simile alla nostra, e la capacità di reagire ai cambiamenti. La mentalità si forma nel tempo in base all'esperienza vissuta, all'educazione ricevuta, all'interazione sociale e quindi può modificarsi ed evolversi sulla base di nuove interazioni, informazioni e prospettive. Tuttavia, tale cambiamento non è sempre visto come un'opportunità e non sempre vi sono persone che sposano facilmente nuovi approcci o che pensano di essere in grado di adottarli. Da un lato, sono poco inclini al rischio di lasciare un qualcosa di conosciuto in favore di un nuovo percorso potenzialmente insicuro, dall'altro non sono completamente convinti che attraverso il ragionamento e l'esercitazione si possa realmente cambiare la propria mentalità.

Durante l'esperienza di tesi in azienda presso Santander Consumer Bank Italia¹⁷ – presentata nel terzo capitolo – è capitato spesso di confrontarsi con alcuni colleghi sulla tematica. Si potrebbe pensare che le figure più giovani possano avere una mentalità più incline al cambiamento, ma è interessante notare che non vi sia una particolare correlazione tra anni di esperienza lavorativa e propensione al cambiamento. In generale, per alcuni l'idea di sperimentare un modo diverso di gestione dei progetti era considerata completamente assurda e quasi impossibile da applicare in azienda; per altri, l'interesse a cercare nuovi modi di approcciare il lavoro e le sfide quotidiane, inevitabili in un ambiente in costante cambiamento come quello del credito al consumo, era la motivazione a fare ogni giorno una riflessione su ciò che era andato bene e ciò che poteva essere gestito diversamente, consapevoli che si possa sempre migliorare.

¹⁷ Santander Consumer Bank Italia è una banca specializzata nel credito al consumo, offrendo soluzioni finanziarie come prestiti personali, conti deposito e carte di credito. Nel sottocapitolo 3.1 è possibile approfondire ulteriormente.

In tal senso, è possibile introdurre i concetti di *fixed mindset* e di *growth mindset* definiti da Carol Dweck (2006)¹⁸, psicologa e docente alla Stanford University, che ci permettono di capire come le persone con diversa mentalità siano influenzate nel loro modo di affrontare le sfide e i cambiamenti. Una persona con una mentalità fissa crede che le sue qualità siano scolpite nella roccia, necessitando quindi di provare costantemente a se stessa e agli altri di possederle. Inoltre, tende a evitare le sfide in quanto teme un fallimento. Invece, una persona con una mentalità di crescita è convinta che le proprie qualità possano migliorare nel tempo a fronte di uno sforzo e vede quindi i fallimenti come delle opportunità per fare meglio la volta successiva. Si pensi, per esempio, a come può essere vissuto diversamente un tremendo lunedì mattina di rientro in ufficio dopo una settimana di ferie da Maria (nome di fantasia), *fixed mindset*, e da Alessandro (nome di fantasia), *growth mindset*. All'apertura del computer e del client di posta compaiono circa 340 e-mail da leggere; a metà mattinata la manager comunica sia a Maria, sia ad Alessandro che i loro progetti sono in ritardo e che è profondamente delusa della loro gestione. Entrambi contattano i loro rispettivi colleghi preferiti per ricevere un po' di conforto, ma scoprono che sono in smart working e il loro messaggio viene visualizzato senza risposta per l'intera giornata. Molto probabilmente Maria trovandosi di fronte a delle difficoltà tenderà a confermare quelle che crede siano le sue limitazioni. Quindi di fronte alla notevole quantità di e-mail non lette e alla spiacevole notizia ricevuta dalla manager reagirà istintivamente con una profonda frustrazione. Non solo, tenderà a sentirsi sopraffatta e prenderà la critica della manager in modo personale, interpretandola come una prova del suo fallimento. In più, Maria si sentirà isolata per via della non risposta del suo collega preferito, rafforzando la sua convinzione di non essere sufficientemente capace. È quindi probabile che la reazione di Maria la porterà a evitare ulteriori sfide, con la preoccupazione di un possibile nuovo fallimento, concentrandosi su come proteggersi piuttosto che su come risolvere la problematica.

La reazione di Alessandro invece è ben diversa. La mole di e-mail è una sfida da affrontare e da superare, per esempio, andando a capire come impostare delle regole che smistino

¹⁸ Dweck C. S., *Mindset: The New Psychology of Success*, Random House, New York, 2006, pp. 6-7.

e filtrino automaticamente la posta in ingresso giorno dopo giorno così da padroneggiare meglio tutte le comunicazioni. Non rimane demoralizzato dalla critica della manager, ma è motivato a riprendere in mano il controllo della situazione e a comprendere come possa migliorare la gestione del progetto per ripartire alla grande. Non penserà quindi di non essere all'altezza, ma di poter imparare da ciò che è stato per scovare un approccio diverso che gli consentirà di dare una svolta al progetto. Non rimane demoralizzato dalla mancata risposta del collega preferito, cerca di aprirsi con nuove persone ed è sicuro che il giorno dopo, una volta tornato in ufficio, potrà comunque contare dell'appoggio del suo collega. La prospettiva di Alessandro è quindi completamente diversa e orientata alla risoluzione dei problemi e al miglioramento personale, trovando nelle difficoltà delle opportunità di crescita.

È bene però non pensare che allora Maria non sarà in grado di cambiare il proprio approccio. Infatti, la sua mentalità non è fissa di natura, ma lo è diventata nel tempo a causa del proprio vissuto e delle esperienze che ha man mano accumulato. La mentalità è frutto di un processo dinamico e quindi può essere cambiata con delle opportune esperienze, una specifica formazione e una maggiore consapevolezza. Prima di tutto è bene che le persone siano costantemente stimolate per favorire un'apertura all'apprendimento, creando dei momenti dedicati che favoriscano il dialogo e il confronto tra persone con mentalità diversa. A volte, è proprio scoprendo il diverso modo di porsi di altre persone che consideriamo al nostro stesso livello in determinate situazioni che stimola la curiosità e il desiderio di provare un altro approccio. Un ambiente lavorativo in cui la collaborazione e il feedback costruttivo vengono incoraggiati e in cui vi è la promozione di una cultura di sperimentazione può aiutare notevolmente le persone che hanno una mentalità fissa ad aprirsi verso una mentalità di crescita, favorevole al cambiamento. È bene che i colleghi e i manager promuovano e rimarchino i progressi piuttosto che sottolineare costantemente gli errori commessi, in modo tale da incentivare tutti i lavoratori a riconsiderare il proprio modo di affrontare le sfide, imparando dagli errori, trovando nuove strade creative per risolverli e abbracciare una mentalità di crescita continua. Ovviamente si tratta di un processo tutt'altro che semplice e tutt'altro che veloce, soprattutto in un contesto aziendale in cui per anni sono stati adottati certi approcci e in quelle

realtà in cui il numero dei lavoratori è particolarmente elevato. Sicuramente è molto più semplice che una startup adotti un approccio che favorisca una mentalità di crescita e che questa sia la filosofia iniziale, prima ancora di avviare l'attività, ma ciò non rende impossibile un cambiamento in tal senso anche in aziende che operano da un numero di anni molto più ampio. Si tratta di un vero e proprio processo di transizione durante il quale avvalersi di figure esterne esperte e formate a supporto continuo e costante di tutte le persone coinvolte. E poi non bisogna dimenticarsi che non si tratta di un periodo circoscritto alla transizione, ma è bene essere consapevoli che l'apprendimento non deve trovare una fine e che le capacità e le conoscenze di ognuno sono in costante raffinamento e possono quindi migliorare, esperienza dopo esperienza, in un circolo virtuoso infinito.

Quando si parla di *lean and agile mindset* si fa riferimento proprio a una mentalità di crescita. Come visto precedentemente, entrambi i modelli fanno riferimento a principi e convinzioni come l'importanza di sapersi adattare rapidamente ai cambiamenti, di accettare i fallimenti e di imparare dagli errori oltre che di impegnarsi in un miglioramento continuo. Da un lato, il lean thinking mira a eliminare gli sprechi e a ottimizzare ogni fase del processo produttivo così da aggiungere ai prodotti valore per il cliente finale. Il tutto avviene in un processo continuo in cui ci si concentra sull'identificazione delle problematiche così da disporre costantemente di soluzioni atte alla loro risoluzione. Questo viene fatto in un'ottica a lungo termine e porterà un notevole vantaggio in termini di risorse risparmiate e di flessibilità nella produzione. Dall'altro lato, Agile si basa su un processo iterativo che permette, basandosi sui continui feedback dei clienti, di produrre dei prodotti che siano in linea il più possibile con la richiesta, in breve tempo, basandosi su cicli di sviluppo e di test molto rapidi, aggiungendo valore al termine di ogni ciclo. La combinazione dei due permette quindi a un'azienda di raggiungere ottime capacità di problem-solving e di reagire molto rapidamente ai cambiamenti del mercato e delle normative, oltre che di aderire più precisamente alla mission aziendale. Si pensi, per esempio, ai notevoli vantaggi che una mentalità lean e Agile può apportare alle aziende nel settore bancario e del credito al consumo. Prima di tutto si potrebbe ottimizzare la maggior parte dei processi interni, favorendo l'implementazione di automazioni che

semplifichino i processi stessi che sono a volte molto macchinosi e spesso svolti manualmente dagli operatori, riducendo così anche l'alto rischio di introduzione di errori. Successivamente, si potrebbe migliorare notevolmente l'esperienza del cliente finale rilasciando software applicativo in costante aggiornamento, al passo con le nuove tecnologie e che garantisca quindi la massima sicurezza e disponibilità dei servizi digitali offerti. Per concludere, deve essere ben chiaro che queste sono solo alcune delle auspicabili conseguenze a seguito di quello che è anzitutto un cambio di mentalità a livello aziendale, dal primo all'ultimo dipendente. Risulta cruciale la formazione, lo svolgimento di sessioni e di attività *ad hoc* per migliorare l'approccio di ogni persona alle sfide, la fermezza di gestire situazioni complesse e la capacità di lavorare in gruppo eterogeneo. È fondamentale migliorare e stimolare la comunicazione tra le persone in modo tale che questa sia la più libera, la più aperta e la più chiara possibile, onde evitare fraintendimenti o rischiare di non sentire la voce di tutti, sovrastata da chi presenta un carattere più forte. Grazie alla collaborazione e al rispetto delle idee di ognuno, i gruppi di lavoro possono prendere decisioni autonomamente e in modo più consapevole, in un clima più sereno che favorisce il desiderio di innovazione e di miglioramento incessante, adottando quindi una nuova mentalità, il lean and agile mindset.

2. AGIRE AGILE

Dal momento in cui il pensiero Agile inizia a diffondersi in un'azienda e il cambio di mentalità ha dato il via a una trasformazione sostanziale del modo in cui vengono svolte le attività quotidiane, entrano in gioco una serie di strumenti e di metodologie che aiutano la messa in pratica dei principi finora esposti. È importante rimarcare come questi siano da considerarsi unicamente dei mezzi a supporto dell'adozione e della successiva trasformazione di un'azienda verso un approccio lean e Agile. Ciò che sarà presentato all'interno di questo capitolo non è quindi il punto di inizio che permette di compiere una trasformazione aziendale. Il mero utilizzo di strumenti non sta a indicare che un'azienda abbia un reale approccio snello e Agile. Tra l'altro, proprio in un'ottica di flessibilità, non esiste un unico strumento o un'unica metodologia che possa adattarsi alle diverse realtà aziendali. Nelle prossime pagine, verrà analizzato più nel dettaglio il framework Scrum, il quale fornisce una struttura semplice e funzionale per la gestione del lavoro in team agili. Grazie all'identificazione di ruoli trasversali, alla presenza di eventi ricorrenti che scandiscono l'avanzamento dei progetti e agli artefatti chiave, Scrum permette ai diversi dipartimenti aziendali di creare dei flussi di lavoro che sono trasparenti e ottimizzati, rispondendo rapidamente alle esigenze dei clienti e della produzione. Tuttavia, esistono altri strumenti che possono essere adottati per implementare i principi Agile e del lean thinking. Ognuno di questi offre diverse modalità di gestione del lavoro, presentando punti di forza specifici che li rendono adatti a diversi contesti aziendali. Generalmente, è un adeguato mix opportunamente plasmato sulle reali necessità di una determinata azienda che garantisce un'ottima implementazione, un po' come è stato per il cosiddetto *Modello Spotify*.

Oltre a Scrum, una delle metodologie più flessibili e spesso utilizzata è il Kanban, termine già incontrato precedentemente quando si è presentato il Toyota Production System, contesto nel quale trova la sua origine, ma dal quale ha avuto modo di svilupparsi e adattarsi anche al mondo dello sviluppo software e della gestione dei progetti. Attualmente consiste in un metodo che permette di visualizzare più comodamente il flusso di lavoro

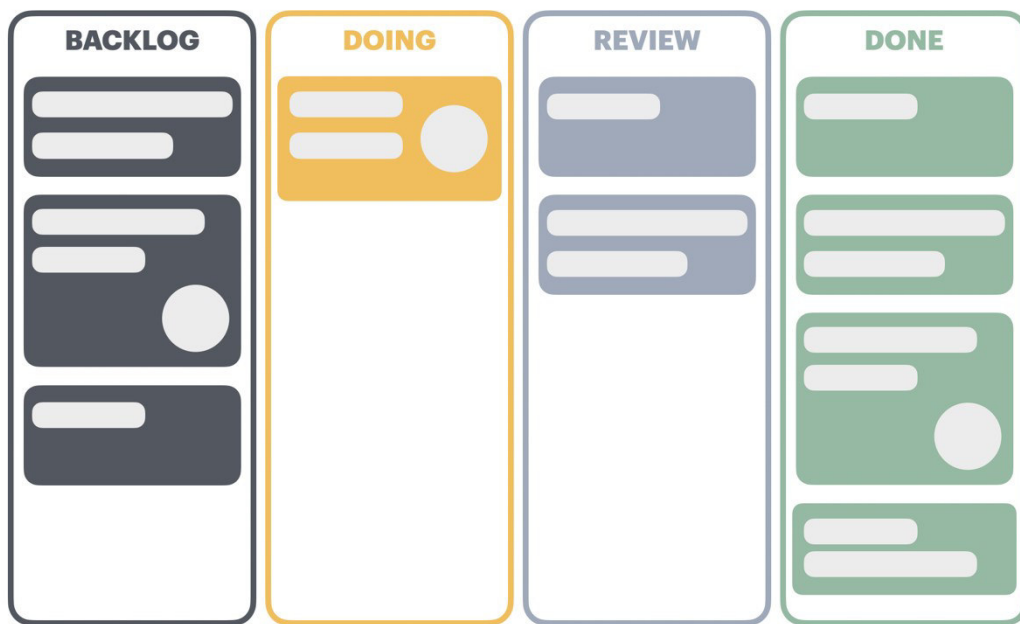


Figura 4 - Esempio di Kanban Board.

avendo così la possibilità di monitorare in tempo reale l'avanzamento delle attività su quella che viene chiamata la *Kanban Board*. Si tratta di una lavagna suddivisa in diverse colonne che rappresentano le varie fasi di un progetto. In Figura 4 vi è un esempio di una Kanban Board a quattro colonne con le fasi: *backlog*, *doing*, *review* e *done*. In ogni colonna si posizionano le diverse attività in base alla fase in cui si trovano in quel determinato momento. Il numero di colonne non è fisso e può essere scelto in base al numero di fasi che si intende rappresentare, ma in generale i punti di partenza e di arrivo (prima e ultima colonna) rappresentano rispettivamente l'elenco di tutte le attività che si desidera realizzare e l'elenco delle attività portate a termine. Un'altra colonna generalmente presente è quella delle attività che sono in corso di lavorazione. È molto importante perché permette di accorgersi a colpo d'occhio del carico di lavoro e quindi del numero di attività che vengono svolte contemporaneamente, evitando quindi di raggiungere situazioni di sovraccarico. Non vi sono iterazioni, ma quello che si crea è un flusso continuo in cui i task vengono ordinati a seconda della loro priorità ed elaborati man mano che le risorse sono disponibili. Questo strumento può essere molto utile quando vi è la necessità di gestire molteplici task avendo sempre una visione chiara di ciò che si sta facendo e garantendo che a fronte di un cambio di priorità, anche frequente, il flusso di lavoro non si interrompa, ma continui a fluire opportunamente essendo in grado di assorbire

rapidamente nuove attività con priorità elevata. Si può utilizzare una vera e propria bacheca posizionata in un angolo dell'ufficio sulla quale appiccicare dei foglietti di carta con sopra scritta l'attività da fare e le rispettive persone coinvolte, ma esistono anche molteplici software con i quali crearne una rappresentazione digitale e accedere a una pletora di funzionalità molto comode. Un esempio di applicativo che può essere utilizzato è Trello¹⁹. Ogni progetto viene rappresentato attraverso una bacheca sulla quale si possono creare diverse liste (le colonne della Kanban Board) che vengono man mano popolate dalle attività che sono rappresentate da schede. Queste possono essere spostate da una lista all'altra man mano che cambiano di stato. Ogni scheda può contenere vari dettagli come descrizioni, checklist, date di scadenza e può essere assegnata ai membri del team appartenenti alla bacheca. Inoltre, vi è la possibilità di allegare dei documenti e di lasciare dei commenti in modo tale che vi sia una costante comunicazione e uno scambio di informazioni all'interno del team.

¹⁹ Trello, disponibile online all'indirizzo <https://trello.com> (ultimo accesso in data 10/10/2024).

2.1 IL FRAMEWORK SCRUM

Troviamo per la prima volta il termine Scrum all'interno del paper *The New New Product Development Game* pubblicato nel 1986 da Hirotaka Takeuchi e Ikujiro Nonaka. Lo sviluppo di nuovi prodotti viene paragonato al gioco del rugby dove le squadre si muovono in modo unito lungo il campo per raggiungere la meta piuttosto che in modo sequenziale come avviene in una staffetta. Il termine Scrum può quindi essere tradotto come mischia, cioè quella situazione di gioco nel rugby in cui, a seguito di un'infrazione minore o di un'interruzione, il gioco riparte con una lotta per il possesso del pallone. Gli autori osservano come le aziende giapponesi di successo avevano adottato un metodo di sviluppo dei prodotti maggiormente flessibile, dinamico e inter-funzionale che permetteva ai team di lavorare contemporaneamente a diverse fasi dello sviluppo. Successivamente, nei primi anni Novanta, Ken Schwaber e Jeff Sutherland (anche loro saranno tra i firmatari nel 2001 del Manifesto Agile) utilizzarono nuovamente il termine Scrum per descrivere un diverso approccio per lo sviluppo del software che fosse più flessibile e iterativo. Il loro desiderio era quello di disporre di una metodologia che permettesse di adattarsi rapidamente ai cambiamenti e di gestire anche le situazioni più complesse con agilità, fornendo al cliente finale successive aggiunte di valore al software sviluppato. Sarà nel 1995 che Schwaber e Sutherland presenteranno per la prima volta Scrum alla conferenza OOPSLA (Object-Oriented Programming, Systems, Languages & Applications) come un framework che avrebbe cambiato radicalmente il modo di gestire i progetti, non solo in ambito software, ma in qualsiasi settore industriale. Per permettere una maggiore diffusione del framework, ma soprattutto una sua migliore comprensione globale, nel 2010 crearono una guida²⁰ contenente la seguente definizione di Scrum:

“Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems”.

²⁰ Schwaber K., Sutherland J., *The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game*, 2020, disponibile online all'indirizzo <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf> (ultimo accesso in data 10/10/2024).

La guida è stata costantemente aggiornata ed è a oggi disponibile nella sua versione aggiornata al 2020. Il framework viene appositamente definito incompleto dai suoi autori in quanto fornisce una base sulla quale ogni team ha la possibilità di muoversi autonomamente per adattarlo al meglio alle proprie caratteristiche e ai contesti in cui opera.



Figura 5 - I pilastri di Scrum

Come già detto più volte, non si tratta di disporre di una lista di regole propinate attraverso una guida e che una volta seguite alla lettera semplificano la complessità del contesto produttivo, ma Scrum è un framework che si fonda sull'empirismo e sul pensiero snello. Grazie all'osservazione e all'esperienza è possibile prendere decisioni andando a ridurre gli sprechi concentrandosi sull'essenziale. La guida mette in risalto tre pilastri fondanti:

- 1. Trasparenza:** è fondamentale che tutti gli aspetti del processo di lavoro siano ben visibili e molto chiari a tutti i soggetti coinvolti in un progetto, siano essi all'interno del team di lavoro, siano essi esterni (i cosiddetti *stakeholder*). Tutti devono poter prendere decisioni che siano informate e che si basino su una descrizione chiara e trasparente sullo stato di avanzamento del progetto. Basarsi su informazioni incomplete potrebbe portare a decisioni sbagliate aumentando la possibilità di incorrere in problematiche che potrebbero causare il fallimento del progetto.

- 2. Ispezione:** monitorare costantemente i progressi verso il raggiungimento degli obiettivi di un progetto è importante perché permette di identificare possibili disallineamenti. Anticipare la comparsa di problemi permette di attuare azioni correttive di piccola entità proprio perché applicate prima ancora che questi abbiano impatti notevoli.

- 3. Adattamento:** è bene che il team adotti immediatamente delle contromisure correggendo il proprio approccio a seguito di quanto emerso dall'ispezione. Se vengono identificate delle deviazioni o delle problematiche che potrebbero compromettere il raggiungimento degli obiettivi del progetto è cruciale che il team intervenga tempestivamente per minimizzare le conseguenze negative, dato che una mancata risoluzione comporterebbe un aumento degli ostacoli che rischiano di immobilizzare il fluido avanzamento del progetto.

Inoltre, i componenti dei team che utilizzano il framework Scrum, sono invitati a vivere e a comprendere cinque valori che permettono loro di assumere l'adeguato atteggiamento mentale necessario per aderire alla cultura collaborativa e trasparente del team. I cinque valori sono presentati qui di seguito.

- 1. Impegno:** ogni membro del team si impegna nei confronti dell'intero team a raggiungere i propri obiettivi contribuendo al successo comune. Ogni membro adatta il lavoro alla propria capacità di soddisfare le aspettative degli altri e non accetta carichi di lavoro che non può gestire. In questo modo si crea un rapporto di fiducia all'interno del team che permette di superare con successo anche gli ostacoli più insidiosi e di portare a termine progetti molto complessi.

- 2. Focus:** è molto importante che i team possano lavorare con la massima concentrazione sugli obiettivi e sulle priorità stabilite. Risulta quindi necessario un

ambiente in cui non vi siano potenziali distrazioni e l'utilizzo di strumenti efficienti, software e non, che aiutino la focalizzazione del team sul lavoro da svolgere.

3. **Apertura:** non è ammissibile che alcuni componenti del team non si sentano liberi di esprimere la propria opinione o di esternalizzare le proprie perplessità. Va promossa una comunicazione trasparente all'interno del team e ogni componente deve essere aperto a ricevere e a fornire critiche costruttive. In questo modo è possibile prevenire l'insorgere di possibili conflitti e preservare un clima proattivo e di fiducia, migliorando la collaborazione e la qualità del lavoro.
4. **Rispetto:** ogni membro del team gode del rispetto degli altri in quanto individuo competente e capace. In un team eterogeneo in cui ognuno ha esperienze, idee e pensieri diversi, il rispetto e il riconoscimento della diversità come un'opportunità sono essenziali per creare un clima positivo.
5. **Coraggio:** il team affronta sfide complesse e prende decisioni difficili con coraggio. Non deve temere di riconoscere l'insorgere di problemi e deve avere la forza di proporre soluzioni innovative, assumendosi la responsabilità delle decisioni prese. È fondamentale avere il coraggio di prendere decisioni anche quando il rischio da assumersi è alto, perché comunque tali decisioni possono portare al miglioramento del progetto.

Il team e le relazioni tra le persone che lo compongono sono quindi molto importanti ed è grazie all'adesione a questi valori che il lavoro di squadra può risultare vincente nel successo dei diversi progetti. È essenziale che venga promossa una cultura di collaborazione e di trasparenza che facciano instaurare un rapporto di fiducia sia all'interno del team, sia con tutti i soggetti impattati. In questo modo, ognuno ha la possibilità di mettere in campo le proprie conoscenze e competenze per essere un valore aggiunto su cui poter contare per dare valore al cliente finale con tempistiche brevi, con un'ottima

capacità di risposta ai cambiamenti e con grande abilità nel gestire anche le situazioni più scomode senza che diventino degli scogli insormontabili. Oltre ai principi e ai valori visti finora, vi sono alcuni elementi, così come presentati all'interno della guida, che forniscono ai team un quadro di riferimento concreto e chiaro per poter gestire il lavoro in modo iterativo ed efficiente. Si tratta dei ruoli chiave delle persone che formano il team, degli eventi che scandiscono le giornate di lavoro e degli artefatti che sono una sorta di cartina tornasole dello stato di avanzamento di un progetto.

2.1.1 I RUOLI CHIAVE

Al centro del framework Scrum troviamo l'unità di base che è il gruppo di lavoro. È formato da un numero ridotto di persone e prende il nome di Scrum Team. Ogni Scrum Team è composto da uno Scrum Master, un Product Owner e da alcuni Developers. All'interno del gruppo non vi sono sottogruppi o gerarchie: tutti collaborano unitamente concentrandosi di volta in volta su un obiettivo, cioè quello che viene chiamato il *Product Goal*. Le persone che compongono i team possiedono diversi background sia di formazione, sia di esperienza lavorativa disponendo quindi di molteplici competenze che unite insieme sono in grado di creare valore in ogni *Sprint*. Con Sprint si intende un ciclo di lavoro della durata di circa uno/quattro settimane, in cui il team lavora per completare un obiettivo. Ciò consente loro di avere una maggiore autonomia e di potersi organizzare e distribuire il lavoro autonomamente. Si parla quindi di team cross-funzionali auto-gestiti. Per garantire un approccio agile e la flessibilità richiesta è necessario disporre di gruppi non troppo grandi, ma che siano comunque opportunamente dimensionati per portare a termine il lavoro previsto in uno Sprint. Generalmente, si tratta di team con una dimensione che non supera le dieci persone, tenendo a mente che il dialogo all'interno del gruppo è molto importante e diventa più complicato quando il numero di persone coinvolte aumenta. Va da sé che un'azienda non disporrà di un singolo Scrum Team, bensì di tanti piccoli team che possono condividere lo stesso obiettivo e che a loro volta dovranno coordinarsi e comunicare efficacemente tra di loro. È responsabilità del gruppo seguire e gestire tutte le attività collegate a un determinato prodotto, incluso il rapporto con gli stakeholder, il testing e la manutenzione. La metodologia Scrum definisce tre ruoli chiave con diverse mansioni e responsabilità così come presentati di seguito.

DEVELOPERS

Gli sviluppatori sono delle figure tecniche responsabili di realizzare concretamente il prodotto, applicando le proprie competenze tecniche. Non si tratta solo di sviluppatori software così come vengono intesi odiernamente, ma riguarda in generale quelle figure che hanno delle competenze in grado di curare ogni aspetto del prodotto, dall'ideazione all'implementazione, dal testing al rilascio. I Developers hanno quindi delle competenze

che possono essere molto varie sulla base del lavoro che stanno svolgendo per raggiungere l'obiettivo. Oltre a quanto detto, hanno la responsabilità di:

- a. creare un piano per lo Sprint, cioè collaborano per creare l'elenco delle attività da completare durante un ciclo di lavoro, chiamato Sprint Backlog,
- b. garantire la qualità assicurandosi che sia in linea con i livelli minimi di un incremento di prodotto, cioè la *Definition of Done*,
- c. adattare giornalmente il piano di lavoro verso il Product Goal,
- d. responsabilizzarsi reciprocamente.

Inoltre, i Developers partecipano a degli eventi, che vedremo meglio dopo, come per esempio il Daily Scrum. In queste occasioni è loro responsabilità individuare all'interno del team chi di volta in volta terrà l'evento. Risulta molto importante anche il modo in cui sono invitati a lavorare insieme, trasformandosi a volte in veri e propri coach per i propri colleghi ai quali trasmettere competenze e conoscenze che possiedono e che condividendo accrescono l'efficienza del gruppo stesso. Quindi un Developer non si limita soltanto ad applicare le proprie competenze tecniche, ma è responsabile in prima persona di collaborare con il team per garantire il raggiungimento dell'obiettivo dello Sprint.

PRODUCT OWNER

Il Product Owner è responsabile di massimizzare il valore del prodotto realizzato dallo Scrum Team, garantendo che il lavoro del team sia costantemente allineato con gli obiettivi aziendali e con le esigenze dei clienti. È colui che detiene una visione chiara del prodotto e ha il compito di trasmetterla a tutti i membri del team. Per poter fornire valore agli stakeholder, interni o esterni che siano, assegna una priorità al lavoro da svolgere per raggiungere il Product Goal. Il suo focus non deve quindi limitarsi al singolo Sprint, ma deve avere bene in mente e massimizzare il valore del prodotto per l'intero ciclo di vita. Inoltre, agisce come collegamento tra gli stakeholder (clienti, utenti finali, management) e il team Scrum. Il Product Owner è il responsabile effettivo della gestione del Product Backlog che consiste nel:

- a. sviluppare e comunicare a tutto il team il Product Goal, in modo tale che risulti chiaro e raggiungibile, dandovi un contesto e chiarendo i risultati concreti attesi al termine dello Sprint,
- b. creare e comunicare con chiarezza gli elementi del Product Backlog,
- c. ordinare, assegnando l'opportuna priorità, gli elementi presenti nel Product Backlog in base alle esigenze dei clienti e all'impatto sul business, assicurandosi che il team si concentri sugli elementi che apportano un maggior valore in linea con le esigenze degli stakeholder,
- d. assicurarsi che il Product Backlog sia trasparente, visibile e comprensibile da tutti gli attori coinvolti.

Oltre alla gestione del Product Backlog e alla prioritizzazione delle attività, il Product Owner mantiene una comunicazione costante con gli stakeholder per raccogliere i feedback e per assicurarsi che il team comprenda a pieno i requisiti. Deve preoccuparsi di gestire il carico di lavoro che il team è in grado di accettare e quindi è responsabile di prendere decisioni in merito agli elementi da inserire o meno all'interno del Product Backlog. Non è detto che tutte le richieste dei clienti debbano per forza essere accolte ed è proprio il Product Owner che ha la responsabilità ultima di prendere una decisione in merito in base alla capacità del team e all'apporto di valore al prodotto finale. Questo aspetto è molto delicato perché non è sempre semplice prendere delle decisioni che garantiscano una riduzione degli sprechi di risorse e di tempo, senza tradire le aspettative dei clienti. Vista la delicatezza del suo ruolo è richiesta una sua partecipazione attiva al fine di garantire il successo di un progetto.

È curioso notare come spesso non risulti chiara la differenza tra la figura del Project Manager e quella del Product Owner, benché si tratti di due ruoli ben distinti. Se ormai risulta chiaro il ruolo del secondo, per quanto riguarda il Project Manager possiamo affermare che si tratta di colui che è responsabile della gestione complessiva del progetto. Si occupa di definire l'intera pianificazione, tenendo conto dei costi e delle tempistiche, oltre che delle risorse necessarie per completare un progetto nella sua interezza. Deve quindi garantire che vengano rispettati i vincoli temporali e di costo, intervenendo in

prima persona qualora vi fossero degli scostamenti. Prevede anche dei piani di contingenza così da mitigare possibili problematiche a cui un progetto può essere soggetto. Per comprendere meglio come questi due ruoli si differenzino possiamo prendere in considerazione alcuni aspetti diversi per entrambi, così come mostrato in Tabella 1.

	PRODUCT OWNER	PROJECT MANAGER
FOCUS	Prodotto	Progetto
RESPONSABILITÀ	Cosa sviluppare	Come gestire
TEAM	Contatto costante	Contatto nei SAL
APPROCCIO	Agile	Tradizionale
DECISIONI	Strategiche	Operative

Tabella 1 - Product Owner e Project Manager: un confronto.

Da un punto di vista del focus, se da un lato abbiamo visto come la massimizzazione del valore del prodotto è un punto cruciale per il Product Owner (PO), il quale quindi ha al centro del suo lavoro il prodotto, gestendo al meglio il Product Backlog, dall'altro lato il Project Manager (PM) invece si focalizza sull'esecuzione del progetto, con un monitoraggio costante dei tempi, dei costi e delle risorse, per garantirne il completamento nei limiti stabiliti. Per quanto riguarda le responsabilità, il PO, gestendo gli elementi presenti nel backlog, decide cosa sviluppare con un criterio di prioritizzazione che tiene conto anche dei desiderata del cliente; il PM decide come è gestito il progetto, coordinando le attività e risolvendo i problemi di pianificazione, senza perdere di vista i costi. Da un punto di vista del rapporto con il team, è possibile notare una grande differenza: il PO è parte

attiva nel team ed è invitato a partecipare costantemente e attivamente ai lavori, essendo molto prezioso il suo apporto in termini di coordinamento di tutte le parti e di conoscenza del prodotto; il PM invece non ha la necessità di essere sempre presente nei team, ma piuttosto si interfaccia con loro su base settimanale per un controllo sull'avanzamento lavori nelle riunioni chiamate SAL (Stato Avanzamento Lavori). Inoltre, l'approccio del PO, essendo una figura nata proprio all'interno del framework Scrum, è agile dal momento in cui è richiesta una buona dose di flessibilità per lavorare in cicli di lavoro brevi per adattare al meglio il prodotto a seguito dei feedback raccolti e in base alla costante revisione del lavoro svolto dal team; il PM invece utilizza un approccio più tradizionale in cui non è richiesta di base una grande flessibilità, ma piuttosto opera una pianificazione a lungo termine in cui non sempre i cambiamenti vengono gestiti con facilità. Infine, da un punto di vista decisionale, possiamo affermare che le decisioni prese da un PO sono strategiche in relazione alle funzionalità di un prodotto, cercando di allineare costantemente gli obiettivi del business al valore atteso dal cliente; il PM dal suo lato prende decisioni più operative per assicurare che tutte le attività del progetto vengano eseguite correttamente, rispettando i vincoli di tempo e di budget.

Il Product Owner assume diversi atteggiamenti che gli consentono di raggiungere l'obiettivo di massimizzazione del valore a seconda delle situazioni in cui si trova. Come normalmente adattiamo il nostro modo di porci con le persone a seconda del contesto in cui ci si trova, il Product Owner si adatta a seconda delle circostanze, del ruolo delle persone con cui ha a che fare, siano esse clienti oppure membri del team. La sua figura richiede quindi una grande versatilità che è quindi da considerarsi una skill fondamentale di chi ricopre questo ruolo. Sulla guida Scrum troviamo sei diversi atteggiamenti assunti da un Product Owner.

1. **Visionario:** un PO è visionario quando comunica efficacemente la visione a lungo termine del prodotto relazionandosi con tutti i soggetti coinvolti, dai colleghi nel team agli stakeholder, comportandosi quindi come una bussola che orienta tutti nella direzione giusta per navigare verso l'orizzonte del prodotto.

- 2. Collaboratore:** un PO è collaboratore quando lavora a stretto contatto con il team e con gli stakeholder. Questo atteggiamento gli permette di favorire una comunicazione che sia chiara ed efficace dando il giusto supporto per il corretto raggiungimento degli obiettivi di uno Sprint. Il PO assumendo questo atteggiamento favorisce un clima più sereno all'interno del team e una maggiore tranquillità e fiducia da parte degli stakeholder.
- 3. Rappresentante del cliente:** il PO rappresenta gli interessi del cliente, sia esso interno o esterno all'azienda. Assumendo questo atteggiamento si assicura che il team stia realmente portando avanti le necessità espresse dal cliente e da lui comunicate e chiarite nei diversi incontri. Il PO si prende cura del cliente, raccoglie i suoi feedback con attenzione, traducendoli in priorità all'interno del Product Backlog.
- 4. Decision Maker:** traducibile in decisore, il PO deve spesso prendere decisioni sapienti in tempi stretti, che siano efficaci per garantire il corretto avanzamento del lavoro. Quindi è bene che assuma questo atteggiamento quando si trova di fronte alle situazioni più critiche, valutando opportunamente i rischi, le opportunità e i vincoli e affinando negli anni le sue capacità decisionali in modo da essere più confidente.
- 5. Sperimentatore:** il PO assume questo atteggiamento incoraggiando il team a testare nuove idee e funzionalità per vedere quali siano le soluzioni migliori. Sperimentare significa avere l'opportunità di toccare con mano qualcosa che non sempre è realmente comprensibile in modo astratto. Si pensi, per esempio, a una nuova funzionalità che, sulla carta, potrebbe sembrare una richiesta un po' strana da parte del cliente, ma che una volta implementata anche solo in un prototipo risulta essere vincente, oppure viceversa, una richiesta del cliente che mentre la si realizza e sperimenta si rivela essere molto più insidiosa o di poco

valore effettivo. Quindi questo atteggiamento è fondamentale per garantire rapidi tempi di adattamento e la continua sperimentazione di nuove funzionalità per selezionare solo le migliori che rappresentino una vera aggiunta di valore per il cliente finale.

- 6. Influencer:** il PO utilizza le sue capacità di persuasione e di comunicazione per ottenere l'approvazione degli stakeholder e del team. È in grado di influenzare le decisioni aziendali e di convincere gli stakeholder delle sue conclusioni in merito alle priorità assegnate, alle nuove funzionalità e alle sue ponderate azioni in merito al prodotto.

SCRUM MASTER

Il ruolo dello Scrum Master è molto importante in uno Scrum Team in quanto è il primo responsabile della corretta applicazione del framework Scrum. Si occupa di garantire che tutti gli attori coinvolti, sia interni sia esterni all'azienda, comprendano al meglio le modalità di organizzazione del lavoro e gli elementi chiave del framework prima di tutto da un punto di vista teorico e poi, successivamente, soprattutto da un punto di vista di applicazione pratica. Lo Scrum Master deve possedere competenze trasversali che gli consentano di essere un punto di riferimento per il team e per l'azienda, ricoprendo un ruolo di coach e mentore dei membri del gruppo di lavoro e degli altri soggetti coinvolti nei progetti. Offre assistenza su base individuale o in gruppo per facilitare il progredire dei lavori garantendo che non vi siano ostacoli che impediscano la corretta applicazione delle pratiche agili di Scrum. In tal senso fornisce consigli, esercizi e da informazioni utili a garantire la corretta autonomia decisionale dei team. Insomma, l'importanza del ruolo sta proprio nel fatto che gli Scrum Master si occupano di creare un ambiente favorevole alla crescita professionale dell'intero team, garantendo e promuovendo una cultura di collaborazione e di miglioramento continuo. Aspetto critico del lavoro dello Scrum Master risiede nella rimozione degli impedimenti. Infatti, oltre a favorire le dinamiche del team, è sua responsabilità muoversi in vantaggio per identificare e risolvere possibili ostacoli che potrebbero minare il normale progresso del lavoro del team. Si potrebbe

trattare di problemi tecnici, organizzativi o legati alla comunicazione, ma è importante che agisca costantemente per assicurarsi che tutte le problematiche abbiano una risoluzione tempestiva e quasi trasparente al team che può continuare il proprio lavoro in modo efficiente e senza interruzioni. Quindi lo Scrum Master mette in campo un set di competenze per aiutare il team, il Product Owner e anche l'azienda stessa così come riportato di seguito.

Responsabilità dello Scrum Master nei confronti del team:

- a. supportare l'intero team attraverso il coaching per favorire la gestione autonoma e la cross-funzionalità,
- b. favorire la concentrazione sulla creazione di incrementi di valore che siano in linea con la Definition of Done,
- c. rimuovere possibili impedimenti che ostacolerebbero il progresso del team,
- d. garantire che si svolgano tutti gli eventi Scrum e che risultino positivi e produttivi rispettando i tempi previsti.

Lo Scrum Master aiuta il Product Owner a:

- a. trovare tecniche per una definizione efficace del Product Goal e per la gestione degli elementi nel Product Backlog,
- b. fornire modalità utili a far comprendere l'importanza della chiarezza ed efficacia degli elementi presenti nel Product Backlog,
- c. stabilire una pianificazione empirica del prodotto in ambienti complessi,
- d. facilitare la comunicazione e la collaborazione con gli stakeholder quando richiesto o se necessario.

Infine, lo Scrum Master supporta l'azienda:

- a. agendo come coach e formando i dipendenti nell'adozione di Scrum intrapresa dall'azienda,
- b. aiutando i dipendenti e gli stakeholder a comprendere e mettere in pratica un approccio empirico per i lavori complessi,

- c. rimuovendo le barriere tra gli stakeholder e lo Scrum Team.

Anche lo Scrum Master, come il Product Owner, deve essere in grado di assumere atteggiamenti diversi a seconda delle situazioni e delle sfide che il team sta affrontando. Nello specifico, Barry Overeem, noto Agile Coach e Scrum Trainer, ha identificato otto atteggiamenti che garantiscono la flessibilità della figura dello Scrum Master e aiutano il team al raggiungimento del massimo potenziale.

1. **Servant Leader:** atteggiamento fondamentale, per il quale lo Scrum Master mette il benessere del team al primo posto aiutandolo nell'organizzazione autonoma, nella rimozione degli ostacoli e fornendo gli strumenti per garantire un miglioramento continuo.
2. **Facilitatore:** lo Scrum Master si assicura che ogni evento Scrum sia produttivo, coinvolgente e realmente utile al benessere del team. Non interferisce nelle decisioni che vengono prese, ma fa sì che il processo decisionale sia facilitato.
3. **Coach:** lo Scrum Master guida il team verso il miglioramento sia delle competenze, sia della performance di ogni membro, fornendo critiche costruttive per favorire la riflessione sul lavoro svolto e promuovendo il continuo apprendimento.
4. **Manager:** di per sé lo Scrum Master non gestisce direttamente il team, ma comunque si occupa di garantire che l'ambiente nel quale opera sia adatto e che l'interazione team-azienda siano fluide, lavorando anche con altri dipartimenti dell'azienda riducendo ciò che potrebbe causare un rallentamento al team.
5. **Mentore:** atteggiamento molto utile soprattutto nei confronti di nuovi membri che si affacciano da poco nel mondo Scrum. È bene che lo Scrum Master si occupi

di aiutare i membri del team a crescere professionalmente e a conoscere al meglio le pratiche e gli strumenti utilizzati dallo Scrum Team.

6. **Insegnante:** legato in qualche modo all'atteggiamento precedente, ma esteso all'intera azienda, lo Scrum Master informa e spiega i principi e le pratiche Scrum anche al di fuori del team, quindi coinvolgendo anche gli stakeholder. In questo modo, si assicura che tutti comprendano il funzionamento del framework e le motivazioni per cui è importante seguirlo attentamente.
7. **Rimuovi Impedimenti:** atteggiamento dal nome buffo, ma assolutamente cruciale che permette allo Scrum Master di far sì che tutti gli impedimenti rilevati o che si presentano durante i lavori del team siano opportunamente rimossi, al di là della natura stessa degli ostacoli.
8. **Agente del Cambiamento:** per garantire che l'adozione di Scrum a livello aziendale avvenga il più fluidamente possibile, superando le fisiologiche resistenze al cambiamento, questo atteggiamento fa sì che lo Scrum Master promuova e guidi l'intera azienda ad abbracciare a pieno i principi e le pratiche agili in un cambiamento organizzativo più ampio.

2.1.2 GLI EVENTI

Un elemento molto importante del framework Scrum sono gli eventi. È proprio grazie agli eventi che l'intero lavoro del team viene opportunamente scandito permettendo di valutare e adattare l'operato e fornire la trasparenza necessaria per dare a tutti la corretta percezione di quanto sta accadendo durante il ciclo di lavoro. Gli eventi fanno sì che vi siano riunioni solamente necessarie, sono contenuti all'interno di quello che viene chiamato Sprint e sono: lo *Sprint Planning*, il *Daily Scrum*, la *Sprint Review* e la *Sprint Retrospective*. Come vedremo ogni evento ha caratteristiche proprie e obiettivi ben precisi aiutando il team a rimanere focalizzato, flessibile e produttivo. Sono proprio gli eventi che permettono al team di interrogarsi costantemente sulla bontà del lavoro svolto, di prendere decisioni, di identificare possibili miglioramenti e di conseguenza di mettere in pratica l'evoluzione continua delle prestazioni e l'assorbimento costante di possibili cambiamenti senza andare incontro a conseguenze stravolgenti. Elemento base di Scrum sono gli Sprint, cicli di lavoro iterativi e incrementali che il team utilizza per sviluppare il prodotto. Ogni Sprint ha una durata fissa che generalmente non supera il mese – tipicamente due settimane – durante la quale il team crea un incremento di valore aggiungendo al prodotto nuovi elementi funzionanti attraverso l'esecuzione di diverse attività. Terminato uno Sprint ha inizio il successivo, senza sosta. Vi sono alcune regole che è importante rispettare durante uno Sprint:

- a. non sono ammessi cambiamenti che potrebbero compromettere l'obiettivo dello Sprint,
- b. la qualità del prodotto non deve diminuire,
- c. il Product Backlog può essere sempre perfezionato qualora ve ne sia la necessità,
- d. il piano di lavoro può variare e adattarsi sulla base delle nuove informazioni apprese previo consulto e approvazione del Product Owner, ma senza variare l'obiettivo.

Ogni Sprint prevede che il team controlli attivamente il lavoro svolto e si adatti nella direzione di quello che viene chiamato il *Product Goal*. Si tratta di uno stato futuro del prodotto che funge da obiettivo a lungo termine per il team. Viene inserito all'interno

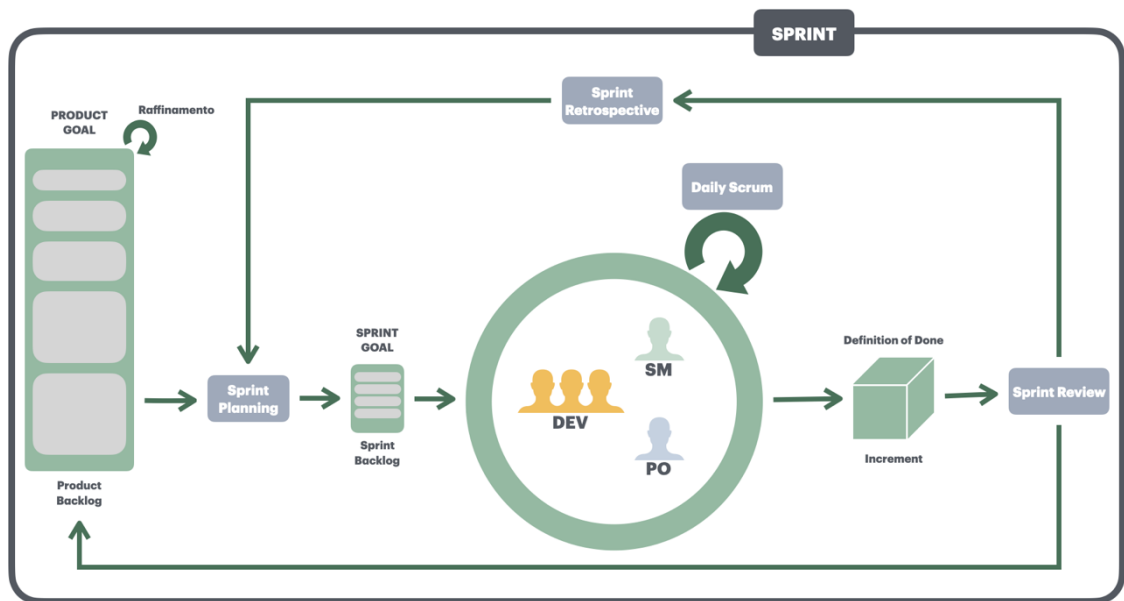


Figura 6 - Schema di uno Sprint Scrum.

del Product Backlog dando una visione più chiara su ciò che si vuole raggiungere nel futuro più lontano, attraverso l'evoluzione del Product Backlog, formato dagli elementi che contribuiranno alla realizzazione dell'obiettivo. È quindi il Product Goal che guida le decisioni strategiche di ogni team che non può passare al successivo se non ha prima terminato o abbondato un obiettivo. Parallelamente, ogni Sprint ha un obiettivo più circoscritto e a breve termine, chiamato *Sprint Goal*. Si tratta di un singolo obiettivo specifico per ogni Sprint sul quale il team dovrà concentrare i propri sforzi e lavorare collaborando unitamente senza disperdere energie su molteplici iniziative separate. È molto importante che i team abbiano un approccio empirico soprattutto all'inizio dei lavori. Come già detto più volte non vi è una regola precisa, ma vi sono molte persone che insieme possono ascoltare, osservare e comunicare idee, suggerimenti e perplessità così da adattare man mano gli strumenti del framework alle proprie esigenze. Per esempio, se si nota che la durata degli Sprint di una settimana risulta troppo corta per raggiungere lo Sprint Goal, non vi è alcun impedimento nell'estenderla. L'importante è che le decisioni vengano prese sempre coinvolgendo l'intero team. Questo approccio empirico che permette di calibrare il tiro in corso d'opera permette ai team di gestire progetti sempre più complessi con la stessa agilità di progetti più semplici. In Figura 6 è possibile vedere una rappresentazione schematica di cosa si trova all'interno di uno Sprint Scrum, compresi gli eventi principali descritti qui di seguito.

SPRINT PLANNING

All'inizio di ogni Sprint si tiene come primo evento lo *Sprint Planning*. Questo permette al team di andare a definire in modo chiaro quali saranno le attività da svolgere durante l'intera durata dello Sprint, definendo in modo chiaro gli obiettivi in base alle aspettative. Tutti i membri del team prendono parte al planning contribuendo in base al proprio ruolo. È responsabilità del Product Owner di presentare gli elementi più importanti presenti nel Product Backlog e di come questi contribuiscano al raggiungimento del Product Goal, così che si possa decidere di comune accordo quali elementi entreranno a far parte della lista delle attività da svolgere durante lo Sprint. In questo modo è anche possibile identificare e definire in modo chiaro l'obiettivo dello Sprint. È possibile che al planning partecipino anche soggetti esterni al team in quanto possono fornire ulteriori informazioni utili a comprendere a pieno le richieste. Si pensi, per esempio, all'importanza delle informazioni che può dare un utente interno all'azienda che conosce a pieno lo stato attuale di un applicativo per il quale si richiedono nuove funzionalità, oppure, quanto potrebbe essere preziosa la partecipazione di alcuni stakeholder che possono ribadire i propri interessi principali, permettendo di raffinare ulteriormente la priorità di ogni attività. Per facilitare lo svolgimento dell'incontro e la pianificazione dello Sprint è opportuno porsi tre domande, così come indicato nella guida Scrum.

- 1. Perché questo Sprint ha valore?** Sapere come il ciclo di lavoro che avrà inizio apporterà un'aggiunta di valore al prodotto è un aspetto molto importante ed è quindi responsabilità del Product Owner comunicarlo chiaramente. L'intero team collabora di conseguenza per definire lo Sprint Goal che dovrà essere finalizzato entro il termine della riunione e rappresenterà appunto il perché questo Sprint ha valore per gli stakeholder.
- 2. Cosa può essere completato in questo Sprint?** Gli sviluppatori selezionano dal Product Backlog gli elementi da inserire nel backlog dello Sprint che sta per iniziare confrontandosi con il Product Owner. Gli elementi possono essere specificati più nel dettaglio per renderli più chiari. Per gli sviluppatori si tratta

effettivamente di prevedere quanto saranno in grado di realizzare durante questo ciclo di lavoro. Va da sé che dovranno quindi basarsi sull'esperienza passata e sulla loro capacità attuale per evitare di assegnarsi un carico di lavoro eccessivo.

- 3. Come verrà completato il lavoro scelto?** A questo punto, gli sviluppatori sono invitati a organizzare autonomamente il modo in cui ogni elemento selezionato dal Product Backlog verrà implementato giorno dopo giorno. Non c'è un metodo unico, ma gli sviluppatori sceglieranno il modo e gli strumenti che più ritengono adatti in base alle proprie competenze ed esperienze pregresse.

Lo Sprint Goal, gli elementi selezionati dal Product Backlog per lo Sprint attuale e il piano di come verrà svolto il lavoro compongono quello che viene chiamato lo *Sprint Backlog* che è quindi l'output dell'evento di Sprint Planning. Generalmente è bene cercare di non dedicare più di otto ore per la pianificazione di uno Sprint della durata di un mese.

DAILY SCRUM

Una volta preparato lo Sprint Backlog può avere inizio lo sviluppo vero e proprio. In questa fase l'evento introdotto da Scrum è il *Daily Scrum*. Questo permette di fare il punto della situazione sull'avanzamento dei lavori. È un'occasione molto importante per tutti i componenti del team perché possono evidenziare punti critici emersi durante gli sviluppi oppure richiedere l'aiuto nelle attività che stanno svolgendo. In questo modo è possibile aggiustare costantemente la pianificazione prevedendo eventuali modifiche che consentano di non restare bloccati a causa di imprevisti che vengono in questo modo immediatamente palesati e gestiti. Il Daily ha una frequenza giornaliera e una durata di quindici minuti. Generalmente il Product Owner e lo Scrum Master vi partecipano solo se impattati dalle attività in corso di lavorazione altrimenti al Daily partecipano solo i Developers e sono invitati ad autogestirsi nelle modalità che più prediligono. Inizialmente, lo Scrum Master può spiegare come far sì che in soli quindici minuti si svolta il meeting e verificare che non vi siano distrazioni che compromettano il corretto svolgimento, ma non interviene perché sono gli sviluppatori che sono incaricati di condurlo. La durata è

volutamente breve per permettere di discutere rapidamente solo delle questioni più rilevanti e che necessitano dell'attenzione di tutti. Ciò non toglie che durante la giornata gli sviluppatori sono comunque invitati al dialogo e allo scambio continuo di informazioni utili all'avanzamento del progetto o all'analisi più precisa di determinate criticità segnalate durante il Daily, cosa che spesso accade proprio immediatamente dopo.

SPRINT REVIEW

Al termine dello Sprint il team e gli stakeholder sono invitati a incontrarsi per revisionare quanto è stato prodotto e per discutere dell'avanzamento verso il Product Goal. Bisogna notare come questo evento non preveda una partecipazione passiva degli attori al tavolo, limitati a osservare una presentazione da parte degli sviluppatori. Si tratta di un momento molto utile per capire se c'è qualcosa da cambiare nel Product Backlog ora che è stato apportato del nuovo valore al prodotto finale e che quindi ha già assunto una forma diversa da quella iniziale. Si pensi, per esempio, alla richiesta di aggiunta di un pulsante che permetta di segnare come lette le notifiche presenti in un elenco. Finché il pulsante non è implementato non vi è altro modo per togliere l'indicatore di notifica non letta e nel Product Backlog sono presenti questi tre elementi: man mano che le notifiche vengono cliccate devono essere segnate come lette; creare un pulsante che una volta cliccato segni come lette tutte le notifiche; aggiungere un pulsante con un filtro che permetta di filtrare le notifiche. Durante la preparazione allo Sprint solo il primo elemento viene selezionato per lo Sprint Backlog. Durante la Sprint Review gli stakeholder si rendono conto che il numero di notifiche non è generalmente molto elevato e non è così scomodo cliccare una notifica dopo l'altra per segnlarla come letta ora che questa funzionalità è stata implementata e ha aggiunto valore. Decidono quindi che il pulsante che permette di segnare come lette tutte le notifiche in un solo click non risulti più di immediata necessità in favore del pulsante per la selezione dei filtri che potrà essere implementato durante il prossimo Sprint, insieme ovviamente a tutti gli altri elementi che verranno scelti dai Developers nel Product Backlog. C'è stato quindi un cambio di priorità avvenuto proprio grazie alla partecipazione attiva di tutti durante la Sprint Review. Generalmente, è bene dedicare non più di quattro ore per questo evento per Sprint durate

un mese, altrimenti per Sprint più corte anche il tempo dedicato alla review va accorciato.

SPRINT RETROSPECTIVE

Ultimo evento di uno Sprint è la *Sprint Retrospective*. A differenza della review, in questo evento non sono presenti gli stakeholder, ma solo il team il quale porta avanti una discussione per comprendere come è andato l'ultimo Sprint. È utile capire se c'è qualcosa che non ha funzionato al meglio e se per qualche motivo non è stata rispettata la qualità prevista per l'incremento di valore al prodotto. Ovviamente non si tratta di evidenziare i punti di debolezza per poter attribuire la colpa di eventuali mancanze a un determinato membro, ma si tratta di andare a comprendere le cause alla radice di tutto ciò che ha messo in difficoltà il team, a prescindere dal membro direttamente più o meno coinvolto. Nello spirito di condivisione rientra proprio anche questo aspetto: le sfide sono di gruppo e del gruppo, allo stesso modo, i fallimenti anche sono di gruppo e spetta proprio al gruppo riuscire a far meglio nella sua unità la volta successiva. Potrebbero essere anche identificate delle modifiche utili per migliorare quanto realizzato che potrebbero entrare a far parte del Product Backlog. Durante la Sprint Retrospective il team può, per esempio, utilizzare una lavagna suddivisa in tre colonne per indicare in ognuna cosa è andato bene nello Sprint, cosa potrebbe essere migliorato e cosa si decide di migliorare effettivamente durante il prossimo Sprint. Questo evento ha generalmente una durata di tre ore per Sprint di un mese. La Sprint Retrospective è quindi un evento molto importante perché permette di adattare e migliorare costantemente il lavoro del team, Sprint dopo Sprint.

2.1.3 GLI ARTEFATTI

Finora abbiamo già introdotto alcuni degli artefatti del framework Scrum in quanto sono elementi chiave che permettono di dare un'evidenza immediata e chiara sullo stato di avanzamento di un progetto e quindi sul lavoro svolto dal team. Gli artefatti sono fondamentali per il team in quanto permettono di raccogliere e di organizzare le informazioni significative che consentono di prendere decisioni consapevoli e di comprendere il reale avanzamento verso gli obiettivi prefissati. Tra gli artefatti Scrum abbiamo il Product Backlog e il Product Goal, lo Sprint Backlog e lo Sprint Goal, l'*Increment* e la Definition of Done. Questi permettono di mantenere allineato il lavoro del team con gli obiettivi generali, garantendo che il focus sia costantemente sul valore da creare per il cliente. Favoriscono, inoltre, la trasparenza in quanto permettono anche agli stakeholder oltre che al team di monitorare comodamente e chiaramente il progresso. In base a quanto indicato sulla guida Scrum, è possibile leggere di seguito una descrizione più dettagliata di ogni artefatto.

PRODUCT BACKLOG E PRODUCT GOAL

Il Product Backlog è una lista di ciò che è necessario per apportare un miglioramento a un determinato prodotto. Gli elementi nella lista sono opportunamente ordinati in base alla priorità assegnata a ognuno di essi. È da questa lista da cui il team Scrum attinge all'inizio di ogni Sprint per decidere quali attività svolgere durante l'evento Sprint Planning. Generalmente, gli elementi presenti nel Product Backlog sono pronti per uno Sprint quando raggiungono un livello di dettaglio che li rende molto chiari. Si procede quindi con una scomposizione di elementi più complessi in elementi più semplici fintanto che non si ottengono delle piccole attività più specifiche. Inoltre, si aggiungono dettagli come una descrizione, una priorità e una dimensione. Quest'ultima generalmente è assegnata dagli sviluppatori che possono essere influenzati dal Product Owner per trovare dei compromessi. Il processo di continua scomposizione e aggiunta di dettagli degli elementi del backlog prende il nome di *Product Backlog Refinement* e può avvenire durante delle riunioni appositamente programmate oppure in qualsiasi momento in cui lo si ritenga

necessario durante uno Sprint. Potenzialmente potrebbe anche non essere fatto, ma non bisogna dimenticarsi dell'importanza di disporre di attività chiare e trasparenti, piuttosto che di rischiare di mal interpretare i requisiti e le richieste del cliente. È importante notare che vi è una relazione uno a uno tra un determinato Product Backlog e uno specifico prodotto. Di conseguenza, qualora più team lavorassero contemporaneamente sullo stesso prodotto, non sarà necessario disporre di due backlog, ma i vari gruppi di lavoro attingeranno dallo stesso.

Il Product Goal permette di contestualizzare un Product Backlog in quanto descrive lo stato futuro di un prodotto identificandolo come punto d'arrivo a cui tendere per il team Scrum. Si tratta quindi dell'obiettivo a lungo termine raggiungibile attraverso la realizzazione successiva di tutti gli elementi presenti nel Product Backlog. È il Product Owner che grazie al contatto costante con gli stakeholder si impegna a creare e a comunicare chiaramente il Product Goal, cioè la visione a lungo termine del prodotto. Non può esserci più di un Product Goal per un singolo Product Backlog e deve essere chiaro e conciso. Per comprendere meglio come potrebbe essere formulato, facciamo un esempio di Product Goal per un prodotto di una banca:

“Sviluppo di una piattaforma di home banking che permetta ai clienti di monitorare i propri conti correnti, effettuare pagamenti e controllare i propri investimenti mediante un'interfaccia utente intuitiva e accessibile sia da dispositivi desktop, sia da dispositivi mobile.”

Qui viene descritto opportunamente lo stato futuro del prodotto e quindi risulta chiara la direzione verso cui il team dovrà orientarsi per sviluppare le funzionalità presenti nel Product Backlog.

SPRINT BACKLOG E SPRINT GOAL

Come visto in precedenza parlando dello Sprint Planning, uno Sprint Backlog è formato dallo Sprint Goal, cioè il perché si sta facendo quel determinato Sprint, dagli elementi selezionati dal Product Backlog, cioè il cosa andrà fatto, e infine la programmazione di

come verrà effettivamente implementata l'aggiunta di valore. Ha significato principalmente per i Developers in quanto rappresenta in tempo reale il lavoro che dovranno svolgere all'interno del ciclo di lavoro per raggiungere l'obiettivo prefissato. Lo Sprint Backlog deve essere costantemente aggiornato man mano che avanzano i lavori e deve essere sufficientemente dettagliato per favorire il confronto durante il Daily Scrum. Svolgere le diverse attività indicate nel backlog permette al team di avvicinarsi allo Sprint Goal. Si tratta infatti del singolo obiettivo a breve termine che il team si impegna a raggiungere al termine di uno Sprint. Aiuta a mantenere il focus del team favorendo un approccio collaborativo per raggiungere insieme l'obiettivo comune. Benché l'obiettivo durante uno Sprint non possa cambiare, può accadere che mentre i Developers svolgono il loro lavoro emerga qualcosa di inaspettato, come, per esempio, delle complicanze che aumentano il carico di lavoro rispetto a quello previsto. In tal caso sarà opportuno confrontarsi con il Product Owner e con gli altri membri del team per cercare una soluzione che permetta di raggiungere comunque l'obiettivo dello Sprint. Tuttavia, in alcuni casi potrebbe anche ritenersi opportuno l'annullamento del ciclo di lavoro in corso, sempre in una logica di adattamento rapido e non bloccante. Così come per il Product Goal, anche lo Sprint Goal deve essere chiaro perché è fondamentale che il team si renda conto immediatamente di quale sia l'obiettivo da raggiungere durante lo Sprint. Tale chiarezza permette anche di comprendere immediatamente se l'obiettivo sia effettivamente raggiungibile e soprattutto se è opportunamente commisurato alla reale capacità del team.

INCREMENT E DEFINITION OF DONE

Vi sono ancora due artefatti del framework Scrum di cui è bene conoscerne il significato. Il primo, l'Increment, viene definito come un concreto passo avanti verso il Product Goal. Va notato che ogni incremento va a sommarsi a quelli precedenti ed è molto importante che non vi siano quindi dei conflitti o delle regressioni. I team devono quindi tener conto delle possibili dipendenze tra un incremento e i precedenti, assicurandosi che tutti gli incrementi possano funzionare correttamente insieme. Si pensi, per esempio, in ambito IT, alle possibili conseguenze del rilascio in produzione di un Increment del quale non si sono verificati i possibili impatti su quanto finora rilasciato: un disastro. Durante uno

stesso Sprint possono essere realizzati più Increment che saranno resi disponibili agli stakeholder ancora prima della Sprint Review. Tuttavia, il frutto del lavoro svolto non può essere considerato un Increment se non rispetta la Definition of Done, cioè l'ultimo artefatto non ancora presentato fin qui, ma che è stato nominato già precedentemente. In pratica, la Definition of Done indica quelle che sono le caratteristiche minime che un Increment deve rispettare per essere considerato realmente un qualcosa di pronto per essere rilasciato. In altri termini, si tratta di verificare se quanto è stato prodotto durante uno Sprint è in linea con gli standard attesi, i quali possono essere quelli minimi imposti direttamente dall'azienda, ma anche, ove non presenti o in aggiunta a essi, quelli che il team stesso ha deciso di definire. Grazie alla Definition of Done (DoD) tutti gli attori coinvolti nel progetto sono in grado in modo trasparente di condividere e di comprendere il livello minimo di qualità che ogni Increment deve avere per essere considerato completo. Se il lavoro svolto non è in linea con la DoD allora sarà necessario analizzare le cause e conseguentemente agire diversamente in modo tale che ogni risultato ottenuto durante uno Sprint garantisca un livello minimo di qualità. Per rendere meglio l'idea, di seguito è possibile trovare esempio di alcuni dei possibili elementi contenuti all'interno di una DoD per un applicativo web:

- System Integration Test (SIT) terminati con successo
- User Acceptance Test (UAT) terminati con approvazione degli utenti
- Revisione del codice completata
- Test di sicurezza completati e superati
- Bug critici risolti
- Documentazione aggiornata
- Documento di deploy pronto
- Backup effettuati

Qualora quanto realizzato non soddisfacesse uno degli elementi presenti nella lista, allora non lo si potrà considerare un Increment in quanto non è pronto per essere rilasciato garantendo un determinato livello di qualità.

2.2 SAFE®

Quanto presentato finora per agire in modo Agile mediante l'applicazione di Scrum, funziona molto bene con dei team di piccole dimensioni così come previsto dal framework stesso. Tuttavia, come è stato affermato più volte nel primo capitolo di questa tesi, il cambiamento verso l'adozione di pratiche più flessibili è bene che riguardi l'intera azienda per trarne così l'atteso e più ampio beneficio. In tal senso, negli anni sono nati diversi framework con l'intento di applicare i principi e le pratiche Agile a intere organizzazioni molto più complesse in cui il numero dei team e dei prodotti è notevolmente elevato. Si pensi banalmente alla maggiore complessità introdotta dal solo fatto che più team lavorano sullo stesso prodotto contemporaneamente. Garantire il corretto allineamento, la coerenza e l'efficienza è una vera e propria sfida da approcciare con degli strumenti che aiutino un'intera organizzazione ad agire in modo snello e agile. Più in generale si parla di framework Scaled Agile o Scaled Scrum che, in un modo o nell'altro, cercano di applicare i principi agili a livello organizzativo, facilitando la comunicazione e la collaborazione tra team multipli e garantendo che gli obiettivi strategici aziendali vengano rispettati. Ne sono un esempio *Scrum@Scale* e *Nexus*, sviluppati rispettivamente da Jeff Sutherland e Ken Schwaber (fondatori di Scrum), i quali permettono di scalare in modo modulare quanto detto su Scrum senza aggiungere eccessiva complessità.

Uno degli *scaled framework* attualmente maggiormente diffuso è SAFE® (Scaled-Agile Framework®)²¹. Basato sui concetti lean e Agile, fornisce una struttura per gestire progetti molto complessi nei quali sono coinvolti molteplici team con un occhio di riguardo agli obiettivi strategici dell'azienda. Pubblicato nella sua prima versione nel 2011 da Dean Leffingwell, esperto di sviluppo software con una notevole esperienza nelle pratiche snelle e agili, è evoluto negli anni con l'aggiunta di nuove componenti fino alla versione 6.0, la più recente, pubblicata a marzo 2023²². Anche in questo caso è bene in prima battuta concentrarsi su quelli che sono i principi alla base del framework per poi comprendere successivamente al meglio le soluzioni pratiche che esso propone. In questo

²¹ SAFE and Scaled Agile Framework are registered trademarks of Scaled Agile, Inc.

²² Scaled Agile Inc., *Welcome to Scaled Agile Framework® 6.0!*, disponibile online all'indirizzo <https://scaledagileframework.com/about/> (ultimo accesso in data 18/10/2024).

caso, SAFe si propone di sintetizzare i principi Agile e lean insieme alle pratiche di sviluppo del prodotto partendo dall'esperienza di leader del settore con l'obiettivo di creare un sistema integrato che, sulla base di pratiche affinate negli anni, migliora la qualità, la produttività e il lavoro di gruppo. Si parla sempre di principi che fungano da guida, garantendo all'azienda l'elasticità opportuna per loro applicazione in situazioni eterogenee, proprio perché non bisogna dimenticare che ogni realtà aziendale presenta delle caratteristiche uniche e specifiche. Di seguito sono presentati con alcuni esempi i dieci principi SAFe Lean-Agile²³.

- 1. Assumere una prospettiva economica:** è molto importante che tutte le persone coinvolte nella realizzazione di un prodotto siano consapevoli dei costi. È bene che i leader, i manager e i lavoratori comprendano l'impatto economico delle proprie scelte in modo tale che queste siano opportunamente guidate grazie a una sapiente consapevolezza dell'economia del progetto. I team riescono in questo modo a bilanciare al meglio tempo, costo e qualità raggiungendo l'obiettivo di massimizzazione del valore per il cliente e per l'azienda così come auspicato dai principi Lean-Agile. Si pensi, per esempio, a un team che, trovandosi a dover sviluppare una determinata funzionalità, possa optare per uno sviluppo interno o esterno. Consapevoli degli impatti economici della loro scelta, si rendono conto in modo più immediato che in questo caso uno sviluppo interno avrebbe tempistiche e costi più elevati rispetto all'assegnazione del lavoro a un fornitore esterno, che garantisce di ottimizzare i tempi di rilascio e la qualità del prodotto. In SAFe troviamo strumenti come il *Lean Portfolio Management*, il quale permette di allineare le risorse finanziarie alle priorità aziendali.
- 2. Applicare il pensiero sistemico:** l'azienda va considerata come un sistema unico all'interno del quale ogni elemento è interconnesso e tutte le decisioni che vengono prese dalle diverse parti hanno delle ricadute su altre. Le diverse

²³ Scaled Agile Inc., *SAFe Lean-Agile Principles*, disponibile online all'indirizzo <https://scaledagileframework.com/safe-lean-agile-principles/> (ultimo accesso in data 18/10/2024).

componenti del sistema, quindi, collaborano per produrre valore sia per il prodotto finale, sia per l'organizzazione stessa. Non vanno quindi mai perse di vista quelle che sono le interconnessioni di ogni componente e la progettazione va sempre fatta con un occhio di riguardo sempre complessivo. Si pensi, per esempio, al rilascio di una nuova funzionalità che si rivela di difficile comprensione per l'utente finale. Dal momento del rilascio, il reparto di assistenza potrebbe ritrovarsi sovraccaricato dalle richieste di supporto a causa delle difficoltà riscontrate dai clienti che necessitano di continui chiarimenti. Va da sé che la pressione che si crea sul team di assistenza riduce la qualità del lavoro svolto e aumenta i tempi di risposta complessivi causando un vero e proprio collo di bottiglia pericoloso. In poche parole, in questo esempio è bastata una singola componente con scarsa usabilità per generare delle ripercussioni sull'intera esperienza utente. SAFe propone alcuni strumenti, come il *Value Stream Mapping*, che permettono di visualizzare e quindi ottimizzare i flussi, riducendo i possibili attriti e aumentando la collaborazione costante tra le diverse parti di un sistema unico.

- 3. Assumere variabilità e preservare le opzioni:** bisogna considerare la variabilità come un qualcosa di normalmente insito nei progetti complessi. Bisogna quindi accogliere la variabilità come un'opportunità, ritenendo vantaggioso un approccio che sia iterativo e flessibile. È altresì importante esplorare diverse alternative e non vincolarsi a singole soluzioni frettolose. Vi sono diverse strade e possibilità per affrontare le questioni che man mano si presentano durante la vita di un progetto ed è quindi bene non concentrarsi solo su una soluzione specifica che rischia nel tempo di rivelarsi poco adatta. Un approccio di questo tipo permette di disporre di varie strade verso cui si è pronti a svoltare nel momento in cui i feedback e le informazioni raccolte palesano la necessità di un adattamento in corso d'opera, minimizzando così il rischio di sprecare delle risorse. Per esempio, un team che si occupa dello sviluppo di un'interfaccia grafica per un software potrebbe creare diverse versioni e avvalersi di alcuni *mockup* da proporre agli utenti con cui effettuare dei test che permettano l'identificazione di quella migliore,

riducendo il rischio di aver ideato una soluzione non in linea con le aspettative dei clienti. SAFe propone l'uso di prototipi, di versioni beta e di cicli di test frequenti come strumenti per facilitare l'esplorazione di più soluzioni e la successiva scelta di quella che possa aggiungere più valore al prodotto.

4. Implementare incrementalmente con cicli di apprendimento rapidi: risulta molto vantaggioso organizzare il lavoro in modo tale che possa fornire costantemente piccole aggiunte di valore che si sommano alle precedenti in un approccio quindi iterativo che migliora nel tempo grazie a cicli di apprendimento rapidi. In altri termini, incrementare il valore di un prodotto in tempi brevi vuol anche dire avere la possibilità di ridurre il tempo che intercorre dalla richiesta di una funzionalità alla sua realizzazione con quindi l'opportunità di valutarne la bontà più frequentemente. Ridurre il tempo che si impiega per comprendere se quanto implementato è effettivamente in linea con le aspettative consente di fornire incrementi sempre più di valore e di qualità maggiore man mano che si avanza nella realizzazione del prodotto. Al contrario, si pensi, per esempio, agli svantaggi di un approccio più tradizionale in cui i clienti attendono che l'intero prodotto sia realizzato prima di poter dare un feedback prezioso su quanto questo sia in linea con la loro richiesta iniziale, pervenuta a volte anche più di un anno prima. Non solo diventa difficile per il cliente ricordare il dettaglio di quanto effettivamente era stato richiesto, ma nel tempo le necessità sono molto probabilmente mutate, ma anche piccole variazioni diventano molto difficili da accogliere a lavori ultimati. Per questo motivo SAFe incoraggia i team a utilizzare un approccio basato su eventi che favoriscano l'ispezione e l'adattamento come il *Program Increment* e la *Sprint Review*, assicurando che ogni passo avanti sia effettivamente portatore di valore aggiunto.

5. Basarsi su milestone oggettive per valutare il progresso: per far sì che quanto esposto nei principi 3 e 4 possa essere realmente messo in pratica, è necessario identificare dei punti di controllo oggettivi in cui il team e le persone coinvolte in

un progetto possano verificare se le funzionalità implementate rispondano agli standard prefissati e alle esigenze. In un sistema Waterfall che ragiona a fasi stagne i punti di controllo rischiano di essere troppo vincolati alle singole fasi piuttosto che alla reale necessità di monitorare l'andamento di un progetto. L'idea che si possa definire a priori una strada da percorrere come l'unica e la migliore e che quindi un prodotto possa essere creato opportunamente al primo tentativo non prende in considerazione la variabilità e crea lunghi processi di sviluppo e quantità enormi di test da effettuare durante i quali è difficile misurare al meglio la reale bontà di quanto è stato realizzato. Identificare i momenti più opportuni per monitorare il progresso, scomponendo lunghi periodi stagni di attività, permette di misurare più puntualmente l'andamento e di orientare conseguentemente il lavoro futuro. In SAFe, per esempio, l'introduzione delle *System Demo* permette di mostrare l'integrazione delle funzionalità sviluppate avendo la possibilità di misurare i progressi effettivi verso gli obiettivi del Program Increment.

6. Far fluire il valore senza interruzioni: questo principio è in linea con quanto è stato presentato precedentemente per quanto riguarda la realizzazione di un flusso continuo nel Toyota Production System. Anche SAFe si pone come obiettivo quello di favorire un flusso di lavoro che non preveda interruzioni e che quindi sia in grado, in un costante miglioramento, di apportare un'aggiunta di valore costante al prodotto finale. In tal senso vengono proposti otto acceleratori del flusso:

- 1) Visualizzare e limitare i lavori in corso: è fondamentale che ogni team non sia caricato di una quantità di lavoro che non è in grado di svolgere in quanto questo genera confusione e una notevole riduzione della qualità del lavoro svolto. È opportuno quindi utilizzare dei meccanismi che permettano anzitutto di visualizzare il carico di lavoro per poi prendere delle decisioni consapevoli per limitarlo a un valore massimo che dipenda dalla capacità del team stesso.

- 2) Gestire i colli di bottiglia: quando la domanda supera la capacità disponibile si creano dei colli di bottiglia, cioè il valore in uscita al flusso diventa più basso di quello che potrebbe essere realmente. Sono quindi molto pericolosi perché creano uno squilibrio che da una parte blocca i processi a monte e dall'altra non alimenta a sufficienza i processi a valle che quindi performano meno della loro capacità. È fondamentale quindi intervenire tempestivamente per ripristinare la normale capacità dell'intero flusso mediante l'aggiunta di risorse e l'identificazione di alternative.

- 3) Ridurre al minimo i passaggi e le dipendenze: in una realtà complessa dove più team lavorano insieme, può accadere che il lavoro di uno non possa procedere fintanto che quello di un altro non è terminato. Questo crea delle dipendenze e dei passaggi che rischiano di generare tempi morti poco in linea con l'idea di flusso continuo. È bene quindi che i team dispongano di tutte le competenze necessarie per essere i più autonomi possibili nel proprio lavoro non dovendo così dipendere dagli altri. Anche la possibilità di prendere decisioni in completa autonomia riduce notevolmente l'attesa di risposte da soggetti esterni al team.

- 4) Ottenere feedback più velocemente: ottenere feedback positivi e negativi già in fase di sviluppo è molto prezioso. È possibile ottenere riscontri più rapidi mediante l'utilizzo del ciclo di apprendimento Plan-Do-Check-Adjust (PDCA) e ponendo sempre al centro l'utente finale, coinvolgendo il più possibile i clienti in tutti i processi. Inoltre, favorire il focus del team mediante obiettivi chiari su piccole attività, ne velocizza il lavoro, promuovendo una cultura che affronta immediatamente i problemi quando si verificano piuttosto che accumularli. I feedback permettono sia di comprendere se si sta realizzando un prodotto in linea con le aspettative degli utenti, sia se le tecnologie in uso siano effettivamente le più adatte.

Insomma, un riscontro costante e rapido è vitale per garantire la qualità all'interno del flusso produttivo.

- 5) Lavorare in piccoli lotti: lavorare con lotti più piccoli permette di avere riscontri più rapidamente e favorisce un miglioramento continuo riducendo anche il carico di lavoro. L'intero sistema viene attraversato più velocemente da elementi più piccoli e la probabilità di andare incontro a variazioni diminuisce. La dimensione ideale di un lotto da lavorare dipende sia dai costi di mantenimento, come, per esempio, i ritardi nelle consegne, sia dai costi di transazione, cioè quelli di preparazione e di sviluppo. Favorire l'automazione della pipeline di *Continuous Delivery* riduce i costi di transazione e migliora l'intero flusso lavorativo.
- 6) Ridurre la dimensione delle code: è essenziale per migliorare il flusso ed effettuare consegne più veloci andare a ridurre la lunghezza delle code di lavoro. Una coda lunga introduce sprechi e ritardi, allungando il tempo di attesa per altre nuove funzionalità. Maggiore è la coda e maggiore sarà il tempo di attesa. Quindi è meglio avere delle code che siano corte per garantire che il flusso di lavoro sia più rapido, con meno ritardi e con una maggiore prevedibilità. Se un team è in grado di realizzare una funzionalità al mese e la coda risulta lunga quattro, il cliente dovrà attendere per ben quattro mesi prima che nuove funzionalità possano essere prese in considerazione.
- 7) Ottimizzare il tempo di concentrazione: quando un team effettivamente è concentrato su un'attività si ottiene il massimo dal lavoro svolto. Tuttavia, non è semplice in un ambiente lavorativo mantenere il più a lungo possibile uno stato di massima concentrazione visto che il più delle volte si è esposti a interruzioni continue per i più diversi motivi. Sicuramente un minore carico di lavoro facilita la riduzione delle distrazioni favorendo

un focus maggiore senza dover saltare da un argomento all'altro in poco tempo. Massimizzare il tempo in cui si è realmente concentrati su un'attività aiuta a favorire il flusso continuo e una maggiore tranquillità del team impegnato nei lavori.

8) Rimediare alle politiche e alle pratiche precedenti: per una trasformazione Lean-Agile è importante che si vada oltre alle pratiche adottate precedentemente e che non sempre garantiscono la buona riuscita di nuovi modelli. È fondamentale che si riconoscano quali siano e che si proceda il prima possibile alla loro sostituzione in favore di nuove pratiche che facilitino la trasformazione.

7. Applicare la cadenza, sincronizzarsi con la pianificazione: abbiamo visto anche con Scrum di come il lavoro possa essere organizzato secondo cicli cadenzati da eventi ben precisi. A più alto livello, lo Sprint è proprio quell'elemento che fa sì che il lavoro venga svolto con una certa cadenza. Quando il numero dei team aumenta, risulta molto importante che la cadenza prenda le forme di un segnale di clock che permetta la corretta sincronizzazione di tutti i team nel loro lavoro. Un ritmo costante e sincronizzato permette un migliore coordinamento e prevedibilità del lavoro. In SAFe, per esempio, i Program Increment fanno sì che i team dispongano di obiettivi comuni e pianifichino cicli di lavoro opportunamente allineati. Si possono prevedere degli allineamenti mensili o trimestrali tra i diversi team in modo tale che si possa verificare che tutte le parti si stiano muovendo all'unisono senza perdere il ritmo. In questo modo si riduce ampiamente la deviazione dal piano perché ciclicamente ci si confronta sullo stato di avanzamento e si applicano correzioni per evitare di allontanarsi eccessivamente dall'obiettivo finale. Inoltre, la sincronizzazione favorisce la parallelizzazione del lavoro, in un'ottica di maggiore efficienza e di riduzione dei tempi di lavorazione anche per progetti notevolmente ampi e articolati.

8. Sbloccare la motivazione intrinseca dei lavoratori: disporre di un team di lavoro motivato e autonomo è un grande vantaggio perché significa avere a disposizione un gruppo che lavora meglio e produce risultati che sono di qualità maggiore. Risulta quindi fondamentale trovare il modo migliore per motivare i lavoratori nella loro creatività, proattività e produttività. Spetta ai leader essere dei facilitatori in tal senso per garantire una buona motivazione dei dipendenti. Una distinzione che è bene fare sulla motivazione permette di identificare due tipologie di motivazione: quella intrinseca e quella estrinseca. La motivazione intrinseca è molto improntante in un contesto come quello Lean-Agile in cui il coinvolgimento e la partecipazione attiva dei componenti dei team hanno un ruolo fondamentale. Possiamo identificare tre aspetti che alimentano la motivazione intrinseca:

- 1) **Autonomia:** dare ai membri del team la possibilità di prendere delle decisioni e di organizzare il lavoro in autonomia stimola l'iniziativa e il senso di responsabilità. Il fatto che una persona possa fare le proprie scelte genera un certo grado di soddisfazione ed è quindi compito dei leader quello di preparare un ambiente che, entro certi limiti, consenta ai dipendenti di beneficiare di un certo grado di autonomia. Dall'altra parte, va osservato che spetta anche al dipendente di cogliere l'occasione e di mettere in campo il proprio spirito di iniziativa, di aprirsi al dialogo con colleghi che hanno competenze diverse per mettersi in gioco arricchendo il proprio sapere e di allinearsi costantemente con i propri leader.

- 2) **Padronanza:** una persona è interessata a vedere che le proprie competenze possano ampliarsi costantemente, godendo quindi della soddisfazione dei risultati personali e del progresso. Si tratta di una vera e propria spinta appagante che facilita la crescita in modo strettamente legato alla curiosità, all'interesse e al divertimento. Per i membri di un team, prendere parte a progetti stimolanti e ricevere un feedback costante fa sì che aumenti la voglia di migliorare. È quindi importante che l'ambiente di

lavoro promuova una mentalità di crescita, in cui i dipendenti con passione, duro lavoro e dedizione sentono di poter sviluppare le loro capacità. Questo è possibile anche grazie a una collaborazione con i colleghi stessi, promuovendo iniziative di gruppo e di scambio di competenze tra pari.

- 3) **Scopo:** molto importante per la motivazione intrinseca è dare significato al lavoro che viene fatto. Grazie a un maggiore contatto con i clienti diventa più semplice comprendere ciò che spinge verso il raggiungimento di certi obiettivi e lavorare in un team in cui si condividono gli obiettivi con gli altri membri crea un forte senso di appartenenza. Ciò motiva l'intero team ad avanzare coeso e a superare le sfide per raggiungere uno scopo ultimo che ha una certa importanza e rilevanza.

Vi è poi la motivazione estrinseca che è spinta da fattori esterni alla persona come un opportuno stipendio, i premi e i riconoscimenti particolari per il lavoro svolto. Tuttavia, è importante notare come questi incentivi debbano essere bilanciati opportunamente e non risultano così efficaci come quelli intrinseci nel lungo termine. Infatti, diversamente dalla motivazione intrinseca, quella estrinseca rischia di distogliere l'attenzione del dipendente dall'obiettivo principale, cioè il successo del progetto, in favore del mero ottenimento di un premio. È bene quindi mantenere un equilibrio tra incentivi estrinseci e intrinseci per favorire l'impegno personale e il miglioramento continuo, con un ritorno a lungo termine.

- 9. Decentralizzare il processo decisionale:** un aspetto molto importante nel framework SAFe è la decentralizzazione delle decisioni. Per garantire una maggiore reattività è bene che le decisioni vengano prese dai team che si trovano a un livello più vicino alla problematica. Infatti, questo permette non solo di accelerare i tempi decisionali, ma anche di prendere delle decisioni che sono più consapevoli e guidate dalla disponibilità di maggiori informazioni. Da un lato, quando le

decisioni vengono prese in un modo centralizzato può succedere che si creino delle code perché molteplici team si trovano contemporaneamente a necessitare dell'intervento di una figura centrale con la conseguente formazione di un collo di bottiglia. In linea con i principi finora esposti si tratta di una situazione che si vuole evitare per garantire che non vi siano ritardi e intoppi lungo tutto il flusso di lavoro. Dall'altro lato, la decentralizzazione delle decisioni può avvenire solo se si investe nella formazione e nella crescita delle competenze dei dipendenti. Solo in questo modo è possibile riporre in loro la fiducia necessaria e trarre vantaggio dalla loro preparazione tecnica che consente di prendere delle decisioni in modo sapiente. Risulta importante che vengano comunicati con chiarezza i valori e la *mission* dell'azienda in modo tale che tutte le decisioni decentralizzate siano comunque in linea con gli obiettivi generali.

Di notevole importanza per la corretta realizzazione della decentralizzazione del processo decisionale è l'ambiente che i leader sono invitati a creare per far sì che tutti i dipendenti siano incentivati a prendere decisioni con sicurezza e senza eccessive preoccupazioni. Un ambiente in cui gli errori vengono penalizzati oppure in cui vi è una chiusura nella libertà di espressione dei propri pensieri, non permette assolutamente ai team di muoversi con tranquillità nella presa di posizione. Anzi, così facendo tenderanno a delegare sempre di più qualsiasi decisione onde evitare di incorrere in spiacevoli conseguenze. È quindi necessario un ambiente completamente opposto, nel quale si promuovano gli errori come occasioni da cui imparare e che quindi dispone di una serie di contromisure atte a contenere le possibili conseguenze di scelte potenzialmente meno corrette. Il dialogo e l'apertura al confronto devono essere garantiti per aumentare la sicurezza e la confidenza di ogni membro del team e dell'intera organizzazione così da affidare loro la responsabilità delle decisioni quotidiane, riducendo i tempi di attesa, migliorando la risposta ai cambiamenti e incentivando l'autonomia. Si pensi, per esempio, a un team che si occupa dello sviluppo del front-end di un applicativo aziendale. Grazie a una decentralizzazione decisionale saranno più autonomi nella scelta delle tecnologie da utilizzare garantendo una maggiore reattività al

costante progresso tecnologico che vede la nascita di nuove tecnologie migliori in tempi brevi, le quali potranno essere adottate molto più rapidamente. In questo modo, le decisioni più operative quotidiane vengono delegate ai team più vicini al contesto specifico nel quale risultano più preparati e informati, restando comunque in linea con le decisioni centralizzate strategiche a lungo termine.

10. Organizzarsi intorno al valore: la gestione di un'azienda di grandi dimensioni è di notevole complessità. Si tratta di disporre di diverse strutture di gestione che si occupano delle operazioni, delle risorse umane, della finanza, della governance e di tutte le altre responsabilità aziendali. Non va però dimenticato che se l'azienda è cresciuta così tanto nel tempo è perché inizialmente si è concentrata opportunamente sulla realizzazione di uno o più prodotti che sono stati accolti con entusiasmo dai clienti. È quindi molto importante che man mano che aumenta la complessità aziendale, non venga meno la cura e l'attenzione alla qualità e al valore dei propri prodotti. In quest'ottica SAFe si propone di organizzare i team attorno ai flussi di valore in modo tale che vi sia una certa attenzione e focalizzazione sulla creazione del valore per il cliente finale. In tal senso, vi è una promozione di team cross-funzionali che dispongano di competenze diverse in grado di gestire l'intero flusso di lavoro per un determinato prodotto dalla sua ideazione allo sviluppo, dai test al rilascio.

A partire da quest'ultimo principio, è possibile presentare quali sono gli strumenti introdotti dal framework e che più praticamente rendono possibile l'ottimizzazione del lavoro in aziende molto grandi e con una struttura organizzativa articolata. SAFe per potersi organizzare intorno al valore utilizza tre componenti annidate:

- costruire portafogli tecnologici di flussi di sviluppo del valore,
- realizzare dei flussi di valore grazie agli Agile Release Train (ART) incentrati sul prodotto,
- creare dei team Agile in grado di fornire direttamente valore.

Per poter creare dei portafogli tecnologici di flussi di valore è fondamentale avere a mente quelle che sono le caratteristiche della mentalità lean e Agile così come sono presentate nel primo capitolo di questa tesi. Infatti, grazie all'adozione di questa mentalità è possibile garantire che vi sia una corretta focalizzazione sul prodotto e sul cliente. Solo in questo modo è possibile creare dei prodotti che siano realmente in linea con le aspettative e che quindi risultino realmente di valore per il cliente finale. Un flusso di valore è costituito da tutti quei passaggi, quelle risorse e quelle informazioni necessarie per apportare del valore a quanto viene rilasciato al cliente. Permettono di far scorrere il valore attraverso i dipartimenti di una azienda come se fosse un sistema unico, così come visto nel secondo principio, per poi farlo fluire verso il cliente. In SAFe si possono trovare due tipologie di flussi di valore: flussi di valore operativi e flussi di sviluppo del valore. I primi hanno una struttura che dipende fortemente dall'obiettivo e dalla tipologia di valore che si intende consegnare, mentre i secondi hanno una forma standard. Permettono, a seguito della richiesta di una nuova funzionalità, di ottenere un aumento di valore al termine del flusso, passando attraverso le seguenti fasi: *define, build, validate, release*. Questo flusso si ripete per l'intera vita del prodotto e permette quindi di accogliere nuove richieste e produrre aggiunte di valore iterativamente. Un portafoglio SAFe è quindi un insieme di flussi di sviluppo del valore opportunamente allineati per riuscire a fornire ai

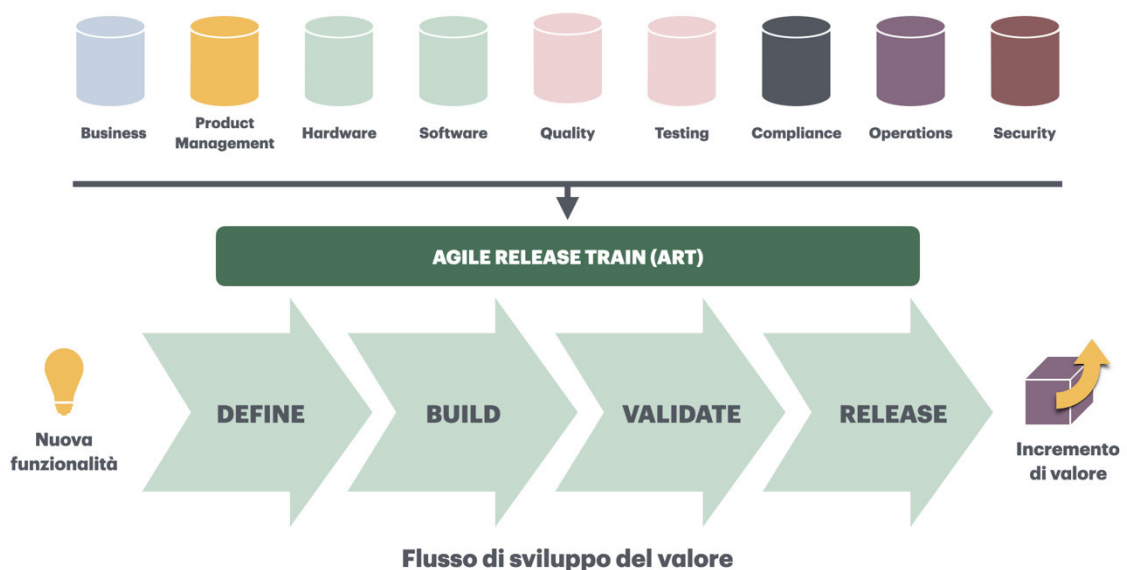


Figura 7 - Flusso di sviluppo del valore e ART in SAFe

clienti i prodotti e i servizi richiesti, in modo tale che l'intera azienda sia correttamente focalizzata sulla produzione di valore nel più breve tempo possibile. Lavorare quindi alla creazione di un portafoglio aziendale per flussi di valore permette di avere una visione d'insieme che metta al centro i prodotti e i clienti su scala aziendale, allineando le strategie e l'esecuzione grazie alla trasparenza sul lavoro svolto. I risultati possono essere misurati più facilmente mediante l'uso di KPI (*Key Performance Indicator*) basati direttamente sull'output dei flussi di valore. Questo grazie anche alla riduzione delle dimensioni dei singoli incrementi che ottimizza il flusso di lavoro andando a rendere più evidente il valore aggiunto che ogni ciclo di lavoro sarà in grado di produrre.

Un obiettivo che deve essere inseguito da un'azienda è quello di ridurre il tempo che intercorre dalla richiesta di una nuova funzionalità al rilascio della stessa. Detta in altri termini, si vuole ridurre la durata di un flusso di sviluppo del valore. Per fare ciò SAFe introduce il concetto di *Agile Release Train* (ART), un modo per organizzare al meglio le persone e le risorse così da ridurre al minimo le dipendenze e i tempi morti all'interno del flusso. È proprio l'ART che rende possibile l'estensione di quanto visto precedentemente per i piccoli team Agile. Infatti, un ART è un team più grande composto da tanti piccoli team Agile. Quando un singolo team non può sostenere il lavoro per la realizzazione di un prodotto più complesso allora si organizzano degli ART appositi, cioè dei team di team, composti da circa 50-125 persone, che sono opportunamente allineati e che collaborano per raggiungere obiettivi comuni in modo fluido e sincronizzato. Come abbiamo visto per i team Scrum, anche gli ART basano il proprio lavoro su una lista di funzionalità richieste dai clienti che prende il nome di ART Backlog, per creare un flusso continuo di analisi, sviluppo, test e rilascio di valore. È interessante notare che, rispetto a un approccio tradizione in cui i diversi dipartimenti aziendali lavorano separatamente ed è quindi necessario attendere risposte dai diversi reparti ogni volta che vi è la necessità, in un ART vi sono direttamente tutte le figure con le competenze necessarie per garantire la massima autonomia dall'ideazione alla messa in produzione di una nuova funzionalità. Non è più necessario che un team attenda le risposte dai diversi dipartimenti, ma al suo interno avrà già tutte le risorse necessarie per rispondere autonomamente alle questioni che emergeranno durante il flusso di sviluppo del valore. Ogni

dipartimento aziendale è quindi organizzato in team Agile che utilizzano le diverse tecniche viste precedentemente, come Scrum e Kanban, per far sì che si possano realizzare interi prodotti come frutto del lavoro comune all'interno di un ART. Per far sì che ciò avvenga nel modo migliore è necessario disporre di alcune figure che si occupano del coordinamento dei diversi team che fanno parte degli ART.

- 1. Product Management:** generalmente una o più persone responsabili di definire ciò che andrà realizzato in base alla mission aziendale e agli elementi presenti nell'ART Backlog. Collaborano con i Product Owner dei vari team e con i clienti.
- 2. System Architect:** generalmente una persona o un team che definisce l'architettura del sistema. Definiscono gli standard operativi del sistema, le sue componenti e il modo in cui sono interconnesse e di conseguenza come interagiscono.
- 3. Release Train Engineer (RTE):** generalmente una persona che si occupa di rendere possibile il normale fluire del flusso di sviluppo del valore, rimuovendo gli ostacoli, gestendo le dipendenze e facendo sì che la velocità del flusso sia elevata.
- 4. Business Owner:** generalmente una o più persone che sono i diretti responsabili dei risultati di business di un ART. Sono gli stakeholder e si occupano del ROI (*Return of Interest*), della governance e della conformità.

Così come gli eventi in Scrum hanno una grande importanza per far sì che il lavoro venga svolto al meglio, in una logica di miglioramento continuo, anche in SAFe vi sono degli eventi che permettono prima di tutto di far sì che gli eventi a livello di singolo team Agile si allineino a quelli degli altri, creando così un unico e grande sistema ben orchestrato che si muove in modo compatto e armonico. All'interno di un ART, il tempo è scandito dai *Planning Intervals* (PI) che hanno generalmente una durata dalle otto alle dodici settimane e contengono al loro interno diversi cicli di sviluppo così come analizzati in Scrum. In ogni PI quindi i team Agile eseguono i propri Sprint in modo coordinato con quelli degli

altri team e ogni PI è a sua volta un più ampio ciclo di lavoro scalato su più team che appartengono all'ART. Grazie a un'opportuna pianificazione dei PI tutti i partecipanti di un ART si allineano con quelle che sono le priorità legate al business e sono quindi in grado di selezionare il proprio lavoro sulla base della capacità e delle dipendenze necessarie per raggiungere gli obiettivi di un Planning Interval. Il termine di ogni ciclo all'interno di un PI è caratterizzato da un evento importante che prende il nome di System Demo. In questo evento l'RTE si occupa di mostrare agli stakeholder l'integrazione delle nuove funzionalità sviluppate dai team Agile. È così possibile misurare concretamente l'avanzamento verso gli obiettivi del PI e raccogliere gli opportuni feedback per intervenire immediatamente su possibili disallineamenti. A differenza delle Sprint Review che riguardano il singolo team, le System Demo coinvolgono tutti i team di un ART e tutti gli stakeholder. Al termine di un PI vi è un evento molto importante chiamato *Inspect & Adapt* durante il quale tutti gli stakeholder insieme ai team che compongono un ART riflettono sugli avanzamenti e identificano ciò che può essere migliorato. Mediante una System Demo dell'intero Planning Interval, una retrospettiva e un workshop incentrato sul problem-solving, si cerca di trovare le problematiche più impattanti così da risolverle per rendere sempre migliore il lavoro svolto in un ART.

Quando la complessità aumenta, SAFe prevede la messa in campo di quelli che vengono definiti *Solution Train*. Si tratta di un insieme di ART che collaborano in modo sincronizzato per raggiungere obiettivi ancora più complessi e sfidanti, dimostrando come ancora una volta sia possibile scalare quanto detto finora per adattarsi anche alle realtà organizzative più grandi e articolate. Infine, non bisogna dimenticare che tutto ciò è reso possibile grazie alla presenza dei team Agile che sono quindi il cuore di tutta questa struttura annidata.

2.3 IL MODELLO SPOTIFY

A conclusione di questo secondo capitolo in cui ci siamo concentrati su quali sono i possibili strumenti che possono aiutare le aziende, dalle più piccole alle più complesse, nella messa in pratica di una mentalità lean e Agile, è interessante dare una breve visione a quella che è stata una vera e propria adozione di una mentalità piuttosto che la definizione di un framework: il modello Spotify. Molto probabilmente non sono necessarie grandi presentazioni, tuttavia, è bene sapere che si tratta di un servizio sviluppato dall'azienda svedese Spotify AB che fornisce lo streaming on demand di brani musicali delle più disparate case discografiche di tutto il mondo, ma anche di podcast e di audiolibri. Il servizio nasce nel 2006 e viene lanciato sul mercato ufficialmente nel 2008. Negli anni vede una crescita esponenziale che lo porterà ad avere in soli 14 anni, nel 2022, un totale di 422 milioni di utenti con una crescita di +19% su base annua e +4% rispetto al quarto trimestre del 2021²⁴. Insomma, non stiamo parlando di una piccola startup, ma di una realtà che attualmente rappresenta un esempio di azienda molto ampia e articolata. Nel 2012 Henrik Kniberg e Anders Ivarsson pubblicano un paper intitolato *Scaling Agile @ Spotify* con l'obiettivo di fornire una descrizione del lavoro fatto all'interno dell'azienda per far sì che i principi Agile potessero essere applicati in una realtà complessa nella quale diversi team si trovano a collaborare per la costante evoluzione di un prodotto. Prima di addentrarci nei dettagli di quanto presentato, è bene tenere a mente quanto sottolineato dagli autori stessi:

*"We didn't invent this model. Spotify is (like any good agile company) evolving fast. This article is only a snapshot of our current way of working - a journey in progress, not a journey completed. By the time you read this, things have already changed."*²⁵

²⁴ Filippetti A., *Il (non) modello Spotify*, in Agile Plaza, 9 settembre 2022, disponibile online all'indirizzo <https://agileplaza.it/modello-spotify/> (ultimo accesso in data 20/10/2024).

²⁵ Kniberg, H., Ivarsson, A., *Scaling Agile @ Spotify*, October 2012, disponibile online all'indirizzo <https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf> (ultimo accesso in data 20/10/2024).

Infatti, qui l'obiettivo non è quello di presentare il modello Spotify come un possibile strumento da adottare per organizzare il lavoro di un'azienda articolata, ma è curioso comprendere come un'azienda che è cresciuta notevolmente in soli quattro anni (dal lancio sul mercato di Spotify nel 2008 al 2012, anno di pubblicazione dell'articolo) sia stata in grado di continuare ad abbracciare i principi agili anche contando di un numero sempre più grande di team.

L'unità base che si occupa dello sviluppo prende il nome di *Squad* ed è organizzata come uno Scrum Team. I componenti dei team dispongono di tutte le competenze e gli strumenti necessari per occuparsi di tutto il processo produttivo, dall'ideazione al rilascio in produzione delle differenti funzionalità. Non vengono imposte delle modalità con cui organizzano il proprio lavoro e sono quindi invitati a muoversi in autonomia. Ogni Squad ha un obiettivo a lungo termine che riguarda una parte del prodotto intero. Per esempio, una Squad si può occupare del sistema di riproduzione audio, un'altra delle soluzioni per il pagamento degli abbonamenti, e così via. In questo modo, ogni Squad si concentra su un blocchetto specifico che compone l'intero prodotto e acquisisce nel tempo una profonda conoscenza su tutti gli aspetti funzionali e tecnici di quel blocchetto. L'ambiente di lavoro è stato opportunamente adeguato per favorire il lavoro in squadra e la collaborazione tra tutti i colleghi. In più, vista l'importanza data all'apprendimento continuo e all'innovazione, ogni Squad è invitata a spendere parte del suo tempo in quelli che vengono chiamati *hack days*, cioè giornate in cui i dipendenti sperimentano nuove tecnologie e propongono nuove idee tutti insieme. Questo è un aspetto molto interessante perché anche se effettivamente non si sta lavorando al prodotto direttamente, queste giornate sono molto utili per stimolare la curiosità e scoprire le novità tecnologiche che possono apportare effettivamente del valore aggiunto al prodotto in una fase successiva. Così come visto per Scrum, in Spotify non vi sono dei leader all'interno delle Squad, ma un ruolo importante è giocato dal Product Owner che ha la responsabilità di dare una priorità al lavoro che deve essere svolto e si allinea costantemente con i Product Owner delle altre Squad per garantire che l'intera organizzazione si stia muovendo correttamente verso gli obiettivi aziendali. Il Product Owner non dà alcuna direttiva sulle modalità con cui verranno realizzate le diverse funzionalità, ma è responsabile di aggiornare e

prendersi cura del Product Backlog di ogni Squad. Importante per le Squad è anche la figura dell'Agile Coach che è un facilitatore del lavoro di squadra e che aiuta quindi a migliorare il lavoro sia seguendo ogni membro dei team con dei *one-to-one*, sia gestendo i vari eventi che permettono di analizzare l'andamento del lavoro. È interessante notare come ogni Squad di base non ha bisogno di interagire con le altre e vive l'esperienza lavorativa come se fosse una piccola startup. Sempre in linea con l'importanza che si dà all'analisi dell'andamento del lavoro, vengono proposti periodicamente dei sondaggi per comprendere se vi sono degli interventi necessari da parte dell'azienda che potrebbero migliorare il lavoro di ogni Squad. In questo modo è possibile comprendere quali sono le criticità che ogni team sta riscontrando e si può intervenire di conseguenza. Per esempio, possono emergere criticità sui rilasci, oppure un team può segnalare di avere delle perplessità sugli obiettivi e sul carico di lavoro. Insomma, un sistema di monitoraggio interno molto prezioso.

A loro volta, le Squad che lavorano in un'area affine fanno parte di una *Tribe*. Generalmente i vari membri di una Tribe si trovano negli stessi uffici così da facilitare il confronto tra le diverse Squad in un ambiente che favorisce l'incontro e lo scambio. Le Tribes non possono avere un numero maggiore di cento membri altrimenti le relazioni sarebbero più complesse da gestire. Ogni Tribe ha un leader che è responsabile di facilitare il lavoro delle Squad che ne fanno parte e organizzano regolarmente degli incontri informali in cui presentare a tutti i membri appartenenti alla Tribe il lavoro che si sta svolgendo così da stimolare la curiosità ed essere fonte di ispirazione per gli altri. Può succedere che in alcuni casi più Squad si trovino a collaborare e quindi in certi casi a dipendere dal lavoro di un'altra Squad. Costantemente vengono fatti dei sondaggi per capire se vi siano delle dipendenze bloccanti e si cerca di capire come far sì che questo si verifichi il meno possibile. È curioso notare come sia stata rilevata una forte dipendenza tra le Squad e la parte di *operations* con conseguenti rallentamenti. Con Operations in un'azienda ci si riferisce alle attività e ai processi necessari per mantenere e gestire i sistemi, l'infrastruttura e i servizi che supportano l'intero ciclo produttivo. Si pensi, per esempio, alla gestione dei server aziendali, delle reti e dei database. Essendoci una forte dipendenza da parte delle Squad con la parte di Ops, si è previsto un apposito team operations che si

occupa di supportare le diverse Squad nei rilasci e nelle attività che impattano l'infrastruttura. Il team Ops non si sostituisce nelle attività delle Squad, ma fa in modo di assisterle così da facilitare i rilasci e da fornire il supporto di cui necessitano.

Sebbene Squad e Tribe godano di una forte autonomia e indipendenza è comunque vantaggioso che i membri di una Squad comunichino con i membri di altre Squad. In tal senso, Spotify ha introdotto i *Chapter*, cioè gruppi di persone che hanno abilità simili e che lavorano nella stessa area di competenza, all'interno della stessa Tribe. Un Chapter potrebbe includere per esempio tutti i tester, oppure tutti gli sviluppatori web, o ancora tutti gli sviluppatori mobile. I componenti di un Chapter si incontrano regolarmente per discutere di tematiche relative al loro lavoro e delle sfide che hanno affrontato, in modo tale da condividere con gli altri alcune informazioni che potrebbero essere molto preziose per migliorare il lavoro di tutti nelle diverse Squad a cui appartengono. Si noti come ogni Chapter ha un leader che generalmente è un manager che lavora comunque in una Squad esattamente come tutti gli altri. Infine, sempre per garantire uno scambio di punti di vista tra tutti i membri dell'azienda, troviamo le *Guild*. Si tratta di un gruppo di persone più ampio rispetto ai Chapter che coinvolge persone da tutte le Tribe dell'intera organizzazione che vogliono condividere competenze, strumenti e pratiche. Per esempio, tutti gli sviluppatori web formano la web technology guild.

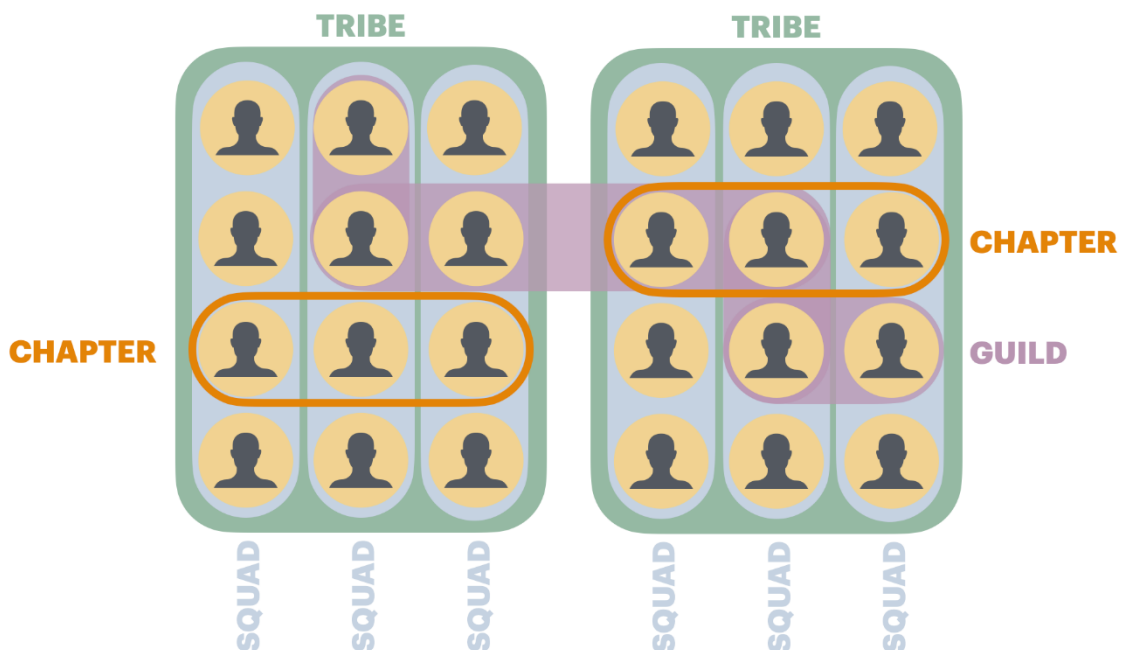


Figura 8 - Il modello Spotify: Squad, Tribe, Chapter e Guild

Infine, dal momento in cui il lavoro di ogni Squad può impattare molteplici sistemi che compongono l'intera architettura, è stato definito il ruolo di *System Owner* per gestire e mantenere la qualità tecnica e l'integrità dei vari sistemi. I *System Owner* sono generalmente due persone per ogni sistema e sono dei membri delle Squad che oltre a svolgere il loro lavoro consueto, dedicano parte del loro tempo anche a monitorare e organizzare il sistema, avendo cura di supervisionare le modifiche che vengono apportate onde evitare conflitti o inefficienze. In più, è presente anche un *Chief Architect* che si occupa di supervisionare le questioni architetturali a un livello più alto con un occhio di riguardo a tutti i sistemi. Si occupa quindi di esaminare lo sviluppo di nuovi sistemi assicurandosi che si evitino errori comuni e che si rispetti la visione architetturale dell'azienda. I suoi sono comunque da considerarsi dei consigli e dei suggerimenti in quanto la decisione finale spetterà alla Squad responsabile della realizzazione del nuovo sistema, in linea con il principio di autonomia dei team.

Notiamo quindi come in Spotify si sia ricercato il modo migliore di organizzare il lavoro anche su larga scala in linea con i principi Agile. La struttura flessibile che è stata messa in piedi permette di bilanciare opportunamente l'autonomia e il coordinamento tra i team facilitando quindi il miglioramento continuo, la produzione di valore e la riduzione dei tempi di attesa anche in contesti più complessi. Il modello organizzativo presentato è stato il frutto di un'osservazione approfondita della realtà aziendale e non una mera applicazione di regole prestabilite. Grazie a una riflessione accurata sulle esigenze specifiche dell'azienda e del contesto in cui opera, è stato possibile raggiungere un ottimo livello di agilità e adattabilità che non può che essere un buon punto di arrivo intermedio da cui però ripartire verso un continuo processo di affinamento e di miglioramento delle tecniche e pratiche adottate. Conoscere gli strumenti per agire Agile e avere un pensiero Agile permette di adeguarli al meglio per ogni realtà aziendale, rendendoli realmente efficaci e ottenendo il massimo valore dalla loro adozione sapientemente e costantemente modellata.

3. LESSONS LEARNED: IL CASO SANTANDER CONSUMER

BANK

In un contesto organizzativo, analizzare le esperienze passate per comprendere quali sono state le criticità o i punti di forza è molto importante. Rappresenta infatti la possibilità di migliorare i processi, le pratiche e le metodologie basandosi su ciò che è stato fatto fino a quel momento. Si tratta di disporre di un feedback concreto che si basa su quanto è stato fatto e che diventa sempre più utile nel tempo arricchendosi costantemente di nuove conoscenze acquisite esperienza dopo esperienza. Un approccio di questo tipo si concretizza nel concetto di *lessons learned* che vede le sue radici nelle metodologie di *Project Management* e di *Knowledge Management* così come formalizzate con l'espansione di standard internazionali come il *Project Management Body of Knowledge - PMBOK*²⁶ sviluppato dal Project Management Institute (PMI). Le *lessons learned* sono tutte quelle conoscenze che vengono acquisite a seguito di progetti e di attività precedenti e vengono identificate come una pratica centrale per il miglioramento continuo. Diventa molto cruciale raccogliere e documentare in ogni fase del progetto, dal suo avvio alla sua chiusura, quelle che sono le lezioni apprese per far sì che in progetti futuri si evitino errori già commessi in passato e si possano invece replicare i successi.

Questo concetto ha trovato la sua applicazione anche all'interno delle metodologie Agile e lean che sono state presentate precedentemente. Si pensi, per esempio, alle Sprint Retrospective di Scrum durante le quali il team si riunisce proprio per riflettere su ciò che nell'ultimo Sprint è andato bene, sugli ostacoli incontrati e sui possibili miglioramenti. In questo modo si dispone di feedback costanti che possono essere trasformati in azioni correttive da intraprendere nel ciclo di lavoro successivo così da favorire il miglioramento continuo e la crescita del team. Ritroviamo lo stesso approccio anche nel lean thinking dove il concetto Kaizen invita a revisionare costantemente le attività, cercando così di ridurre al minimo gli sprechi e di massimizzare il valore, grazie a un miglioramento

²⁶ Project Management Institute, *A Guide to the Project Management Body of Knowledge: PMBOK Guide*, 3ª edizione, Newtown Square, Project Management Institute, 2004.

continuo. Quindi quella delle *lessons learned* è diventata una pratica fondamentale per evitare la ripetizione degli errori e per rafforzare la consapevolezza del team su quelle che sono le *best practices* da adottare.

In tal senso, l'esperienza di tirocinio e di tesi in azienda presso Santander Consumer Bank Italia è stata fondamentale per poter osservare dall'interno quelle che sono le criticità e le complicità che un'azienda complessa e articolata deve affrontare costantemente. Le *lessons learned* durante i sei mesi di lavoro come Product Owner, affiancando inizialmente alcuni membri esperti del team, hanno permesso di comprendere più approfonditamente le dinamiche interne dell'organizzazione e i processi di gestione e di sviluppo dei progetti IT. Concentrandosi principalmente sul progetto chiamato *Nuova Area Riservata – Login e registrazione*, sono emerse sfide legate alla gestione dei progetti con il modello Waterfall tradizionalmente utilizzato, a causa della sua rigidità che lo rende meno flessibile nella gestione di eventuali imprevisti come quelli che sono sopraggiunti e che sono presentati in questo capitolo. L'esperienza è stata anche l'occasione di vedere da vicino quanto è importante testare le diverse funzionalità di un prodotto prima che vengano messe a disposizione del cliente finale, garantendo così un'ottima qualità. È emerso che la fase di testing con gli utenti (UAT – *User Acceptance Testing*) rappresenta un punto critico che ha generato dei rallentamenti per l'intero progetto e in tal senso si espongono le evidenze emerse che sottolineano l'importanza di un approccio iterativo e collaborativo, come sostenuto dalla metodologia Agile. Inoltre, è stato molto importante ritrovarsi in situazioni diverse che hanno permesso di comprendere a pieno quelle che sono le competenze fondamentali del Product Owner per far sì che svolga al meglio il proprio compito. Nel capitolo si presentano esempi di situazioni in cui la sua figura si è resa decisiva sia nelle relazioni con gli stakeholders e nella cooperazione tra i dipartimenti, sia nella massimizzazione del valore del prodotto. Un punto critico emerso sin da subito riguarda la gestione dei fornitori terzi soprattutto in un contesto come quello dell'azienda in cui il rispetto dei tempi e degli standard qualitativi richiede un coordinamento attento e trasparente di non semplice realizzazione quando il numero di fornitori e di progetti è notevolmente elevato.

L'attenta osservazione e il coinvolgimento in prima persona nelle diverse attività svolte in azienda hanno permesso di ricavare delle preziose lessons learned da considerare come il punto di partenza per migliorare i processi interni e di conseguenza l'intera organizzazione. Oltre a rafforzare l'idea di come un approccio Agile e lean possa apportare vantaggi in termini di flessibilità e di adattabilità, queste lezioni apprese suggeriscono delle vere e proprie raccomandazioni per il futuro dell'azienda che permettano di ottimizzare le risorse e aumentare il valore generato per il cliente, rispettando i vincoli normativi e organizzativi essenziali in un contesto come quello del settore bancario e dei prodotti di credito al consumo.

3.1 BANCO SANTANDER E SANTANDER CONSUMER BANK

Banco Santander è un gruppo bancario internazionale fondato nel 1857 con sede in Spagna. Nel corso degli anni, è cresciuto fino a diventare una delle principali istituzioni finanziarie a livello globale, con una presenza significativa in dieci mercati principali, in Europa, in Nord America e in Sud America. Si tratta di una delle banche più grandi al mondo per capitalizzazione di mercato. Il gruppo bancario Santander conta oltre 200.00 dipendenti e circa 157 milioni di clienti, oltre che 4 milioni di azionisti. Il brand Santander è uno dei più valutati al mondo, nominato il marchio spagnolo di maggior valore nelle classifiche Brand Finance Europe 500 e Brand Finance Global 500 del 2024. Da oltre quindici anni rientra tra i primi cento marchi di maggior valore nel pianeta nella classifica Best Global Brands di Interbrand²⁷. La mission del brand è quella di aiutare le persone e le imprese a prosperare, in linea con la vision di essere la migliore piattaforma aperta di servizi finanziari, agendo in modo responsabile e guadagnano la fiducia dei clienti e degli azionisti. Il lavoro che viene svolto quotidianamente è fondato su tre valori principali che garantiscono che ciò che viene fatto sia semplice, personale e giusto. Negli anni si è creata una vera e propria cultura aziendale fondamentale per la crescita costante anno dopo anno basata su cinque comportamenti: *think customer, embrace change, act now, move together, speak up*. È interessante notare, che questi comportamenti rispecchiano alcuni dei principi Agile e lean presentati nei capitoli precedenti, indicando come in Santander sia molto importante una cultura che sia favorevole al cambiamento e al miglioramento continuo, ponendo al centro il cliente finale e la proattività dei dipendenti, in un ambiente che favorisca il lavoro di gruppo. Nello specifico, i cinque comportamenti sono:

- think customer: è importante prendersi cura dei clienti per ottenere la loro fiducia e fidelizzazione,
- embrace change: è opportuno accogliere nuove sfide e nuovi modi di agire come opportunità di crescita,

²⁷ Banco Santander, *We are proud to be one of the 100 most valuable brands in the world*, 21 ottobre 2024, disponibile online all'indirizzo <https://www.santander.com/en/stories/we-are-proud-to-be-one-of-the-100-most-valuable-brands-in-the-world> (ultimo accesso in data 02/11/2024).

- act now: è bene agire in modo veloce e responsabile, puntando alla semplicità,
- move together: è buona norma collaborare e lavorare insieme ai colleghi per raggiungere gli obiettivi comuni,
- speak up: è fondamentale che le persone si sentano libere di proporre nuove idee, di fare domande e di condividere e ricevere feedback.



Figura 9 - Il logo Santander

Santander è presente anche in Italia dove la sua sede principale è a Torino. Nel 1988, dieci banche private nel nord-ovest d'Italia, tra cui Intesa Sanpaolo S.p.A., fondano Finconsumo S.p.A. con l'obiettivo di assicurare una presenza sul mercato del credito al consumo. Tra il 1993 e il 1998, Intesa Sanpaolo aumenta le sue quote al 50% mentre l'altro 50% è di proprietà di una filiale del Banco Santander Central Hispano. Sarà poi tra il 2003 e il 2006 che Santander Consumer Finance acquisirà il 50% della quota azionaria detenuta da Intesa Sanpaolo, diventando così socio unico, e dando così origine all'attuale Santander Consumer Bank S.p.A.. Negli anni a seguire una serie di acquisizioni e di accordi permettono all'azienda di crescere e di affermare la sua presenza nel settore bancario e dei prodotti di credito al consumo. Nascono diverse *joint venture* come per esempio TIMFin nel 2020, di cui Santander Consumer Bank Italia detiene il 51%, oppure Hyundai Capital Bank Europe GmbH nel 2021, partecipata al 51% da Santander Consumer Bank AG e da Hyundai Capital Services per il restante 49%. Nel 2022 nascono Santander Consumer Renting e Drive, due società di noleggio a lungo termine, fondate sempre da Santander Consumer Bank Italia. Nel 2023, nasce Stellantis Financial Services Italia dalla joint venture paritetica tra Stellantis Financial Services Europe S.A. e Santander Consumer Finance S.A.. Vi sono poi molteplici accordi commerciali come quelli nel mondo dell'automotive con Piaggio, Tesla, Vespa, oppure quelli nel mondo bancario con Banca Mediolanum, Poste Italiane, e tanti altri. Di seguito sono elencati i principali prodotti di Santander Consumer Bank Italia:

- **prestito personale:** si tratta di prestiti concessi direttamente al cliente,
- **prestito finalizzato:** si tratta di prestiti concessi a persone per l'acquisto di beni (non automobili) o per la fornitura di servizi,
- **prestito auto/moto:** si tratta di prestiti concessi al cliente per l'acquisto di auto o moto sia nuove sia usate,
- **servizi assicurativi:** i clienti hanno a disposizione sono diversi servizi assicurativi a copertura dei prestiti e/o dei beni finanziati,
- **leasing:** si tratta di leasing finanziario di autovetture e di veicoli commerciali,
- **cessione del quinto:** si tratta di una tipologia particolare di prestito personale che si estingue attraverso la cessione di una parte dello stipendio o della pensione fino a un quinto del suo ammontare al netto delle trattenute,
- **conti deposito:** consentono di depositare denaro per un periodo di tempo definito, offrendo un rendimento sotto forma di interessi, spesso superiore a quello di un conto corrente tradizionale, con un rischio generalmente basso,
- **noleggio vetture:** si tratta di noleggio a lungo termine su vetture nuove o usate con opzione di flessibilità su durata e chilometri.

In Figura 10, è rappresentata la struttura organizzativa di Santander Consumer Bank Italia. Come si può notare, si tratta di un modello aziendale complesso e articolato che è bene avere a mente dal momento che, come detto precedentemente, una delle principali sfide dell'adozione di nuove metodologie, è data proprio dalla complessità di un'organizzazione. L'amministratore delegato e direttore generale è Alberto Merchiori e si trova al vertice della gerarchia composta dalle diverse funzioni e direzioni. Funzioni di supporto direzionale e di carattere trasversale sono rappresentate da Internal Audit, dalla Segreteria e Affari Generali e da Transformation. Vi sono poi le diverse direzioni che forniscono un contributo diretto nei confronti delle diverse attività essenziali dell'azienda: dalla gestione del rischio, alle vendite, dalle relazioni con i clienti, alla conformità normativa. Questa suddivisione delle funzioni permette una definizione chiara di quelli che sono i ruoli e le responsabilità, favorendo la trasparenza e ripartendo opportunamente le diverse attività cruciali. Tale struttura restituisce un modello bilanciato

tra controllo centralizzato e autonomia funzionale, ponendo un'attenzione particolare alla governance, alla gestione del rischio e all'innovazione tecnologica.

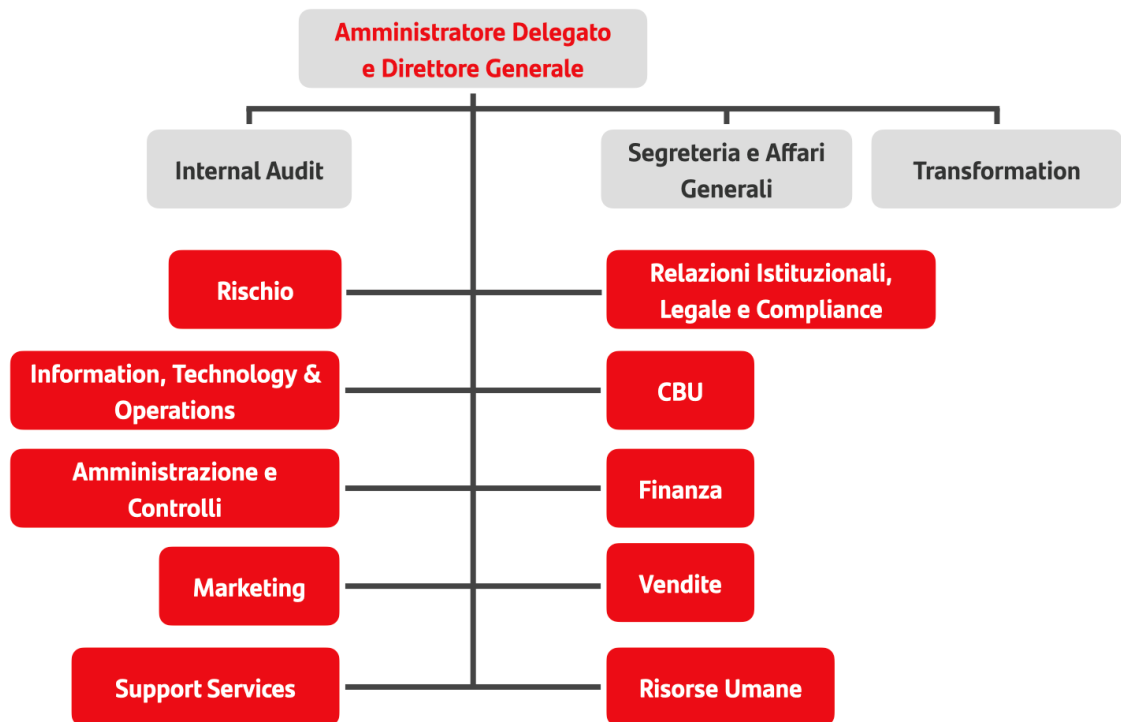


Figura 10 - Struttura organizzativa di Santander Consumer Bank Italia

L'esperienza di tesi in azienda e di tirocinio come Product Owner ha permesso di vedere più da vicino la direzione Information, Technology & Operations che ha come responsabile Flavio Glorio. A sua volta, questa direzione che risulta cruciale per il funzionamento e l'innovazione tecnologica dell'azienda, è suddivisa in diverse aree specialistiche, ognuna con un ruolo ben preciso (Figura 11). Attenzione particolare viene posta su alcuni elementi critici come la Cyber Security, fondamentale in un'azienda che ha alla base un gran numero di sistemi informatici, e l'Application Architecture, molto importante per garantire un'evoluzione e un'espansione dei servizi digitali offerti da Santander. L'area Technology include diversi team con compiti specifici, così come presentati di seguito:

- **Infrastructure & Networking:** il team si occupa della gestione e della manutenzione di tutte quelle che sono le infrastrutture fisiche e virtuali della banca, permettendo ai sistemi di rete e alle connessioni di essere stabili, sicuri e assicurando che sia garantita la continuità operativa. Si occupa di garantire che tutte le risorse

tecnologiche siano sufficienti e in grado di supportare il carico di lavoro dell'azienda e che siano opportunamente protette.

- **Release Management:** il team si occupa di pianificare, di coordinare e di gestire i rilasci di nuove versioni di software e di applicazioni. Ha la responsabilità di garantire che ogni aggiornamento o miglioramento tecnologico sia documentato e distribuito in modo opportuno, onde evitare l'interruzione dei servizi.
- **Client Owner:** i membri del team fungono da collegamento tra i team tecnici e gli utenti finali o i clienti interni dell'organizzazione, assicurandosi che quelle che sono le richieste e le esigenze vengano opportunamente recepite con le opportune priorità. Oltre a Santander stessa, ogni joint venture è seguita da un membro del team dedicato.

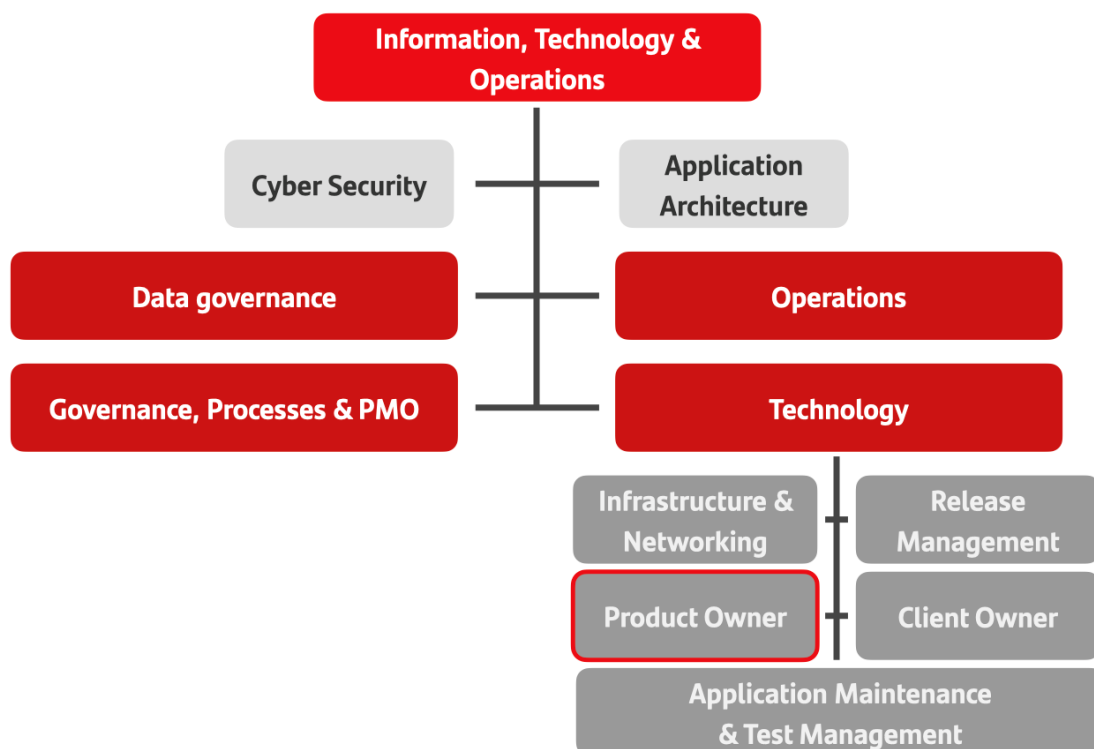


Figura 11 – Dettaglio dell'area Technology nella struttura della direzione Information, Technology & Operations di Santander Consumer Bank Italia

- **Product Owner:** ogni membro del team è responsabile della definizione e della gestione dei requisiti di prodotto, indipendentemente dalla joint venture a cui è destinato, rappresentando gli interessi degli stakeholders e monitorando che il risultato del lavoro del team di sviluppo sia in linea con gli obiettivi del business. Il team ha un ruolo fondamentale in quanto fa da filo conduttore tra i diversi team tecnici e gli utenti delle diverse direzioni presentate prima, massimizzando il valore del prodotto per l'azienda.
- **Application Maintenance & Test Management:** il team si occupa della gestione delle richieste di manutenzione delle applicazioni e dei sistemi aziendali e della preparazione e del coordinamento dei test, fondamentali per garantire che le applicazioni siano in linea con gli standard di qualità aziendali.

La mission quindi di questa direzione è quella di garantire la migliore gestione tecnico-organizzativa aziendale, coordinando e supportando la gestione dell'IT in modo tale che sia in linea con gli obiettivi strategici definiti nel *System Plan*, ottimizzando le soluzioni tecnologiche individuate e rispettando i budget assegnati. Inoltre, si impegna a coordinare e a supportare tutte quelle che sono le attività operative della banca, facendo sì che siano garantite la continuità, l'efficienza e la qualità dei processi aziendali. Ciò permette all'azienda di raggiungere determinati obiettivi e di evolversi costantemente, migliorando l'esperienza dei clienti.

3.2 IL MODELLO WATERFALL MESSO ALLA PROVA

Un modello di gestione dei progetti tradizionalmente adottato dalla maggior parte delle aziende è il cosiddetto *modello Waterfall* oppure *a cascata*. Questo modello trova la sua formalizzazione nel 1970 all'interno del paper *Managing the Development of Large Software Systems* pubblicato da Winston W. Royce nell'ambito dei progetti di ingegneria e di sviluppo software. L'autore propone una metodologia che permetta di gestire al meglio progetti molto complessi e su larga scala evidenziando sin da subito quelli che possono essere i limiti di un modello che non accetta margini di adattamento. Il flusso di lavoro viene suddiviso in una serie di fasi consecutive che descrivono un percorso ordinato per lo sviluppo, aiutando così a evitare errori derivanti dalla scarsa organizzazione. Le fasi presentate sono: raccolta dei requisiti, analisi e progettazione, implementazione, verifica e testing, manutenzione.

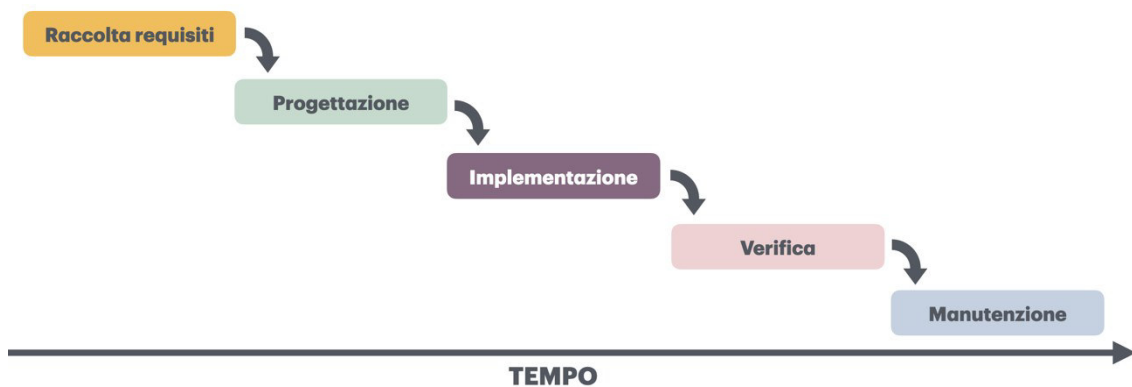


Figura 12 - Fasi del modello Waterfall

Benché vi sia una suddivisione specifica, Royce evidenzia delle possibili limitazioni di un approccio prettamente sequenziale sottolineando come non potrebbe essere adatto per un processo di sviluppo software troppo rigido in quanto non in grado di gestire la variabilità dei progetti più articolati. In tal senso, la mancanza di iterazioni tra le diverse fasi rischia di portare a costosi errori che vengono rilevati solo nelle ultime fasi quando operare una modifica comporta un intervento notevolmente più oneroso. La proposta dell'autore è quindi quella di prevedere idealmente delle iterazioni tra le fasi principali, come, per esempio, tra la progettazione e il testing. Infatti, come è stato affermato più volte, è proprio durante i test che ci si rende conto di possibili errori di progettazione o

di disallineamenti rispetto ai requisiti iniziali e la necessità di apportare delle modifiche potrebbe essere rovinosa, così come indicato nel paper:

“The required design changes are likely to be so disruptive that the software requirements upon which the design is based and which provides the rationale for everything are violated. Either the requirements must be modified, or a substantial change in the design is required. In effect the development process has returned to the origin and one can expect up to a 100-percent overrun in schedule and/or costs.”²⁸

Tuttavia, l'applicazione che comunemente viene fatta del modello Waterfall lascia ben poco spazio alla flessibilità. Nella maggior parte dei casi ci si confronta con delle sequenze rigide di passaggi da seguire in modo lineare che non prevedono dei cicli iterativi tra di essi e ogni fase successiva non può avere inizio fintanto che la fase precedente non è completamente terminata. Rispetto a quanto detto finora sulle metodologie Agile e lean ci troviamo di fronte a un'applicazione inadeguata di un modello che viene considerato solo nella sua struttura rigida che poco si adatta alle sfide di un settore in continua evoluzione come quello dello sviluppo software. Di seguito sono presentate le differenti fasi con una breve descrizione.

- 1. Raccolta requisiti:** per prima cosa si raccolgono e si definiscono quelli che sono i requisiti del progetto. In questa prima fase, gli analisti lavorano a stretto contatto con i clienti e gli stakeholder per comprendere al meglio e documentare le richieste. L'obiettivo è produrre un elenco di quelli che sono i requisiti che verranno presi in considerazione nelle fasi successive. È una fase delicata che richiede un'attenta partecipazione di tutte le persone coinvolte perché, come già detto

²⁸ Royce W. W., “Managing the Development of Large Software Systems”, in *ICSE '87: Proceedings of the 9th international conference on Software Engineering*, pp. 328 - 338, IEEE Computer Society Press, Washington DC, 1987, disponibile online all'indirizzo <https://dl.acm.org/doi/10.5555/41765.41801> (ultimo accesso in data 30/11/2024).

prima, con questo modello non sarà così semplice accogliere eventuali modifiche successivamente.

- 2. Progettazione:** si procede con la definizione dell'architettura del sistema. In questa fase si effettuano le opportune analisi tecniche per stabilire la struttura generale del sistema. Si pone attenzione anche ai dettagli andando a definire i diversi moduli del sistema, le diverse componenti e il modo in cui queste si interfacciano tra di loro, oltre che tutte le specifiche tecniche che permetteranno di mettere a terra i requisiti definiti nella fase precedente.
- 3. Implementazione:** in questa fase avviene la vera e propria realizzazione del codice del sistema da parte del team di sviluppo. Ogni modulo viene sviluppato esattamente così come è stato definito nella fase precedente, rispettando rigorosamente le dettagliate specifiche tecniche. In questa fase difficilmente si possono accettare adattamenti o modifiche.
- 4. Verifica:** una volta che il sistema è stato realizzato, è possibile verificarne il funzionamento mediante dei test. Ogni modulo che compone il sistema viene sottoposto a diversi test che permettono di individuare e correggere errori, difetti e piccole incongruenze con quanto richiesto inizialmente. Questa fase è molto importante perché permette di garantire che il software sia effettivamente conforme con le aspettative e che eventuali difetti siano corretti tempestivamente. Modifiche eccessive non possono essere accolte in quanto necessiterebbero di una nuova analisi attenta e dettagliata e poi di una nuova fase di sviluppo, comportando un aumento dei costi e delle tempistiche, non accettabili in questa fase quasi finale del progetto.
- 5. Manutenzione:** a seguito del rilascio in produzione, l'ultima fase del modello a cascata, prevede l'aggiornamento e il miglioramento del sistema una volta che viene utilizzato dai clienti finali. Qualora si rilevassero dei bug o dei

malfunzionamenti, il cliente contatta il servizio clienti che mette al corrente gli sviluppatori che si occuperanno di gestire il problema e risolverlo. La manutenzione, quindi, prevede che da un lato si risolvano eventuali bug non emersi durante i test, ma anche l'implementazione di aggiornamenti, di miglioramenti di funzionalità e di piccoli adattamenti per rispondere a eventuali cambiamenti richiesti.

Durante l'esperienza di tirocinio presso Santander Consumer Bank sono emersi alcuni dei punti critici e dei limiti del modello Waterfall. Ai fini di questa tesi si presentano successivamente alcune delle criticità riscontrate prendendo come riferimento un progetto fittizio chiamato *Nuovo sistema di accesso all'area clienti* che, benché non mostri dati reali, rappresenta fedelmente la struttura e le tempistiche di un progetto simile a quello seguito in azienda, affiancando il Product Owner che lo stava gestendo.

Si vuole realizzare un nuovo sistema di accesso all'area riservata dei clienti della banca che includa:

- **login:** pagina in cui effettuare l'accesso con username e password,
- **registrazione:** pagina in cui i clienti con un contratto attivo possono registrarsi creando una nuova utenza,
- **funzionalità di cortesia:** alcune pagine che permettano di recuperare il numero contratto, la password e la username in modo autonomo o di aggiornare il numero di cellulare,
- **certificazione:** invio di OTP per certificare i recapiti del cliente e per validare gli accessi e le richieste effettuate attraverso le nuove funzionalità di cortesia.

La pianificazione delle attività del progetto è mostrata nella Tabella 2. Come si può notare le attività vengono pianificate in sequenza partendo dalle analisi iniziali e arrivando al go-live, evidenziando come solo il termine dell'attività precedente dia il via all'attività successiva, caratteristica tipica del modello Waterfall.

Una prima criticità che emerge riguarda il coinvolgimento degli utenti nelle diverse attività. Con utenti si intendono le persone che fanno parte delle direzioni impattate da una

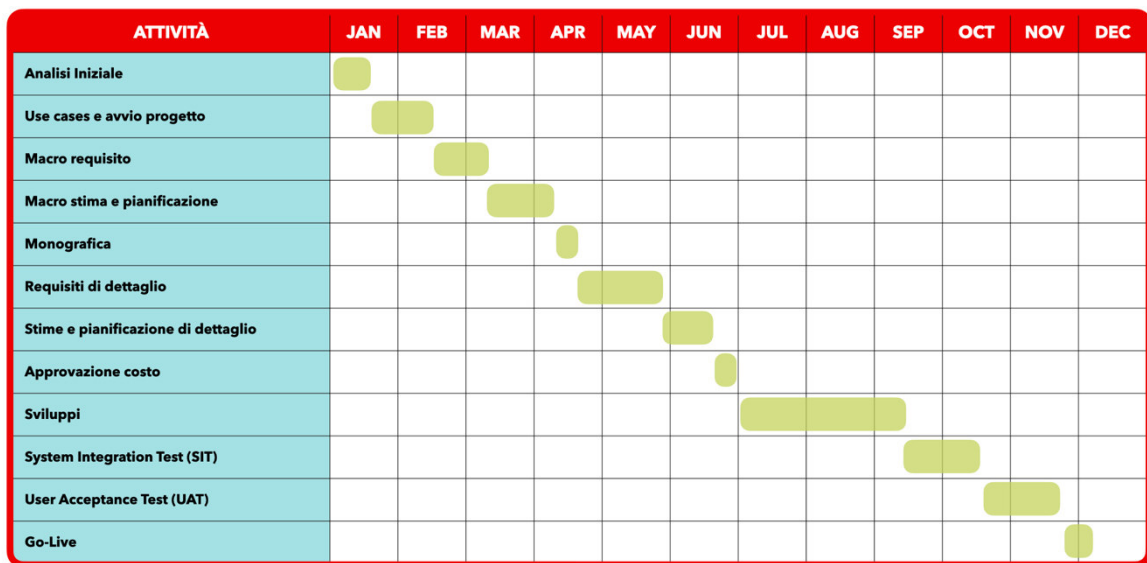


Tabella 2 - Esempio di GANTT del progetto fittizio Nuovo sistema di accesso all'area clienti

determinata progettualità e che quindi generalmente sono invitate a fornire durante la raccolta dei requisiti quelle che sono le richieste e le funzionalità che dovranno essere implementate dal team di sviluppo. Tra le attività elencate, gli utenti vengono coinvolti nelle fasi iniziali del progetto, in particolare nella definizione degli *use cases*, nella formulazione dei macro requisiti e successivamente, se il progetto viene effettivamente approvato, nella raccolta dei requisiti di dettaglio. Successivamente, fintanto che gli sviluppi non saranno terminati e i test tecnici da parte del team di sviluppo non saranno effettuati con esito positivo, gli utenti generalmente non vengono più coinvolti. È poi durante gli User Acceptance Test (UAT) che gli utenti saranno nuovamente chiamati in causa per testare le funzionalità realizzate durante la fase di sviluppo e verificare che queste rispettino i desiderata. Il tempo che intercorre tra la raccolta dei requisiti e la validazione della loro implementazione è generalmente di molti mesi. In particolare, nell'esempio presentato intercorrono cinque mesi, dalla fine di maggio alla fine di ottobre. Si tratta effettivamente di un lungo periodo al termine del quale le richieste iniziali potrebbero non essere più completamente in linea con le necessità reali e attuali. In più, bisogna considerare che si sta analizzando una situazione ideale, che non tiene conto di possibili ritardi che, con un approccio di questo tipo, si ripercuoterebbero inevitabilmente su tutte le attività successive ritardandone l'inizio. Come verrà presentato meglio nel sottocapitolo 3.3, durante i SIT sono emerse delle problematiche che non erano state prese in

considerazione durante le analisi tecniche che hanno causato la necessità di apportare delle modifiche a quanto era stato sviluppato. Alcune funzionalità non potevano essere testate perché mancavano alcune implementazioni cruciali per garantire il funzionamento del sistema di accesso. Questo ha causato un aumento della durata della fase dei System Integration Test (SIT) con conseguente ritardo nell'inizio degli UAT. Quindi il tempo che intercorre tra la definizione dei requisiti e l'inizio delle verifiche che forniscono un feedback concreto sulla coerenza degli sviluppi è veramente ampio e destinato a incrementare rispetto alla pianificazione. Questo ha causato principalmente due effetti indesiderati:

- **Disallineamento tra requisito e implementazione:** in diversi casi quanto è stato sviluppato non aderiva con quanto era stato richiesto dagli utenti all'inizio del progetto. Accorgersene negli UAT ha causato dei ritardi dovuti alla necessità di tornare a sviluppare mettendo mano a un prodotto già articolato e ricco di tutte le funzionalità richieste, con il rischio di introdurre errori e bug che non erano previsti.
- **Difficoltà nel ricordare i dettagli delle richieste iniziali:** è inevitabile che dopo così tanti mesi, durante i quali si svolgono attività diverse rispetto al tema del progetto, non sia così immediato ricordare ciò che era stato effettivamente richiesto nei minimi dettagli. In tal senso, esiste un documento che viene redatto a seguito della raccolta dei requisiti che contiene la descrizione dettagliata di ogni funzionalità. Tuttavia, si è potuto notare che non sempre il livello di dettaglio è sufficiente per evitare che alcune richieste risultino ambigue e quindi, lasciando spazio all'interpretazione, genera confusione.

È quindi interessante notare come questo primo limite del modello a cascata emerso durante l'esperienza sul campo conferma la necessità di coinvolgere maggiormente gli utenti durante tutte le fasi di lavoro di un progetto. Si pensi, per esempio, ai vantaggi che sarebbero derivati dall'anticipare i test su alcune funzionalità già durante lo sviluppo,

andando subito a chiarire possibili ambiguità e a segnalare tempestivamente tutti i possibili disallineamenti emersi già durante l'implementazione, senza che si palesino tutti alla fine, quando ormai la data di conclusione del progetto si avvicina e diventa molto importante non superarla. In più, qualora il disallineamento risultasse troppo grande si rischierebbe di non poter più applicare delle azioni correttive in quanto risulterebbero troppo impegnative in termini di tempi e di costi. In questi casi, a volte, si preferisce tralasciare la data di go-live con ancora dei punti aperti che saranno gestiti successivamente e potrebbero comunque essere gestiti come delle richieste di cambiamento con quindi un costo ulteriore.

Un altro aspetto limitante del modello Waterfall è emerso proprio al raggiungimento della data di go-live del progetto. Per comprendere al meglio quanto segue è opportuno introdurre un nuovo requisito fondamentale per il progetto esempio Nuovo sistema di accesso all'area clienti:

- **disponibilità Web:** i clienti dovranno poter accedere all'area riservata utilizzando il proprio browser e navigando tramite il sito istituzionale, sia da desktop, sia da mobile,
- **disponibilità App:** i clienti avranno la possibilità di accedere all'area riservata utilizzando l'applicazione mobile, sia da dispositivi iOS, sia da dispositivi Android.

Questi due requisiti aggiungono non poca complessità al progetto. In effetti, l'azienda dispone già di un sistema di accesso dedicato ai clienti che è utilizzabile sia da sito web, sia da applicazione nativa sui dispositivi mobili. L'intera architettura è predisposta per far sì che i sistemi sottostanti forniscano le informazioni necessarie indipendentemente da quello che è lo strumento che il cliente utilizza per interfacciarsi con la propria area riservata. Non trattandosi quindi della realizzazione di un nuovo prodotto, ma del miglioramento di uno già preesistente, diventa ancora più cruciale tener conto delle sfide che si potrebbero dover affrontare. In questo caso, la pianificazione però non fa una distinzione tra sito web e applicazione mobile, di conseguenza, tiene conto che alla data del go-live entrambi siano disponibili e pronti per il rilascio simultaneo in produzione. A

causa di una problematica che non ha permesso al team di sviluppo di utilizzare i software opportuni per lo sviluppo mobile che ne permettessero il rilascio negli ambienti predisposti ai test, gli User Acceptance Test delle nuove funzionalità sono stati effettuati solo sulla parte web. Potenzialmente, essendo i sistemi sottostanti gli stessi del mobile, questo non ha rappresentato in un primo momento un problema in quanto i test web avrebbero permesso comunque di verificare e validare tutte le funzionalità e in un secondo momento si sarebbe testata solamente l'applicazione mobile, potenzialmente in poco tempo, essendo semplicemente l'interfaccia verso i sistemi sottostanti già testati. Non solo, qualora si fosse raggiunta la data del go-live senza disporre ancora completamente dell'applicazione mobile, si sarebbe potuto procedere in un primo momento con la sola parte web e poi successivamente con la parte mobile. In realtà, dietro a questa situazione potenzialmente gestibile si cela una problematica ben più bloccante. Dal momento in cui non è stata considerata la possibilità che il rilascio della parte web sarebbe potuto avvenire in un momento diverso da quello della parte mobile, gli aggiornamenti ai sistemi sottostanti non sono stati progettati per gestire questo disallineamento. In altri termini, l'attuale applicazione mobile che si basa sui servizi sottostanti che sono stati oggetto di modifica durante il progetto avrebbe smesso di funzionare con il rilascio della

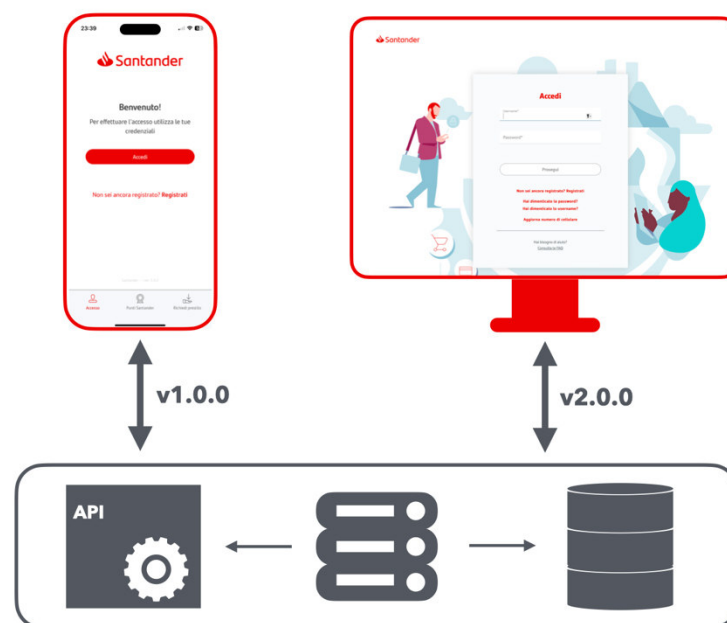


Figura 13 - Schema semplificato a scopo rappresentativo dell'architettura del sistema di accesso in grado di comunicare con diverse versioni di front-end.

parte web e dei nuovi servizi, in quanto essenziali per la nuova versione web ma non più adatti all'attuale versione mobile. In un'ottica di rilascio simultaneo degli aggiornamenti sia al sito web sia all'applicazione mobile ciò non avrebbe causato alcun disservizio, ma in questo caso, invece, avrebbe impedito l'accesso all'area riservata da applicazione mobile a tutti i clienti che non utilizzano il sito web. Insomma, una problematica non da poco che ha portato alla decisione di posticipare la data del go-live di ben due mesi, evidenziando come il modello a cascata risulti rigido e poco incline alla gestione di imprevisti con risposte rapide e adattamenti che potrebbero mitigare le conseguenze.

Un approccio Agile avrebbe consentito una maggiore flessibilità di fronte a un ostacolo come quello che si è presentato, ma è bene notare che per poter garantire una tale flessibilità non sarebbe sufficiente scomporre in sottogruppi di funzionalità il progetto, prevedendo molteplici rilasci già in partenza. Infatti, sarebbe necessario implementare una serie di accortezze che rendano i sistemi stessi più flessibili, consentendo, per esempio, di comunicare con diverse versioni dei servizi sottostanti, in base alla versione del software applicativo utilizzato dal cliente. Ciò comporta delle modifiche significative che impattano ogni livello dell'intero sistema e un approccio sostanzialmente diverso per il team di sviluppo. L'adozione di una nuova metodologia ha quindi delle ripercussioni anche in tutti i sistemi che necessitano di opportuni adattamenti per poter garantire che siano adatti a un nuovo modo di gestire gli sviluppi e i rilasci.

3.3 SVILUPPARE DURANTE I TEST

Terminata la fase di sviluppo di un software, prima di poter essere rilasciato in produzione, è necessario che passi attraverso due fasi dedicate alla validazione delle implementazioni: i System Integration Test e gli User Acceptance Test. In generale, i System Integration Test (SIT) sono una fase molto importante nei progetti IT, soprattutto quando si ha a che fare con sistemi complessi come quelli di una banca. L'obiettivo dei SIT è quello di verificare che i vari moduli software e hardware che compongono un sistema funzionino correttamente una volta collegati insieme. Potenzialmente, anche se i singoli elementi testati in modo isolato dagli altri non presentano malfunzionamenti o errori, quando si devono interfacciare con l'intero sistema possono insorgere delle problematiche. Quindi è opportuno testare che le diverse componenti si integrino correttamente per garantire che l'intero sistema operi come richiesto. Durante questi test si verifica il comportamento dell'intero sistema, controllando che la comunicazione tra i vari moduli avvenga correttamente e che le interazioni tra hardware e software siano funzionanti. Altro aspetto molto importante da controllare sta nel flusso di dati attraverso le varie componenti, il quale deve avvenire regolarmente evitando che vi siano dei disallineamenti tra dati in input e in output di ogni elemento. Eventuali errori che emergono durante i SIT dovranno essere corretti per garantire che l'interazione tra tutti i moduli avvenga come previsto. Generalmente, viene predisposto un ambiente apposito all'interno del quale il team di sviluppo rilascia le implementazioni dei diversi moduli. Questo ambiente prende il nome di *ambiente di DEV* (Development), o di sviluppo. È importante che replichi il più fedelmente possibile l'ambiente di produzione sia nella sua struttura, sia nei dati presenti. In questo modo, eventuali problematiche potranno emergere prima di raggiungere l'ambiente di produzione dove causerebbero dei disservizi ai clienti finali. Prima di procedere con i SIT è importante che tutti i team di sviluppo coinvolti rilascino i moduli implementati, procedendo quindi a una serie di test che validino l'intero flusso dei dati attraverso il sistema. È bene monitorare gli input e gli output di ogni componente man mano che il sistema viene verificato, in modo tale da accorgersi immediatamente se vi sono delle incongruenze o delle problematiche di interfaccia. Per esempio, è importante verificare che i servizi sottostanti ricevano dallo strato di front-end tutti i dati che

si aspettano in ingresso, oppure che i servizi intermedi confezionino delle richieste ai servizi di back-end con le opportune interfacce definite tra i moduli. I SIT, quindi, sono una fase fondamentale in un progetto IT in quanto permettono di ridurre la comparsa di errori e di malfunzionamenti in produzione a causa di una non completa integrazione e consentono di individuare problematiche prima che il sistema venga validato dagli utenti durante gli User Acceptance Test.

La fase successiva a quella dei SIT è quella degli User Acceptance Test (UAT). Si tratta dell'ultima fase di verifica e di validazione di un progetto software prima che questo venga rilasciato in produzione. In questa fase, vengono chiamati nuovamente in causa gli utenti interni delle diverse direzioni impattate dal progetto per testare che le funzionalità implementate siano in linea con le richieste formulate durante la raccolta dei requisiti di dettaglio. Anche in questa fase, è importante disporre di un ambiente dedicato che replichi il più fedelmente possibile quello di produzione. Questo ambiente prende il nome di *ambiente di PRE* (Pre-production), o di collaudo. È bene notare che a differenza dei SIT, dove le problematiche che emergono sono prettamente di carattere tecnico, durante gli UAT si validano le funzionalità del sistema da un punto di vista dell'utente finale andando quindi a controllare che il sistema soddisfi quelli che sono i requisiti aziendali stabiliti, funzioni correttamente con dati che replicano fedelmente la realtà e che quindi ogni funzionalità sia pronta per essere utilizzata anche dai clienti finali senza incorrere in spiacevoli errori. In tal senso, è molto importante che durante questa fase non emergano errori tecnici in quanto non permetterebbero di comprendere a pieno le singole funzionalità perché i vari use cases testati verrebbero interrotti nel loro normale flusso mostrando dei comportamenti non in linea con quanto richiesto. Ciò non solo andrebbe a evidenziare notevoli problematiche ancora esistenti che impediscono il rilascio in produzione del software, ma non consentirebbe di concentrarsi sul reale obiettivo degli UAT condotti dagli utenti finali, cioè la validazione dell'esperienza e della usabilità del sistema. Rappresentando il momento finale in cui viene deciso se i nuovi moduli sviluppati sono effettivamente pronti per il rilascio in produzione, è una fase cruciale perché l'esito potrebbe influenzare il go-live. Questa fase è delicata anche perché il coinvolgimento

degli utenti interni non è di semplice gestione soprattutto in un contesto articolato come quello presentato per Santander Consumer Bank Italia. Progetti più articolati che impattano diverse aree dell'azienda implicano il coinvolgimento negli UAT di molte persone, ognuna impegnata già in altre attività e interessata a verificare nel dettaglio la parte di competenza. Proprio su questo ultimo aspetto, è importante notare come in questa fase è molto probabile che emergano degli scenari di utilizzo che non erano stati considerati durante le fasi di analisi e di sviluppo. Da un lato, questo rappresenta un'opportunità perché si evita di andare in produzione e di accorgersi solo successivamente di possibili situazioni spiacevoli non gestite, dall'altro lato, questo può introdurre dei ritardi che rischiano di minare il rispetto della pianificazione e il traguardo della data del go-live. In Tabella 3 è possibile vedere le principali caratteristiche dei SIT e degli UAT messe a confronto.

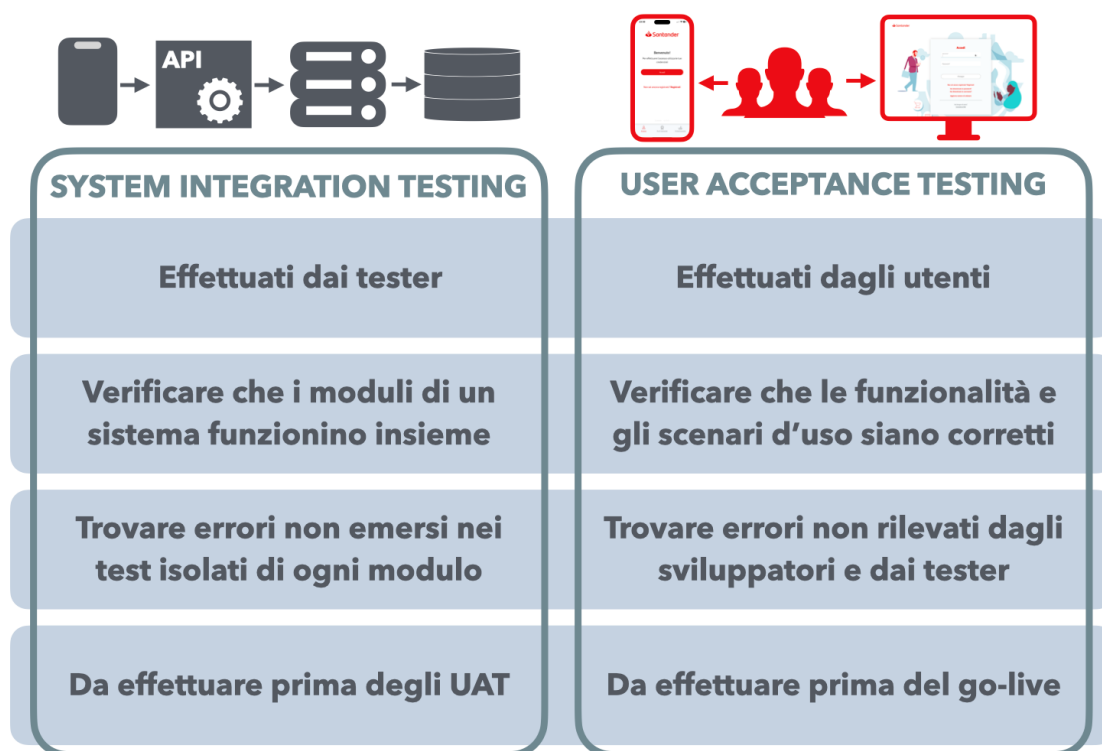


Tabella 3 - System Integration Testing e User Acceptance Testing: un confronto.

Dopo circa due settimane dall'inizio dell'esperienza di tirocinio e di tesi in azienda presso Santander Consumer Bank Italia, il progetto che continueremo a chiamare in modo fittizio Nuovo sistema di accesso all'area clienti stava entrando nella fase di System

Integration Test (SIT). Questa fase è molto importante per l'azienda in quanto l'architettura è molto complessa e articolata come per la maggior parte delle aziende nel settore bancario. Vi sono molteplici elementi come, per esempio, i sistemi legacy per la gestione delle operazioni di *core banking*, le piattaforme interne per gli utenti delle diverse direzioni come i portali web, le piattaforme esterne per i clienti finali come le applicazioni mobile e i siti web, i database centralizzati e distribuiti per l'archiviazione delle informazioni sensibili e i sistemi di sicurezza e compliance per garantire l'aderenza alle normative vigenti. A complicare questo contesto già sfidante, come verrà presentato nel sottocapitolo 3.5, l'azienda si serve di diversi fornitori esterni per gli sviluppi dedicati alle diverse componenti del sistema. Questo implica che venga posta una particolare attenzione al modo in cui l'intero sistema deve funzionare, integrando tutte le componenti attentamente. In tal senso, le analisi tecniche che vengono svolte prima degli sviluppi prevedono una cooperazione da parte dei diversi fornitori che devono impegnarsi nel definire al meglio le interfacce tra i diversi moduli del proprio strato, prevedendo a priori che questi dovranno comunicare puntualmente gli uni con gli altri. Al termine dell'analisi tecnica, tra i vari documenti prodotti, assumono notevole importanza i cosiddetti *sequence*. Si tratta di diagrammi che rappresentano il flusso di interazione tra i diversi moduli e componenti del sistema. Su di essi si segna puntualmente l'ordine cronologico e logico con cui i vari componenti scambiano informazioni per far sì che una determinata funzionalità sia eseguita. Tuttavia, vi sono alcune criticità che sono emerse con maggiore frequenza durante i SIT e che quindi necessitano di una maggiore attenzione, onde evitare di introdurre dei ritardi nella pianificazione. Di seguito sono presentate alcune delle criticità più ricorrenti.

- **Servizi/moduli non raggiungibili:** durante le analisi tecniche si definiscono quelli che saranno i nomi dei vari moduli sviluppati dai singoli fornitori e dai *sequence* è possibile visualizzare la catena delle chiamate per una determinata funzionalità. In più occasioni, diversi moduli non erano in grado di comunicare con gli altri perché non venivano chiamati con il nome previsto, con la conseguente necessità di impiegare del tempo per verificare che effettivamente tutti i nomi utilizzati

fossero corretti. Benché i SIT permettano proprio di rilevare anche queste incongruenze, è sicuramente opportuno prestare una maggiore attenzione su questa tematica nelle fasi precedenti, così da evitare un eccessivo spreco di tempo durante questa fase.

- **Mancanza di dati in input/output:** un'altra criticità emersa con una certa frequenza riguarda il flusso dei dati tra i diversi strati del sistema. In diverse occasioni, i moduli integrati non funzionavano correttamente perché i servizi all'interno della catena andavano in errore non disponendo di tutti i dati che sarebbero stati necessari per effettuare l'elaborazione. Anche in questo caso queste informazioni sono disponibili nei sequence, ma anche nei file YAML utilizzati per configurare i vari componenti del sistema. Questi file contengono i dettagli delle configurazioni dei servizi come i parametri richiesti per l'elaborazione, le dipendenze tra i moduli, le variabili ambientali e i dati necessari per il corretto funzionamento dei processi. Un'errata configurazione causa inevitabilmente degli errori che minano l'intero sistema. Tuttavia, è comprensibile che durante i SIT emergano delle criticità di questo tipo e che vengano di conseguenza sistemate. Va però notato che in più occasioni il tempo necessario per identificare che la problematica fosse legata all'assenza di dati in input è stato significativo. Questo è dovuto al fatto che non sempre i messaggi di errore utilizzati erano chiari e descrittivi. Se da un lato è buona norma non mostrare al cliente finale errori tecnici descrittivi, dall'altro lato è bene che, soprattutto durante una fase come è quella dei SIT, eventuali errori permettano di capire rapidamente la loro causa. Piuttosto che restituire un errore generico dovuto all'assenza di dati in ingresso che hanno causato un arresto anomalo dell'elaborazione, sarebbe opportuna una validazione dei dati e l'implementazione di meccanismi di *fallback* in grado di segnalare chiaramente quali sono i dati mancanti per garantire una risoluzione più rapida.

- **Formato dei dati:** oltre all'assenza dei dati in ingresso a un determinato modulo, alcuni errori emersi con maggiore frequenza riguarda il formato dei dati. In questo caso, i moduli che comunicano con un modulo o un servizio successivo non rispettano il formato previsto causando quindi degli errori di elaborazione. In questo caso anche risulta molto importante il modo in cui vengono gestiti questi errori in modo tale da fornire una segnalazione chiara e immediata che ne favorisca la risoluzione tempestiva. In più, sarebbe bene che in fase di analisi tecnica venissero definiti degli standard da adottare per tutti i moduli attraverso tutti i sistemi. Sicuramente far comunicare diverse componenti nuove con altre componenti già preesistenti che utilizzano tecnologie meno recenti potrebbe essere complicato e quindi, a maggior ragione, è bene che si ponga un'attenzione particolare sin da subito a questo aspetto. Facciamo un esempio molto banale sul progetto fittizio preso in considerazione. Ci troviamo nella pagina dedicata alla login dei clienti e il front-end chiama un servizio sottostante passando in input lo username e la password inseriti nei campi appositi. Lo step successivo prevede l'invio di un codice OTP via SMS al numero di cellulare presente nell'anagrafica del cliente se l'utenza è corretta. Il modulo che si occupa di recuperare il numero di cellulare dall'anagrafica restituisce un numero di cellulare di tipo stringa che contiene uno spazio tra le prime tre cifre del numero e le restanti. Il modulo invece che si occupa dell'invio dell'SMS si aspetta in ingresso un numero intero come numero di cellulare. Va da sé che una volta integrati, i due moduli non comunicheranno correttamente generando un errore a causa del differente formato con cui il dato viene ricevuto in input rispetto a quello atteso. È quindi necessario concordare la conversione del dato in uno dei due moduli comportando quindi uno sviluppo, che richiede del tempo aggiuntivo. Qualora questo si verificasse con una frequenza alta potrebbe allungare notevolmente il tempo effettivo dedicato alla fase dei SIT, generando un ritardo per le fasi successive.

È possibile notare quindi come la fase dedicata ai System Integration Test sia molto importante perché permette di rilevare tutte quelle problematiche che impediscono il

corretto funzionamento dell'intero sistema. Va considerato però che essendo una fase a valle dell'intero lavoro di sviluppo, spesso la quantità di errori che si verificano tendono ad accumularsi e benché siano potenzialmente di rapida investigazione e risoluzione, in grande quantità possono causare dei ritardi notevoli. In effetti, rispetto alla pianificazione originale, nel caso del progetto esempio preso in considerazione, a causa dei molteplici errori emersi durante i SIT, l'inizio degli UAT è stato posticipato di ben due settimane. Sicuramente un'analisi tecnica più attenta e precisa potrebbe garantire una sostanziale riduzione delle problematiche a valle degli sviluppi, ma un modo per evitare che questi compaiano tutti insieme va proprio ricercato nella metodologia Agile. Qui trova la sua applicazione il concetto di effettuare dei test costantemente durante gli sviluppi così da ottenere nell'immediato feedback molto preziosi. Invece di sviluppare durante i test, si renderebbe molto utile testare durante gli sviluppi. Non si tratta in questo caso di coinvolgere gli utenti in quanto si sta parlando di problemi tecnici e non legati ad aspetti funzionali, ma l'idea che in una situazione in cui i fornitori sono diversi e si occupano di componenti diverse dell'intero sistema, sarebbe opportuna una costante verifica delle implementazioni man mano che sono disponibili. Riprendendo l'esempio dell'invio del codice OTP tramite SMS, il fornitore che si occupa del modulo responsabile dell'invio, una volta terminato lo sviluppo invece di proseguire con l'implementazione di altri moduli per poi testare tutto alla conclusione complessiva degli sviluppi, potrebbe contattare il secondo fornitore, responsabile del modulo che recupera il cellulare dall'anagrafica del cliente per effettuare un test congiunto e verificare che effettivamente i due moduli che sono dipendenti l'uno dall'altro comunichino efficacemente e non presentino dei disallineamenti. In questo modo, già in corso d'opera sarebbe possibile risolvere eventuali errori di integrazione mitigando la possibilità che si sommino a ulteriori errori che potrebbero comparire nella fase dei SIT. Certamente un approccio del genere implica un cambiamento nel modo in cui i fornitori affrontano la fase degli sviluppi e l'azienda in tal senso è responsabile di promuovere e di favorire la comunicazione tra di loro. In quest'ottica, Santander Consumer Bank Italia ha dotato tutti i fornitori di laptop corporativi che facilitano la comunicazione tra l'azienda e i fornitori, ma anche tra di essi. Si tratta di una mossa vincente perché in questo modo è possibile utilizzare degli applicativi

di messaggistica e di comunicazione istantanea, favorendoli all'invio di e-mail che spesso richiedono tempistiche maggiori sia di lettura, sia di risposta. È importante quindi che la comunicazione tra i fornitori coinvolti nello sviluppo di moduli appartenenti allo stesso sistema avvenga in modo costante durante tutta la fase degli sviluppi, arrivando così alla fase dei SIT con un sistema che risulta già più stabile e solido, senza rischiare di introdurre ritardi che possono essere molto rischiosi in un modello rigido come quello a cascata.

Terminati i SIT, con un ritardo di circa due settimane, il progetto è passato nella fase di User Acceptance Test (UAT). In questo caso gli utenti delle direzioni impattate dal progetto si sincerano che tutte le funzionalità richieste siano state implementate correttamente. Un primo aspetto che è interessante evidenziare riguarda, anche in questo caso, alcune segnalazioni ricorrenti che sono emerse durante gli UAT, così come presentate di seguito.

- **Codifica dei messaggi di errore:** tutti i messaggi di errore che vengono mostrati a front-end sono stati evidenziati durante la raccolta dei requisiti e mappati durante l'analisi tecnica. In questo modo, ogni fornitore responsabile dei diversi moduli del sistema sarebbe stato in grado di comunicare il codice di errore giusto a fronte di una determinata situazione anomala gestita. Per quanto la raccolta dei requisiti di dettaglio abbia permesso di evidenziare molte delle casistiche di errori da gestire, è stato solo durante i test con gli utenti che, mettendo effettivamente mano al sistema e creando delle casistiche di errore non mappate, è nata la necessità di introdurre nuovi codici. Accorgersene in questa fase è sicuramente importante perché fa sì che il software che verrà rilasciato in produzione sarà in grado di gestire tutte le situazioni di errore comunicando il messaggio opportuno al cliente finale, ma effettivamente rischia di avere degli impatti notevoli sulle tempistiche dell'ultima fase precedente al go-live. Se si considera, infatti, che molte volte questa fase ha subito già dei ritardi accumulati delle fasi precedenti, sarebbe opportuno che si svolgesse il più rapidamente possibile, in un flusso ininterrotto di test. In più, la non gestione a valle di certe casistiche di errore perché

non sono emerse a monte, potrebbe anche significare la necessità di implementare nuove logiche che comportano delle rivisitazioni sostanziali di quanto è stato sviluppato, rischiando addirittura di non pesare solo sul tempo complessivo, ma anche sul costo del progetto. Siccome l'approvazione dei costi del progetto avviene a monte, non sono ammessi ulteriori spese, soprattutto in una fase conclusiva come è quella degli UAT, con il rischio che si debba ricorrere a delle soluzioni che non sempre sono le più adeguate.

- **Copy delle comunicazioni al cliente:** tutto il testo presente nei punti di contatto verso il cliente deve essere approvato da Compliance. Questo fa sì che vi sia conformità nelle comunicazioni e che vengano rispettate le normative interne ed esterne all'azienda, interfacciandosi con i clienti in modo uniforme, chiaro e corretto. Gli utenti effettuando i test si possono rendere conto della necessità di correggere o di cambiare il testo, sia perché non in linea con altre comunicazioni, sia perché potrebbero essere cambiate le normative dalla formalizzazione dei requisiti all'effettiva sessione di validazione durante gli UAT. In più occasioni, è stato necessario modificare il testo presente nelle finestre modali e nei pop-up mostrati al cliente durante la sua esperienza di interazione con il sistema di accesso. Effettuando i test e riproducendo l'esperienza di utilizzo di un potenziale cliente, gli utenti hanno riscontrato che non sempre ciò che veniva comunicato forniva indicazioni utili che aiutassero il cliente nelle sue azioni. Potrebbe sembrare una banalità, ma una modifica di questo tipo costa molto in termini di tempo. Come detto precedentemente, gli UAT avvengono in un ambiente dedicato, chiamato ambiente di PRE o di collaudo. Fare una modifica a un modulo del sistema vuol dire doverla rilasciare causando dei momentanei disservizi che potrebbero avere delle ripercussioni qualora si stiano effettuando altri test nello stesso ambiente. In tal senso, vi è una regola che impone di effettuare i rilasci in ambiente di collaudo prima delle 9:00 oppure dopo le 18:00, in modo tale da contenere possibili disservizi. Apportare una modifica non permette quindi di avere un riscontro immediato, ma implica che si debbano effettuare nuovamente gli stessi test almeno

il giorno successivo, quando sarà rilasciata e disponibile nell'ambiente. È sicuramente vero che le segnalazioni di questo tipo che emergono in una sessione di UAT sono molteplici e quindi potranno essere nuovamente testate il giorno successivo in blocco, ma allo stesso tempo è proprio questa molteplicità che le rende potenzialmente pericolose. Si pensi cosa potrebbe accadere se non emergessero tutte durante la stessa sessione, ma in giornate successive, con un effetto a cascata che farebbe aumentare vertiginosamente la durata della fase degli UAT.

Un'altra criticità emersa durante gli UAT riguarda una determinata funzionalità che non era stata implementata correttamente, a causa di una mancanza nella descrizione del requisito. Per cercare di capire meglio di cosa si è trattato e di quelle che sono state le conseguenze, di seguito è presentata una situazione fittizia, ma simile a quella reale. È fondamentale che i documenti di identità dei clienti della banca siano sempre aggiornati. I clienti sono quindi invitati ad aggiornare il proprio documento attraverso l'area riservata. In questo esempio, supponiamo di avere un requisito di questo tipo:

- in caso di documento scaduto, in fase di accesso, sarà chiesto al cliente di aggiornarlo prima di proseguire con la normale procedura di login,
- bisognerà mostrare al cliente un modulo che richieda l'inserimento dei dati seguenti:
 - Tipologia del documento (menu a tendina)
 - Numero del documento
 - Data emissione
 - Data scadenza
- inoltre, sarà richiesto di allegare copia del fronte e copia del retro del documento di identità.

Questo requisito presenta un'importante mancanza che non è stata notata né dagli utenti durante l'approvazione, né dai fornitori durante le analisi tecniche. È stato successivamente, durante gli UAT, che ci si è resi conto che i servizi sottostanti aggiornavano parzialmente il dato del documento di identità presente in anagrafica, in quanto vi sono

due ulteriori campi da valorizzare, il cui valore non era ricevuto in input. Si tratta della provincia di rilascio e del comune di rilascio. L'errore sta nel fatto che l'assenza di questi dati lascia invariati i campi in anagrafica che quindi non sono in linea con il nuovo documento salvato, ma rimangono quelli del documento precedente. Con questa modalità di lavoro che prevede la validazione delle funzionalità solo al termine degli sviluppi e in una fase conclusiva questo rappresenta un problema non da poco. A peggiorare la situazione si tratta di due campi che non prevedono un formato testuale libero, come potrebbe essere un campo per l'inserimento di un nome, ma vi è la necessità di mostrare due liste dinamiche di tutte le province italiane e di tutti i comuni italiani così come accettati dal sistema in cui è memorizzata l'anagrafica dei clienti. In più, come accennato nel sottocapitolo 3.2, il progetto prevede che i clienti possano utilizzare il sistema di accesso sia da sito web, sia da applicazione mobile iOS e Android, rendendo quindi necessaria l'integrazione dei due nuovi campi in tre front-end diversi con le relative peculiarità. Insomma, tale mancanza ha comportato in questa fase del progetto degli sviluppi ulteriori che hanno richiesto del tempo in più e anche un costo aggiuntivo. Si tratta di una situazione molto delicata che ha avuto delle conseguenze maggiori rispetto ad altre segnalazioni emerse, ma ciò permette di comprendere al meglio quanto sia importante disporre di metodologie e di strumenti che permettano di accogliere con maggiore flessibilità possibili modifiche, siano esse causate da un requisito incompleto o da cambiamenti in itinere. Come detto precedentemente, la chiarezza e la trasparenza dei requisiti è fondamentale per un progetto IT ed è bene che tutte le persone impattate siano coinvolte maggiormente in ogni fase. Si pensi, per esempio, a come gli eventi in Scrum sono pensati proprio per fronteggiare situazioni di questo tipo. Se gli utenti fossero stati coinvolti già durante gli sviluppi, potenzialmente ancora prima che i fornitori avessero iniziato a sviluppare questa funzionalità in uno Sprint, con molta probabilità ci si sarebbe accorti tempestivamente della mancanza, analizzando il requisito nel Product Backlog e dettagliandolo per l'inizio dello Sprint. Altrimenti, durante la Sprint Review sarebbe stato possibile segnalare l'errore e prevedere tramite il raffinamento del backlog di assorbire l'aggiunta necessaria in corso d'opera, senza avere degli impatti significativi né sui tempi né sui costi finali del progetto. Ovviamente, è opportuno ricordare che l'adozione di un

framework come Scrum non è sufficiente per godere dei possibili vantaggi, ma è necessario che l'intero approccio aziendale cambi nella mentalità e nel modo in cui si organizza il lavoro, anche nei confronti e da parte dei fornitori.

3.4 IL RUOLO DEL PRODUCT OWNER

Entrare a far parte del team dei Product Owner (PO) di Santander Consumer Bank Italia durante l'esperienza di tirocinio e di tesi in azienda ha permesso di conoscere più da vicino l'importanza di una figura come quella del PO per quanto riguarda i progetti IT. Benché non vengano applicate delle metodologie in linea con quelle presentate per i framework Scrum e SAFe, il ruolo del Product Owner e le caratteristiche affrontate nel secondo capitolo trovano la loro applicazione e importanza anche in un modello più rigido come è quello a cascata adottato per le progettualità dell'azienda. In effetti, non va dimenticato che molti dei principi presentati sono prima di tutto dei modi di pensare e di affrontare le sfide quotidiane di questo lavoro e quindi è più che normale ritrovarli anche in un contesto dove una vera e propria implementazione di una metodologia Agile e lean non è presente. In più, è curioso notare come l'attuale organizzazione sia una diretta conseguenza delle esperienze che sono state fatte negli anni, delle azioni intraprese a seguito di esperienze più o meno di successo. Questo avvalorava ancora una volta l'importanza di interrogarsi sulle azioni passate per migliorare quelle future e restituisce l'immagine di una realtà che è consapevole di poter migliorare grazie alla collaborazione di tutti e a una leadership che invita le persone ad agire in autonomia e in modo proattivo. In questo contesto, i Product Owner sono chiamati a seguire le diverse progettualità IT fungendo da coordinatori tra gli utenti delle diverse direzioni coinvolte responsabili dei requisiti, tra i fornitori esterni responsabili degli sviluppi e tra tutti gli stakeholders. In ogni fase della pianificazione vista nel sottocapitolo 3.2, il PO ha un ruolo ben definito e delle responsabilità specifiche, così come presentato di seguito.

- **Analisi iniziale:** durante questa fase si analizza quello che è lo stato attuale (chiamato *as-is*) di un determinato processo, di un sistema o di ciò che in qualche modo è già presente rispetto al tema principale del progetto che sta per iniziare. In questa fase il PO è invitato a raccogliere tutte le informazioni necessarie e propedeutiche interfacciandosi di volta in volta con chi le detiene. Potrebbe essere necessario, per esempio, coinvolgere il team di architetture per conoscere quali servizi sono già stati implementati in passato per un determinato

applicativo, oppure, il team di infrastrutture per conoscere quali sono i server attualmente in uso. Il PO deve quindi assicurarsi e garantire che tutte le persone delle diverse aree dell'azienda dialoghino tra di loro per la buona riuscita dell'analisi iniziale.

- **Use cases e avvio progetto:** in questa fase vengono coinvolti principalmente gli utenti delle diverse direzioni che saranno impattate dal progetto. Si vanno a discutere i vari scenari di utilizzo che gli utenti hanno ipotizzato sulla base delle nuove richieste. Si tratta di una fase molto importante per il PO in quanto, per la prima volta, entra in contatto direttamente con le richieste degli utenti. È fondamentale che comprenda al meglio cosa si aspettano gli utenti dal progetto e quale sia il valore aggiunto. Si tratta dell'inizio di un lungo percorso che il PO dovrà fare con gli utenti durante il quale si troverà a dover prendere delle decisioni che dovranno essere consapevoli e informate. Per far sì che ciò sia possibile, è bene che il PO capisca attentamente le motivazioni che spingono gli utenti a proporre un determinato progetto, prima ancora di addentrarsi nelle singole funzionalità. L'obiettivo principe del PO è quello di riuscire a massimizzare il valore del prodotto per l'utente e se non si mettesse opportunamente nei suoi panni sarebbe molto difficile raggiungerlo. Capire il reale valore e le reali esigenze consente al PO di muoversi con maggiore autonomia durante tutto il progetto ottenendo l'adeguata fiducia degli utenti che si sentono rappresentati e sostenuti nelle scelte che vengono prese. È bene quindi che sin da questa fase iniziale il PO inizi a creare un rapporto di qualità con gli utenti.
- **Macro requisito:** in continuità con la fase precedente, la raccolta dei macro-requisiti vede coinvolti gli utenti con l'aggiunta dei fornitori. Il PO che è già entrato in contatto con le richieste degli utenti deve assicurarsi che queste vengano comprese al meglio dal team di sviluppo, in questo caso formato dai diversi fornitori esterni, il quale dovrà produrre una stima dei costi del loro lavoro a un livello più generale. È cruciale che in questa fase dove la stima che verrà prodotta al

termine sarà grossolana e non specifica nei dettagli, si comunichino chiaramente gli aspetti essenziali e la dimensione di quanto si richiede che venga realizzato durante il progetto. In un certo senso il PO è chiamato ad assicurarsi che il prodotto che ha compreso nelle sue caratteristiche e che dovrà valorizzare durante l'intero progetto venga percepito al meglio anche dai fornitori, in quanto i primi responsabili di realizzarlo concretamente.

- **Macro-stima e pianificazione:** in questa fase il PO è tenuto a rispondere a eventuali domande di chiarimento da parte dei fornitori e, se necessario, coinvolgere nuovamente gli utenti per chiarire quei punti che effettivamente non sono sufficientemente trasparenti. Una risposta tempestiva permette di raggiungere in tempo la data definitiva di consegna della macro-stima e della pianificazione da parte dei fornitori. Questi due documenti sono fondamentali per passare alla fase successiva e quindi il PO deve assicurarsi che non si allunghino i tempi.
- **Monografica:** in questa fase il Project Manager che ha preso in carico la gestione del progetto insieme agli utenti della direzione interessata a promuoverlo preparano una presentazione PowerPoint che descriva al meglio la natura del progetto, le motivazioni e i benefici, nonché i costi e le tempistiche. Viene organizzata una riunione apposita a cui generalmente partecipano l'Amministratore Delegato, i direttori delle direzioni direttamente interessate ed eventualmente anche di quelle impattate, il direttore di IT&Operations e, spesso, i loro diretti sottoposti. A questa riunione prende parte anche il PO che generalmente non è chiamato a intervenire, ma può essere interpellato per dare chiarimenti sulle stime e sulla pianificazione che sono state consegnate dai fornitori. Al termine della riunione, solo se il progetto riceverà l'approvazione proseguirà nelle fasi successive.
- **Requisiti di dettaglio:** in questa fase, si raccolgono quelli che sono i requisiti con un livello di dettaglio più specifico per ogni richiesta che è stata fatta. Gli utenti

sono chiamati a fornire tutte le informazioni necessarie nello specifico e il PO deve accertarsi che venga posta l'opportuna attenzione e che le descrizioni risultino chiare. I fornitori anche sono chiamati in causa ed è bene che recepiscano al meglio ogni minima particolarità in modo tale che le successive analisi tecniche avvengano senza lasciare nulla al caso. Il PO si assicura quindi che da un lato gli utenti esponano con chiarezza i requisiti dettagliati e dall'altro che i fornitori li recepiscano in modo chiaro e trasparente. L'obiettivo è che al termine di questa fase tutte le persone coinvolte abbiano una conoscenza sufficiente sia sul prodotto, sia sulle peculiarità che lo caratterizzano e che vanno prese in considerazione per l'effettivo successo delle fasi successive.

- **Stime e pianificazioni di dettaglio:** a questo punto i fornitori dettagliano la stima più ampia effettuata nelle fasi precedenti, andando a indicare tempi e costi per le differenti implementazioni necessarie per i requisiti di dettaglio. In questa fase, il PO deve costantemente monitorare che i fornitori siano in grado di fornire delle stime che siano coerenti e sensate rispetto alla richiesta. Una maggiore chiarezza iniziale e un costante dialogo fanno sì che al termine di questa fase le stime non necessitino di ulteriori modifiche che potrebbero causare delle attese prolungate che impattano l'inizio degli sviluppi. È importante notare che la stima di dettaglio non potrà mai superare gli importi previsti nella macro-stima. In tal senso, il PO deve collaborare anche con gli altri colleghi per comprendere se l'*effort* previsto è da considerarsi ragionevole e andare a rilevare il prima possibile l'insorgere di impedimenti dovuti a requisiti che, una volta dettagliati, risultano molto più sfidanti e onerosi.
- **Approvazione costo:** in questa fase il team preposto all'approvazione delle spese valida o meno la stima di dettaglio. L'approvazione darà il via agli sviluppi.
- **Sviluppi:** in questa fase i fornitori coinvolti si occuperanno di effettuare le analisi tecniche, di proporre e di implementare le soluzioni migliori per la realizzazione

delle funzionalità richieste. In questa fase, è opportuno che il PO si sinceri sugli avanzamenti e funga da ponte costante con gli utenti qualora emergessero delle incertezze sulle richieste. Sarebbe opportuno che i diversi fornitori in questa fase comunicassero costantemente tra di loro per scovare il prima possibile possibili problematiche di integrazioni dei diversi moduli che compongono il prodotto finale. Ciò non accade spesso come già evidenziato nel sottocapitolo precedente, ma sarebbe buona norma che il PO promuovesse la comunicazione costante. Settimanalmente si tengono delle riunioni per verificare lo Stato di Avanzamento dei Lavori (SAL) durante le quali i fornitori possono far emergere se vi sono delle criticità o degli ostacoli negli sviluppi in modo tale che si possa agire di conseguenza. In queste occasioni emergono anche eventuali ritardi, permettendo di agire in anticipo e cercare di mitigare ogni possibile conseguenza sull'intera programmazione.

- **System Integration Test (SIT):** in questa fase il PO partecipa insieme ai fornitori alle sessioni di test. Facendo riferimento a un *test book* dedicato che contiene l'elenco dei test da effettuare per validare il corretto funzionamento di tutte le componenti sviluppate nei vari strati dell'architettura del sistema, si correggono tutte quelle problematiche tecniche che non sono ancora state testate prima di questa fase, in quanto, durante gli sviluppi, i test effettuati dai fornitori sono per lo più unitari e legati alle singole componenti isolate. Per garantire che i test si svolgano con la necessaria attenzione il PO partecipa attivamente in ogni sessione ed è responsabile di comunicare con il team di infrastruttura e di architettura ogni qual volta se ne presenti la necessità. In effetti, è molto probabile che emergano delle esigenze prettamente tecniche la cui responsabilità è dei team tecnici interni. Gli interventi richiesti a volte necessitano di un lasso di tempo sostanzioso causando dei possibili blocchi delle sessioni. È quindi molto importante che il PO si muova d'anticipo e proattivamente per far sì che le segnalazioni bloccanti vengano smarcate il prima possibile, pena il non rispetto della pianificazione e la riduzione del tempo destinato ai test con gli utenti che sono molto

importanti per garantire la vera e propria aggiunta di valore al prodotto. Questa fase è molto importante per la banca perché si può avere sin da subito un riscontro sulla qualità degli sviluppi effettuati dai fornitori esterni. In tal senso, durante l'esperienza di tirocinio, è stato proposto di aggiungere alcune statistiche derivanti dai dati inseriti quotidianamente all'interno dei test book dei SIT. In questo modo è stato possibile avere un riscontro a colpo d'occhio sia sull'andamento dei test, sia sulla bontà del lavoro svolto da ogni fornitore durante la fase precedente.

- **User Acceptance Test (UAT):** in questa fase generalmente il team di Test Management si occupa di accompagnare gli utenti durante i test funzionali. Anche in questo caso si fa riferimento a un test book dedicato. In tal senso, è molto importante che il PO comunichi chiaramente al tester quali sono tutte le funzionalità che sono state richieste e che quindi vanno verificate. È fondamentale ancora una volta che il PO comunichi efficacemente e in modo chiaro il valore del prodotto e quali sono tutte le peculiarità a cui porre particolare attenzione. Gli utenti sono invitati a partecipare attivamente e, come descritto nel sottocapitolo precedente, può succedere che emergano delle casistiche che non erano state previste. È bene che il PO quindi si allinei costantemente con il tester sull'andamento degli UAT e che si renda disponibile a fornire chiarimenti e a intervenire ogni qual volta si creino situazioni ambigue e che potrebbero bloccare il regolare svolgimento di questa fase finale, prima del rilascio in produzione.
- **Go-live:** una volta terminati gli UAT e ottenuta l'approvazione di tutti i test da parte degli utenti, tutte le nuove funzionalità sono pronte per essere rilasciate in produzione. I rilasci avvengono in determinate giornate che sono pianificate per l'intero anno in date specifiche. Proprio per questo motivo è molto importante che tutte le fasi del progetto non generino ritardi perché uno slittamento del go-live potrebbe avere delle ripercussioni non solo sul progetto stesso, ma anche sulle altre progettualità. Il PO, in contatto con i fornitori, si occupa di

raccogliere i *documenti di deploy*, cioè quei documenti che vengono redatti indicando quelle che sono le tempistiche per il rilascio delle nuove funzionalità e quelle per l'eventuale annullamento, i servizi impattati, le machine e gli elementi dell'infrastruttura che subiranno delle modifiche o che dovranno essere predisposti adeguatamente per un corretto funzionamento dell'intero sistema a seguito del rilascio. La preparazione al rilascio è una fase molto importante che non va assolutamente trascurata e i team interni coinvolti devono ricevere in tempo tutte le informazioni necessarie per accompagnare al meglio i fornitori esterni durante il rilascio. I rilasci, che iniziano generalmente alle 6:30 oppure dopo le 22:00, prevedono la temporanea indisponibilità dei sistemi impattati per tutti i clienti finali, interni ed esterni. Una volta terminato il rilascio, il tester che ha seguito la fase degli UAT insieme ad alcuni utenti effettua, quando possibile, la maggior parte dei test praticabili e grazie alla partecipazione attiva dei fornitori e dei team tecnici interni si risolvono rapidamente eventuali problematiche riscontrate. Solo al termine dei test i sistemi vengono resi nuovamente disponibili. I giorni che seguono sono molto delicati perché si effettua un monitoraggio costante dell'utilizzo delle nuove funzionalità da parte dei clienti finali o degli utenti interni. Sicuramente, risulta molto più semplice monitorare quei sistemi che vengono utilizzati internamente in quanto è possibile disporre di un feedback immediato dagli utenti delle varie direzioni impattate. Tuttavia, grazie all'utilizzo di strumenti appositamente dedicati al monitoraggio dei sistemi e grazie a un lavoro coordinato con il team di post-vendita che monitora attivamente le segnalazioni dei clienti, dopo il rilascio seguono diverse settimane di risoluzione di eventuali bug mai emersi prima. In questo caso il PO comunica con tutti gli attori coinvolti coinvolgendoli quando necessario, organizzando al meglio le segnalazioni e assegnando le opportune priorità. Dato che le nuove funzionalità sono disponibili per tutti i clienti ed eventuali blocchi potrebbero avere notevoli ripercussioni sulla reputazione dell'azienda, nonché sul proprio fatturato, è molto importante che il PO faciliti il più possibile la cooperazione tra le parti per garantire tempi di risposta molto rapidi e soluzioni efficaci.

I Product Owner hanno quindi un ruolo molto importante per la buona riuscita delle progettualità IT della banca e si impegnano in ogni fase adottando le più opportune pratiche che permettano di trasmettere una visione chiara e trasparente del prodotto a tutte le persone coinvolte, con particolare attenzione verso i fornitori e gli utenti. In più, si impegnano per far sì che tutti i team tecnici interni che sono coinvolti per la facilitazione e l'approvazione degli aspetti infrastrutturali e architetturali del sistema informatico della banca intervengano tempestivamente quando richiesto, garantendo che il dialogo tra tutte le parti avvenga in modo fluido e costruttivo.

Riprendendo il discorso fatto in Scrum sui diversi atteggiamenti che un Product Owner assume, è stato possibile riscontrarne l'effettiva applicazione in diverse occasioni, così come presentato di seguito. Si farà riferimento sempre al progetto fittizio Nuovo sistema di accesso all'area clienti.

- 1. Visionario:** durante la fase degli UAT sono emersi alcuni disallineamenti tra le richieste degli utenti e quanto era stato sviluppato dai fornitori esterni. Negli UAT i fornitori non sono coinvolti direttamente, ma vengono interpellati qualora vi siano delle segnalazioni. Trattandosi di una fase molto importante durante la quale è fondamentale avere ben chiaro ormai quali siano le diverse funzionalità, è stato necessario comunicare in modo chiaro il valore che si celava dietro ogni segnalazione così che i fornitori impattati dalle modifiche richieste proponessero delle soluzioni molto più in linea e consapevoli, riducendo il numero di modifiche complessive e ottenendo anche dei suggerimenti e una partecipazione, a volte, proattiva. Questo è stato possibile perché si è cercato di lavorare come una squadra che vedeva nei risultati delle modifiche richieste un reale incremento del valore del prodotto, aumentando quindi la motivazione di tutti.
- 2. Collaboratore:** sempre durante gli UAT, in più occasioni si è reso necessario effettuare dei test ulteriori con i soli fornitori di alcune segnalazioni emerse durante i test con gli utenti. Ciò è stato fondamentale perché ha permesso di chiarire le

questioni emerse e di analizzarle sin da subito in modo mirato ed efficace. In questo modo il raggiungimento degli obiettivi della fase degli UAT è stato facilitato. È interessante notare che vi è stato un incremento della fiducia da parte degli utenti e degli stakeholder, migliorando così anche il clima all'interno dell'intero gruppo di lavoro.

- 3. Rappresentante del cliente:** durante la fase di SIT, entrando per la prima volta in contatto con quanto è stato implementato, sono emerse sin da subito alcune discrepanze rispetto alle indicazioni fornite durante la raccolta dei requisiti. In questa occasione, oltre a rileggere la documentazione approvata, è stato vincente coinvolgere sin da subito gli utenti per avere dei chiarimenti e per recuperare i dettagli più aggiornati. In questo modo è stato possibile ottenere dei feedback immediati che hanno permesso degli interventi tempestivi che hanno ridotto il numero delle segnalazioni che sarebbero comunque emerse nella fase successiva degli UAT.

- 4. Decision Maker:** sicuramente trovarsi per la prima volta ad affrontare un lavoro così importante e delicato, senza esperienza pregressa che dia maggiore sicurezza e confidenza potrebbe non essere semplice. Tuttavia, è stato proprio grazie alla costante necessità di dover affrontare situazioni che richiedevano di compiere delle scelte in tempi brevi e grazie a un ambiente che invita all'autonomia e alla condivisione che in breve tempo è stato possibile raffinare le capacità decisionali. Per esempio, per risolvere una problematica che bloccava il test di una funzionalità è stata proposta una soluzione tecnica in linea con quanto era stato definito in precedenza per altri moduli del sistema chiedendone l'implementazione e testandola prima ancora di avere un'approvazione formale dal team di architetture. In questo caso, sicuramente le competenze tecniche e il background informatico in linea con gli sviluppi hanno facilitato la prontezza nella decisione che si è poi rivelata corretta e approvata in seguito. Ciò ha permesso di non prolungare i tempi intercorsi tra la rilevazione dell'anomalia e la sua risoluzione,

senza dover attendere la disponibilità del team tecnico, impegnato su altre progettualità. È importante sottolineare che il PO non è responsabile di dare soluzioni tecniche senza poi ottenerne l'effettiva approvazione, ma grazie a un preventivo e rapido confronto per le vie brevi con il team tecnico si è evitato di dover attendere la disponibilità per una riunione apposita, risparmiando alcune giornate che avrebbero potuto causare dei ritardi sostanziali.

- 5. Sperimentatore:** benché vi fosse una validazione dei dati all'interno dei campi di input dei form del sistema di accesso, sono state proposte delle modifiche che migliorassero l'esperienza utente. Per esempio, in alcune schermate sono presenti dei campi di input in cui il cliente inserisce dati di natura diversa, come il numero di una pratica, il numero di cellulare, il codice fiscale, ecc. È stato richiesto al fornitore responsabile del front-end di prevedere il trim dei dati, cioè la rimozione degli spazi prima e dopo del dato vero e proprio, sia da web sia da mobile e di prevedere la visualizzazione di una tastiera diversa in base alla tipologia di dato atteso da mobile. Questo ha permesso di sperimentare durante i test varie casistiche di varie modalità con cui i clienti possono inserire i dati garantendo che non si verificassero degli errori. Si pensi, per esempio, a un'operazione di copia-incolla oppure alla funzione di completamento automatico dei browser e dei dispositivi mobili che non sempre introducono dati perfettamente "puliti" da spazi nascosti o altre impurezze che avrebbero potuto compromettere l'esperienza utente qualora non fossero state gestite, come era accaduto in passato.
- 6. Influencer:** durante gli UAT sono emerse alcune richieste che necessitavano in alcuni casi di implementazioni aggiuntive in quando non erano state definite nello specifico durante la raccolta dei requisiti. In queste occasioni, è stato molto importante trovare un compromesso che non avesse un costo aggiuntivo o che non richiedesse un eccessivo prolungamento dei tempi prestabiliti. Le soluzioni proposte non sempre aderivano esattamente alla richiesta e si è quindi reso

necessario influenzare gli utenti o gli altri attori coinvolti, spiegando con chiarezza le motivazioni dietro alle scelte fatte e promuovendole come delle ottime alternative che potessero comunque soddisfare i bisogni emersi. Grazie a una comunicazione trasparente e all'opportuna fiducia ottenuta nel tempo è stato possibile ottenere il consenso, evitando i conflitti e senza aggiungere dei ritardi o dei costi durante la fase degli UAT.

3.5 LA GESTIONE DEI FORNITORI ESTERNI

Un punto di attenzione e una delle sfide più significative per le aziende più articolate e complesse riguarda la gestione dei fornitori esterni. Come abbiamo detto nei primi due capitoli, per poter applicare nuove metodologie è necessario che tutte le persone coinvolte ricevano la giusta formazione e siano allineate con una nuova mentalità aziendale. In questo processo di cambiamento è necessario includere anche tutte le entità esterne che entrano costantemente in contatto con l'azienda. In Santander Consumer Bank Italia tutta la parte di sviluppo software è completamente seguita da diversi fornitori esterni e non sono presenti team interni dedicati direttamente allo sviluppo. Il grande numero di sistemi e di strati che compongono l'architettura della banca ha portato negli anni a stringere dei rapporti di lavoro con diversi fornitori che si dedicano ognuno di un sottosistema diverso. In tal modo è possibile disporre di team dedicati con

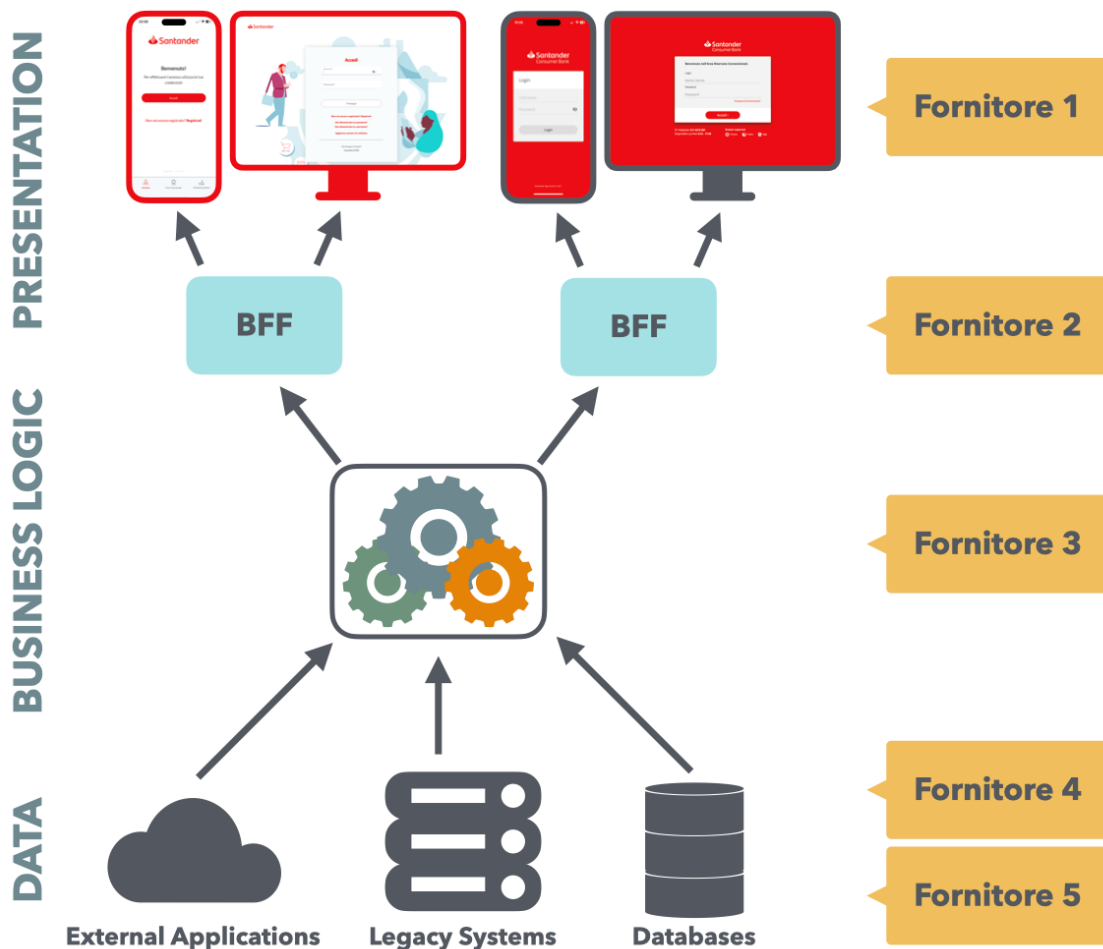


Figura 14 - Esempio di architettura articolata con molteplici fornitori dedicati per ogni strato.

una specializzazione apposita su quelle che sono le differenti tecnologie coinvolte in ogni strato garantendo così una maggiore qualità. Il rovescio della medaglia però è rappresentato dalla complessità di dover gestire e coordinare tutti i fornitori che hanno modalità operative differenti e utilizzano strumenti e metodologie di lavoro diverse. Si presenta quindi una situazione frammentata che porta inevitabilmente a delle conseguenze che se non vengono gestite correttamente rischiano di creare ritardi, disallineamenti e incomprensioni totalmente deleterie per la riuscita dei progetti.

Parlando dell'importanza dei SIT si è già compreso quanto una conseguenza diretta del disporre di un gran numero di fornitori dedicati per parti diverse del sistema sia proprio quella di incorrere in maggiori errori di integrazione spesso dovuti a una scarsa comunicazione e collaborazione. Un altro aspetto critico dovuto alla frammentazione dell'*ownership* dei vari moduli del sistema è da ritrovarsi nell'allungamento dei tempi necessari per l'analisi e la successiva risoluzione di possibili errori riscontrati sia durante le fasi dedicate ai test, sia in seguito alle segnalazioni che emergono in ambiente di produzione. Quando viene segnalata una problematica, individuare il fornitore più indicato a cui assegnare la segnalazione potrebbe essere in alcuni casi un processo molto articolato e complesso. La situazione è spesso aggravata anche dal fatto che non sempre si conoscono gli aspetti tecnici di tutti i sistemi presenti in banca. In molti casi, quindi, la problematica viene inizialmente analizzata dal primo fornitore contattato, che ragionevolmente è stato identificato come quello impattato dalla segnalazione. Tuttavia, potrebbe concludere che la reale causa non risieda nei moduli e nei sottosistemi da lui gestiti, ma in quelli sviluppati da un altro fornitore. Molto probabilmente, il secondo fornitore potrebbe ugualmente riscontrare di non essere il diretto responsabile di un errore in quanto la sua parte si appoggia su moduli sottostanti, rimandando ulteriormente a un terzo fornitore. Questo rimbalzo di responsabilità porta a dei ritardi notevoli nella risoluzione dei problemi e genera del malumore nei vari attori coinvolti che si trovano a scaricarsi le responsabilità senza giungere a conclusioni utili. Una soluzione per mitigare tale situazione sta nel coinvolgere per ogni segnalazione sin da subito tutti i fornitori potenzialmente impattati, ma ciò potrebbe voler significare uno spreco di risorse e un aumento dei costi. È anche in questo contesto che la figura del Product Owner assume notevole importanza

perché diventa molto cruciale coordinare al meglio le diverse parti per evitare che da un lato si allunghino i tempi di risoluzione e che dall'altro intervengano sempre tutti i fornitori anche quando non ve ne è la reale necessità.

La gestione dei fornitori esterni necessita, inoltre, di un grande lavoro di standardizzazione da parte dell'azienda. Definire a priori alcune linee guida condivise e controllare costantemente che queste vengano seguite al meglio fa sì che determinate situazioni spiacevoli si verifichino il meno possibile, garantendo che il lavoro svolto sia il più delle volte già ben strutturato. Si è potuto notare che non sempre vi è uniformità nelle logiche di implementazione adottate per ogni strato che compone l'intera architettura. Si pensi, per esempio, al testo contenuto all'interno di un messaggio d'errore che viene mostrato a front-end. I messaggi d'errore sono stati definiti a priori in modo tale che tutti i fornitori restituiscano sempre lo stesso codice a fronte di una determinata anomalia garantendo la propagazione degli errori dallo strato più basso al livello più alto, restituendo sempre una chiara immagine della situazione. A front-end è stato deciso che qualora non si ricevesse un codice di errore corrispondente a quelli definiti di comune accordo, si sarebbe mostrato al cliente finale direttamente il messaggio d'errore ricevuto dagli strati sottostanti. Tuttavia, a causa di questa scelta, i clienti potevano ritrovarsi con dei messaggi d'errore di natura tecnica che non risultavano chiari o di aiuto per la comprensione della situazione spiacevole, ma ne peggioravano l'esperienza di utilizzo. In tal senso, è stato richiesto al fornitore responsabile del front-end di mappare, per ogni codice di errore non riconosciuto, un errore generico che invitasse il cliente a riprovare in un secondo momento l'azione che stava compiendo a causa di alcuni temporanei malfunzionamenti. Dopo che il fornitore impattato ha applicato la modifica, sono stati effettuati dei test in ambiente di collaudo prima di rilasciarla in produzione. Ciò è stato molto importante perché sin da subito ci si è resi conto di un problema di standardizzazione. Per i codici di errore che il front-end avrebbe dovuto gestire presentando il messaggio d'errore corrispondente, in determinati casi, veniva mostrato direttamente il messaggio ricevuto dagli strati sottostanti. Sostituendo tutti i casi non gestiti con un messaggio d'errore generico, in alcune situazioni i messaggi che ci si sarebbe aspettati di visualizzare a front-end erano scomparsi in favore dell'errore generico. In altri termini, si è scoperto che per alcuni

codici d'errore il messaggio d'errore mostrato a front-end non era quello mappato, bensì direttamente il testo ricevuto dagli strati sottostanti. Se quindi inizialmente solo un fornitore era stato coinvolto per effettuare questa modifica, durante i test è diventato necessario coinvolgere anche altri fornitori che avevano in gestione alcuni moduli impattati dalla modifica. In questo caso, al tempo intercorso dall'analisi della segnalazione al test degli sviluppi di un solo fornitore si sono dovuti sommare altrettanti giorni di lavoro per ampliare l'analisi ed effettuare la modifica anche con un secondo fornitore.

Un altro punto interessante nella gestione dei fornitori riguarda le modalità con cui vengono gestite eventuali modifiche richieste al termine degli sviluppi durante le fasi di SIT e di UAT. Come visto nel sottocapitolo 3.3, è molto probabile che durante i test ci si accorga di alcuni disallineamenti significativi tra quello che è il requisito iniziale e l'effettiva implementazione. È successo in alcuni casi che fosse necessario consultare il documento di analisi funzionale per verificare se alcune delle richieste che erano state indicate non fossero state sviluppate nelle loro interezze. In questi casi eventuali mancanze al termine degli sviluppi sono completamente a carico dei fornitori impattati. Questo vale sia se c'è stata una disattenzione durante le analisi tecniche, sia se non sono stati richiesti dei chiarimenti agli utenti prima della fine degli sviluppi. I fornitori sono quindi invitati a modificare le implementazioni e a svolgere ulteriori analisi e sviluppi assorbendo i costi in quelli già pattuiti a inizio progetto. Una situazione invece diversa la si ha quando compaiono delle modifiche che effettivamente si allontanano da quanto era stato definito in fase di raccolta dei requisiti. Si pensi, per esempio, alla necessità di aggiungere dei campi di input in più in un form, un po' come descritto nell'esempio fittizio fatto nel capitolo 3.3 dell'aggiornamento del documento d'identità in fase di login. Se nel documento dell'analisi funzionale non è presente una richiesta esplicita, o se la richiesta è incompleta, diventa più difficile poter richiedere al fornitore di riassorbire i costi dell'intervento in quanto non opportunamente preventivato. È vero che in alcune situazioni si tratta di modifiche di piccola entità che nella maggior parte dei casi tendono ad essere accolte senza costi aggiuntivi in favore dei rapporti in essere, ma qualora si trattasse di una modifica che richieda un effort maggiore, i fornitori procederanno alla formulazione di una nuova stima dei costi che dovrà essere approvata e che si aggiungerà ai costi già sostenuti

e previsti per quella determinata progettualità. Tale dinamica evidenzia come sia cruciale la capacità di definire con estrema chiarezza i requisiti iniziali e di come sia necessario limitare i margini di ambiguità quando si utilizza un approccio meno flessibile. Tuttavia, in un ambito come quello dei progetti IT nel settore bancario e dei prodotti di credito al consumo, caratterizzato da elevata complessità e da esigenze in costante evoluzione, è inevitabile che per progetti della durata di quasi un anno o più possa emergere la necessità di apportare delle modifiche in corso d'opera che suggeriscono l'evidente necessità di un approccio più flessibile, come quello proposto dallo sviluppo software Agile in contrapposizione a un modello rigido come quello Waterfall.

In tal senso, lavorare costantemente con fornitori esterni implica la necessità di gestire i contratti con delle modalità che garantiscano la corretta implementazione dei principi Agile. Se nei contratti tradizionali le specifiche dei requisiti e i costi vengono definiti rigidamente fin dall'inizio del progetto, in un contesto più flessibile i fornitori sono chiamati ad accettare che i requisiti possano evolversi durante ogni fase per rispondere al meglio alle modifiche che vengono richieste man mano per rimanere in linea con le aspettative dei clienti finali e con il mercato di riferimento. I cosiddetti *Agile Contracts* permettono di superare i limiti dei contratti tradizionali garantendo una maggiore flessibilità senza però ridurre il controllo dei vincoli che regolano il rapporto di lavoro tra il committente e il fornitore. Si tratta di allineare quelli che sono gli obiettivi aziendali con quelli dei fornitori in una collaborazione costante che permetta di modificare le priorità e di introdurre nuove richieste senza compromettere la qualità e le tempistiche delle implementazioni. In tal senso, il framework SAFe prevede una tipologia di contratto Agile definita appositamente per le aziende che si appoggiano su fornitori esterni per gli sviluppi. Si tratta dei contratti *SAFe Managed-Investment*, così come indicato nella guida estesa di SAFe²⁹. Di seguito si presentano alcune tipologie di contratto con le relative caratteristiche che permettono di comprendere al meglio i possibili approcci nella gestione dei rapporti tra i committenti e i fornitori.

²⁹ Scaled Agile Inc., *Extended SAFe Guidance*, disponibile online all'indirizzo <https://scaledagileframework.com/advanced-topics/extended-safe-guidance/> (ultimo accesso in data 15/11/2024).

- **Contratti a prezzo fisso:** si tratta dei contratti tradizionalmente utilizzati dalla maggior parte delle aziende che si interfacciano con dei fornitori esterni. L'idea alla base sta nel definire preventivamente tutte le richieste del committente che dovranno essere realizzate entro una determinata data a fronte di una somma corrisposta al fornitore. In questo modo, il committente si assicura di ricevere esattamente ciò che ha richiesto all'esatto prezzo concordato e nei tempi previsti. Ciò permette di valutare diverse offerte e di scegliere potenzialmente il fornitore che offre il prezzo più vantaggioso. Sebbene a prima vista può apparire un modello vincente in quanto garantisce che a fronte di una spesa il committente ottenga ciò che richiede per quando lo desidera, bisogna notare che vi sono alcuni aspetti critici che non vanno sottovalutati. In primo luogo, va considerato che un contratto a prezzo fisso richiede una mole non trascurabile di lavoro iniziale per definire e delineare dettagliatamente quelle che sono le specifiche della richiesta del committente consentendo così di fissare un opportuno prezzo che non varierà fino alla consegna del prodotto finito. Tuttavia, abbiamo visto che è molto probabile che in corso d'opera si rendano necessarie delle modifiche oppure che i requisiti possano celare delle criticità che sono difficilmente prevedibili con così largo anticipo. In secondo luogo, il committente rischia di ottenere alla fine del rapporto contrattuale un prodotto che così come era stato definito inizialmente ormai non è più in linea con il mercato e con le attuali e reali necessità dei clienti. Tramite delle trattative è possibile negoziare delle modifiche al contratto per accettare eventuali cambiamenti durante lo svolgimento del progetto, ma questo significherebbe investire molto tempo nel cercare di rimaneggiare un contratto che di sua natura è fisso dalla sua stesura. Un'altra insidia che è importante sottolineare riguarda gli interessi di entrambe le parti. Il committente è interessato a ottenere il massimo dal fornitore possibilmente al minor prezzo possibile, mentre per il fornitore la situazione è l'opposta, cioè è nel suo interesse fornire il minimo necessario per adempiere agli obblighi contrattuali massimizzando i profitti. Questi interessi contrapposti spesso generano delle situazioni svantaggiose per entrambe le parti. Da un lato, il committente potrebbe non ritenersi

soddisfatto del lavoro del fornitore percependolo come insufficiente e non in linea con le aspettative iniziali, dall'altro il fornitore per contenere i costi potrebbe ridurre la qualità e limitarsi a fornire il minimo indispensabile rendendosi meno disponibile ad accogliere richieste extra o a risolvere problemi che emergono fuori dal perimetro iniziale. Il progetto fittizio Nuovo sistema di accesso all'area clienti presentato nel sottocapitolo 3.2, può essere un esempio di questa tipologia di contratto. Infatti, solo a fronte dell'approvazione di una stima di dettaglio che contiene già tutti i costi delle implementazioni richieste possono avere inizio gli sviluppi nelle tempistiche previste senza alcuna possibilità di apportare delle sostanziali modifiche in itinere. I fornitori sono tenuti a portare a termine il lavoro previsto in ogni fase rispettando la data di go-live, realizzando quanto richiesto e delineato a monte. Ogni modifica che non rientra nel perimetro iniziale comporta delle trattative e delle modifiche al contratto.

- **Contratti Time & Material:** questa tipologia di contratto è nettamente in contrapposizione con la precedente. In questo caso non è necessario disporre di una richiesta dettagliata iniziale e non si fissa un prezzo complessivo. Si adattano bene per progetti che presentano un'elevata complessità e hanno requisiti variabili. Il committente corrisponde un pagamento in base al tempo e alle risorse effettivamente impiegate dal fornitore per l'implementazione delle diverse richieste che vengono comunicate e definite progressivamente durante il rapporto. Si tratta di una tipologia di contratto che si adatta facilmente ai cambiamenti nei requisiti e che quindi consente di instradare il progetto verso direzioni diverse in base al modo in cui le esigenze del committente cambiano in corso d'opera in linea con le necessità dei clienti finali e del mercato. Dal momento in cui il pagamento avviene sulla base del lavoro che viene effettivamente svolto, il committente è invitato a valutare costantemente i progressi e a fornire dei feedback costanti per definire il lavoro delle fasi successive, in un approccio incrementale e iterativo. Ciò consente una minore pianificazione iniziale per l'avvio di un progetto che può avvenire anche con pochi requisiti minimi che saranno successivamente

dettagliati e affinati nel tempo. Tuttavia, è bene notare che i contratti Time & Material espongono il committente a dei rischi che implicano un controllo continuo. Non definendo un importo fisso a priori, il costo totale del progetto potrebbe lievitare sostanzialmente soprattutto se si incontrano degli imprevisti che ne allungano la durata nel tempo. Inoltre, dal momento in cui il fornitore non è tenuto a rispettare un termine prestabilito obbligatorio, potrebbero crearsi delle situazioni tali per cui i tempi di consegna si allunghino proprio perché non vi sono dei vincoli che incentivino l'ottimizzazione delle tempistiche di implementazione e consegna. In tal senso, il controllo continuo operato dal committente e l'inserimento di clausole o metriche *ad hoc* fa sì che si riduca il rischio a cui è esposto, comportando comunque un opportuno impegno di risorse e di tempo per il monitoraggio costante. Applicare un contratto di questo tipo in un progetto gestito con una metodologia a cascata potrebbe però risultare persino controproducente. In effetti, se si considera il discorso fatto sulla criticità di valutare e di testare le funzionalità implementate solo al termine di tutti gli sviluppi con il rischio che siano poco allineate con la richiesta iniziale, si vedrebbe una dilatazione delle tempistiche eccessiva e un conseguente aumento esorbitante dei costi. È bene quindi prestare molta attenzione alla metodologia adottata per il progetto prima di optare per questa tipologia di contratto con i fornitori.

Per fare un esempio di un caso in cui si potrebbe gestire i rapporti con i fornitori con un contratto di questo tipo, si immagini di commissionare gli sviluppi di una nuova funzionalità che consenta l'accesso all'applicazione mobile dei clienti della banca attraverso un sistema di riconoscimento biometrico. Si decide di seguire un approccio incrementale e iterativo, adottando il framework Scrum. Possono esserci diverse tecnologie che consentono l'implementazione dell'accesso con riconoscimento biometrico e risulta necessario prestare attenzione alle criticità che potrebbero emergere dall'integrazione con l'attuale sistema di accesso. Sottoscrivendo un contratto Time & Material sarà possibile avviare sin da subito alcune sperimentazioni sulla base del requisito generale che potranno essere sviluppate dai fornitori e testate rapidamente dagli utenti man mano che sono

disponibili al termine di ogni Sprint. Ciò renderà possibile l'emergere di nuovi requisiti e di dettagli non previsti inizialmente che verranno implementati successivamente sulla base dei feedback ottenuti dagli utenti e dal team di sviluppo. Grazie a un monitoraggio costante dei progressi sarà possibile comprendere verso quale direzione instradare il progetto apportando delle modifiche che migliorino le implementazioni durante i lavori.

In Tabella 4 sono riportate le principali caratteristiche di queste due tipologie di contratto per facilitarne il confronto. Va notato che esistono anche delle vie di mezzo che includono caratteristiche di una e dell'altra tipologia.

	CONTRATTI A PREZZO FISSO	CONTRATTI TIME & MATERIAL
REQUISITI	Requisiti dettagliati e definiti interamente a priori	Requisiti generali che saranno dettagliati durante il progetto
PREZZO	Prezzo e tempistiche definiti a priori	Il prezzo dipende dalla durata del progetto
FOCUS DEL COMMITTENTE	Evitare modifiche al di fuori del perimetro iniziale	Monitorare il progresso e le tempistiche
RISCHIO	Il rischio ricade principalmente sul fornitore	Il rischio ricade principalmente sul committente
COMPLESSITÀ	Ideali per progetti a bassa complessità	Ideale per progetti sia a bassa che ad alta complessità

Tabella 4 - Contratti a prezzo fisso e contratti Time & Material: un confronto.

Un contratto Agile presenta alcune caratteristiche mutate da quelli descritti precedentemente e altre che lo caratterizzano. È strutturato per favorire la flessibilità e la condivisione del valore tra committente e fornitore, incentivandoli a collaborare per massimizzare il valore entro determinati limiti economici concordati. È bene che garantisca una visibilità costante sul progresso dei lavori e offra un approccio misurato agli investimenti per consentire l'interruzione del progetto al raggiungimento degli obiettivi. In più, è fondamentale che il contratto preveda dei meccanismi di feedback continuo per far sì che il committente tramite dimostrazioni incrementali possa fornire tutte le segnalazioni

necessarie per il miglioramento della qualità del prodotto finale. Un esempio di contratto Agile sono i contratti SAFe Managed-Investment che mirano a ottimizzare il valore generato e a mantenere la flessibilità per adattarsi ai cambiamenti nei requisiti e nelle priorità del business. Nel momento in cui un'azienda decide di impegnarsi in un contratto di questo tipo per la realizzazione di un progetto complesso e articolato, si avvia una fase iniziale di *pre-commitment*. Durante questa fase, che ha come fine ultimo quello di definire opportunamente i termini del contratto, il committente e il fornitore condividono alcune responsabilità specifiche che gettano le basi al rapporto di lavoro. Da un lato, il committente si impegna anzitutto a comprendere quelli che sono i principi fondamentali dei contratti Agile, definendo e comunicando una *mission* chiara del progetto e lavorando a stretto contatto con il fornitore per stabilire una *vision* e una *roadmap*. Dall'altro lato, il fornitore deve eseguire un'analisi preliminare della fattibilità, valutare l'allineamento delle necessità del committente con le proprie competenze principali e stimare le risorse e i costi necessari per i periodi contrattuali iniziali. Insieme, sono tenuti a identificare:

- il *Minimum Viable Product* (MVP), cioè la versione minima di un prodotto che include solo le funzionalità essenziali per soddisfare le esigenze iniziali del committente per poi raccogliere i feedback utili per gli sviluppi futuri,
- il *Solution Intent*, cioè la descrizione iniziale, sia fissa, sia variabile, dei requisiti e delle funzionalità di un prodotto che guida lo sviluppo lasciando spazio all'adattamento e all'evoluzione,
- la prioritizzazione del backlog iniziale per la pianificazione del Planning Interval (PI – presentato nel capitolo 2.2)

Inoltre, entrambe le parti sono chiamate a stabilire i parametri del quadro economico, come, per esempio, gli impegni finanziari per i primi Planning Interval. In alcuni casi, il fornitore per aggiudicarsi il finanziamento iniziale potrebbe fornire una stima preliminare. In altri casi, invece, si adotta un approccio *pay as you go* nel quale il committente finanzia progressivamente il progetto, sulla base dei risultati conseguiti e sulle evidenze oggettive dell'avanzamento dei lavori. È importante notare che il committente ha la

possibilità di valutare diversi fornitori durante questa fase di pre-commitment fino alla decisione di assegnazione del contratto.

A seguito dell'assegnazione del contratto ha inizio il rapporto di lavoro tra committente e fornitore che è caratterizzato dalle seguenti fasi:

- **Preparazione del Planning Interval:** in questa fase sia il fornitore, sia il cliente, investono tempo e risorse per preparare il necessario per l'evento di PI Planning così come previsto dal framework SAFe. È possibile che questo primo evento avvenga già durante la fase di pre-commitment.
- **PI Planning:** come visto precedentemente questo evento è molto importante per la buona riuscita dell'intero PI. Di conseguenza, le parti coinvolte sono invitate a pianificare nel dettaglio le attività che andranno svolte durante il primo PI.
- **Esecuzione del PI:** in questa fase, è il fornitore che impiega maggiormente le risorse per l'effettivo svolgimento delle attività concordate per il Planning Interval. Il committente viene coinvolto solo durante i System Demo, ma potrebbe risultare necessario per progetti più complessi un coinvolgimento più frequente mediante le Solution Demo che non seguono la pianificazione del PI, in modo tale che le attività svolte siano sempre allineate con le richieste.
- **Valutazione del PI:** è molto importante che in ogni punto di controllo e validazione dell'avanzamento dei lavori il committente valuti i progressi del Planning Interval e analizzi le metriche concordate tra le parti. Grazie all'evento di Inspect and Adapt al termine di un PI è possibile valutare il progresso complessivo e identificare dei miglioramenti. È in questo momento che il committente può decidere di modificare i livelli di finanziamento per proseguire i lavori oppure di terminare l'iniziativa qualora si valuti che il valore atteso sia stato raggiunto. In base alle decisioni prese, si procederà con la pianificazione del Planning Interval successivo.

Il contratto Agile SAFe Managed-Investment è a tutti gli effetti una soluzione vantaggiosa per entrambe le parti dato che prevede una collaborazione costante favorendo il controllo economico, ma garantendo un'adeguata flessibilità. Alla base di questa tipologia di contratto vi sono i principi lean andando a ridurre i tempi di rilascio in produzione di nuove funzionalità e riducendo lo sviluppo di funzionalità inutili o superflue. Grazie alla struttura contrattuale costituita da una combinazione di componenti fisse e variabili, vengono garantiti sia gli interessi del committente, sia quelli del fornitore ottimizzando il valore per entrambi. In un certo senso, il successo di uno dipende dal successo dell'altro, raggiungendo degli obiettivi comuni e non contrapposti. È possibile avere una stima iniziale pattuendo un numero iniziale di Planning Interval e identificando il Minimum Viable Product. Tale stima risulta più precisa rispetto a quella che si avrebbe con una lista ricca e completa di requisiti dettagliati. Le parti concordano che a partire dall'ambito iniziale di lavoro definito dal MVP si aggiungeranno nuove funzionalità in modo progressivo in periodi incrementali consistenti in uno più Planning Interval. Grazie a questo contratto il committente ha un controllo totale sul prezzo e sulla spesa poiché può decidere quando proseguire o interrompere i finanziamenti in base al valore raggiunto al termine dei cicli di sviluppo. Il fornitore, dal lato suo, si assicura un flusso di lavoro stabile e una relazione a lungo termine basata sulla soddisfazione del committente. Inoltre, misurando costantemente i progressi e il valore ottenuto, il fornitore può ottimizzare le proprie risorse gestendole al meglio e risparmiando, senza però disattendere le esigenze del committente che rinnoverà il rapporto in essere per successivi periodi di sviluppo e di incremento del valore. Si crea quindi un rapporto collaborativo e stabile nel quale il committente ottiene una soluzione attuale rispetto alle esigenze che emergono in un ambiente in continua evoluzione e alla richiesta del mercato, mentre il fornitore massimizza il proprio ritorno economico adattandosi alle reali necessità.

4. ACTIONS

Analizzando le lessons learned rilevate direttamente sul campo grazie alla tesi in azienda e al tirocinio presso Santander Consumer Bank è stato possibile ricavare un'immagine nitida delle sfide che un'azienda complessa e articolata deve affrontare costantemente nella gestione dei progetti IT nel settore bancario e dei prodotti di credito al consumo. È curioso notare come in realtà tali sfide possano essere concettualmente molto simili alla maggior parte delle aziende così come indicato nei primi capitoli in cui non ci si è focalizzati su un settore specifico, ma si è visto come con un'opportuna analisi è possibile plasmare un nuovo modello Agile e lean di organizzazione del lavoro. L'importante è che le lessons learned vengano considerate dei punti di partenza da cui imparare e migliorare pianificando delle *Actions* correttive e concrete. In effetti, parlare di azioni da intraprendere significa andare oltre la necessaria riflessione teorica e l'analisi delle problematiche iniziando a costruire un vero e proprio percorso di cambiamento e miglioramento continuo che possa portare all'evoluzione di un'azienda. In questo contesto è opportuno avere chiara la definizione di tattica e quella di strategia. Si tratta di due approcci differenti che sono però complementari per il raggiungimento di un obiettivo. La strategia rappresenta un piano ampio che è la visione a lungo termine per il raggiungimento di un determinato obiettivo. Si tratta di un approccio ad alto livello che guida un'intera organizzazione verso lo scopo finale. Per la definizione di una strategia ci si interroga su quello che è il punto da raggiungere sulla base di un'opportuna consapevolezza del presente, dello stato attuale dell'azienda. Per definire una strategia è importante identificare quelli che sono gli obiettivi principali da raggiungere tenendo bene a mente quelle che sono le risorse a disposizione. È bene effettuare un'attenta analisi dei *competitor* e dei *comparable* per conoscere più da vicino le altre realtà nel mercato di riferimento e comprendere quali sono i punti di forza delle loro strategie e quali invece risultano punti deboli, disponendo così di un insieme di *best and worst practices* preziose per la definizione di una strategia vincente. In più, è molto importante che ci si interroghi su come ottenere vantaggi competitivi e risultati di successo avendo bene in chiaro qual è il bacino della clientela di riferimento studiandone i bisogni e le caratteristiche

peculiari. L'insieme di queste pratiche fa sì che si possano pianificare delle azioni che risulteranno vincenti e che apporteranno dei risultati positivi a lungo termine.

La tattica, invece, può essere considerata il modo in cui verrà messa in pratica una strategia. È costituita da azioni concrete, specifiche e a breve termine che servono per la realizzazione di parti di una più ampia strategia. Ragionando su una singola tattica ci si interroga su quelle che sono le azioni concrete per il raggiungimento di un obiettivo specifico e ristretto in un lasso di tempo altrettanto ristretto. Si analizzano i passi concreti da compiere per progredire nel piccolo di una più ampia visione complessiva, scegliendo gli opportuni strumenti e le metodologie specifiche. Una tattica potrebbe quindi essere quella di organizzare degli incontri settimanali per la verifica dell'avanzamento dei lavori di un progetto, ma senza essere parte di una più ampia e lungimirante strategia potrebbe apportare dei benefici solo momentanei; mentre ciò che è ancora più importante è il raggiungimento di un obiettivo più grande, che porti l'azienda verso una crescita e un miglioramento nel futuro. È quindi grazie a una strategia che si possono sviluppare e definire delle tattiche che risultino efficaci e allo stesso tempo i risultati di ogni tattica permettono di affinare in corso d'opera la strategia, garantendo una sempre maggiore consapevolezza per il raggiungimento di un obiettivo a lungo termine in linea con la vision aziendale.

Ogni azione concreta vede come punto di partenza la ricerca di un primo passo da compiere che generalmente comporta la definizione di un processo di adozione di nuove pratiche in modo graduale. Ciò permette di sperimentare e di avere un feedback immediato da parte di un gruppo ristretto di persone identificate per la loro propensione al cambiamento. In questo modo, sarà più semplice introdurre il cambiamento in sempre più parti dell'azienda ed essere pronti a gestire attriti e difficoltà. È bene notare che non tutte le azioni devono per forza essere parte di un più ampio progetto di evoluzione: nel momento in cui si identificano delle criticità e degli ostacoli è possibile intervenire immediatamente. Parlando, ad esempio, dei problemi di comunicazione riscontrati tra i fornitori, non è necessario attendere la definizione di un più ampio piano d'azione, ma si può iniziare sin da subito a sensibilizzare i gruppi di lavoro sul tema così da poter godere dei benefici sin da subito e da comprendere se vi siano delle criticità nascoste che

si celano dietro la più generica comunicazione frammentata e scarsa. È poi in questo senso che altre soluzioni richiedono una visione a lungo termine e un cambiamento sostanziale che coinvolge l'intera organizzazione.

Infine, è bene sottolineare che non bisogna abbattersi se i risultati attesi non saranno immediati. I processi di cambiamento richiedono anche molti anni prima di restituire una percezione tangibile dei benefici che hanno generato. Non bisogna quindi fermarsi dopo le prime azioni intraprese e considerarle come un intervento isolato, poiché l'efficacia delle singole azioni può avere una breve durata. In un ambiente in cui la richiesta del mercato e le aspettative dei clienti sono in continua evoluzione, vi sono costantemente nuove azioni che devono essere intraprese a seguito di una attenta analisi delle lesson learned quotidiane. È in questo contesto che bisogna avvalersi di strumenti adatti per favorire la sperimentazione e il monitoraggio puntuale dei risultati, basando le azioni future su dati concreti. Le Actions sono quindi il prodotto dell'osservazione delle criticità emerse e permettono di realizzare degli interventi pratici e concreti con l'obiettivo di migliorare l'efficienza, la qualità e la capacità innovare un'azienda complessa così come è il caso di Santander Consumer Bank Italia. In tal senso, si parla di Agile Adoption come prima azione concreta e come primo passo verso una vera e propria rivoluzione dell'intera organizzazione: l'Agile Transformation.

4.1 AGILE ADOPTION

Con il termine *Agile Adoption* si descrive uno dei primi passi che un'organizzazione può fare se intende integrare i principi e le pratiche Agile. Inizialmente si esplorano le opportunità e si identificano gli step da seguire generalmente circoscritti a un numero limitato di persone. Nel caso in cui il modello organizzativo attuale lavori a compartimenti stagni, si identificano o si formano dei nuovi piccoli team. Ciò è il punto di partenza per avviare la formazione di queste persone, base fondamentale senza la quale la semplice introduzione di un framework Agile non sortirebbe gli effetti benefici desiderati. Non si attua quindi un cambiamento esteso a tutta l'organizzazione, ma si inizia a sperimentare con alcuni progetti pilota che nasceranno sin da subito seguendo la metodologia Agile. Il team sperimentatore, oltre ai principi dell'Agile verrà introdotto agli strumenti operativi messi a disposizione dal framework Scrum. È importante che il nuovo team famigliarizzi con i concetti di iterazione, di collaborazione e di feedback continuo, iniziando a sperimentare e a vivere in prima persona quelli che sono i vantaggi di Agile in modo controllato e in un ambiente appositamente predisposto. Si tratta a tutti gli effetti di effettuare un esperimento pratico dopo l'acquisizione degli aspetti più teorici per entrare nell'ottica del lean and agile mindset e nella praticità della gestione con strumenti nuovi e approcci radicalmente diversi da quelli a cui si era abituati (ad es. pratiche legate al modello Waterfall). Questa prima esperienza vissuta dalle persone appositamente identificate perché potenzialmente più inclini al cambiamento e favorevoli alla messa in discussione delle attività svolte fino a quel momento fa sì che si crei in loro una maggiore consapevolezza sull'importanza dei principi Agile e sull'impatto positivo che possono avere nel lavoro quotidiano e nei confronti del cliente finale. È importante identificare un progetto pilota che si presti per sua natura a essere facilmente ricondotto alle nuove modalità di lavoro, consentendo al gruppo sperimentatore di valutarne l'efficacia in uno specifico contesto aziendale. In tal senso, i progetti IT, come detto precedentemente, si prestano molto bene e rappresentano un ottimo punto di partenza per l'adozione Agile mirata e in piccolo. Anche i leader e referenti del nuovo team dovranno ricevere la corretta formazione per poter coordinare con nuove modalità il lavoro che verrà svolto, preparando il giusto ambiente e garantendo che le operazioni possano avvenire

in un clima che lascia sufficiente spazio alla sperimentazione senza avere il timore di commettere degli errori. I leader quindi non si dovranno occupare di controllare il lavoro svolto, bensì facilitarlo rimuovendo tutti i possibili ostacoli e preparando un terreno solido. Se ciò non accadesse si rischierebbe di fallire nel processo di adozione creando del malcontento nelle persone coinvolte che si farebbero un'idea sbagliata della nuova metodologia. Uno degli obiettivi è proprio quello di far sì che tutti i membri del nuovo team apprezzino e riscontrino un ritorno positivo durante la sperimentazione così da diventare delle figure di riferimento e promotrici per l'intera azienda.

In questa fase di adozione di Agile, le aziende introducono nuovi strumenti come Kanban oppure quelli messi a disposizione da Scrum per poter monitorare a livello di singolo team l'andamento dei lavori. È molto importante che si pratichi un monitoraggio costante perché proprio in questa fase è possibile rendersi conto di eventuali ostacoli che potrebbero emergere e diventa cruciale trovare delle possibili soluzioni consultando chi sta vivendo in prima persona la nuova esperienza di lavoro. In linea con i principi lean e Agile, sono i lavoratori stessi che affrontando nuove sfide si rendono conto delle problematiche ricorrenti e devono essere ascoltati invitandoli al ragionamento e alla formulazione di azioni da intraprendere che potrebbero far aggirare gli ostacoli. Solo così diventa possibile iniziare un lavoro consapevole di rifinitura per creare un modello adatta alla propria realtà aziendale che, potenzialmente, potrebbe essere esteso anche ad altre aree dell'azienda in un secondo momento. Possiamo identificare alcuni elementi chiave per il successo dell'Agile adoption:

- **Team sperimentatore:** come abbiamo detto è importante creare un team apposito che inizi la sperimentazione delle nuove metodologie. La scelta delle persone che lo compongono andrà fatta con estrema consapevolezza delle loro soft skill, identificando coloro che in azienda si sono già precedentemente contraddistinti per la loro apertura alla sperimentazione e che presentano una mentalità di crescita.

- **Progetti pilota:** anche il progetto con cui sperimentare l'adozione di una nuova metodologia è importante che sia specifico e che si presti a fungere da caso studio per verificare i benefici e le difficoltà. Si vuole creare un primo esempio positivo che possa essere replicato sempre con maggiore consapevolezza anche per altri progetti via via sempre più complessi e articolati.
- **Formazione e coaching:** i membri del team sperimentatore dovranno partecipare a diverse sessioni di formazione prima ancora di scoprire quale sarà il progetto pilota. È bene coinvolgere delle figure esperte come gli Agile Coach che organizzano dei corsi di formazione specifici e che trasmettano con lezioni interattive i concetti più importanti, allenando il team a essere autonomo con gli strumenti e il ragionamento nuovo. È fondamentale che i corsi non avvengano con una normale lezione frontale in quanto non sarebbe efficace: dovranno essere invece delle vere e proprie occasioni di crescita e di partecipazione attiva del team. Questo favorisce non solo una migliore acquisizione dei concetti, ma diventa anche l'occasione per la creazione di un gruppo coeso, che sia aperto alla collaborazione e che accetti i pensieri e le diversità di ogni membro.
- **Monitoraggio continuo:** bisogna implementare una sistema di monitoraggio continuo che permetta di avere sempre ben chiaro lo stato dell'implementazione di Agile. Si potrebbe affermare che l'adozione di Agile è a tutti gli effetti un progetto e quindi è bene che venga gestito come tale. Per questo è bene che si raccolgano feedback costanti dalle persone coinvolte, oltre che si analizzino dati misurabili per valutare gli effettivi impatti positivi e l'efficacia delle pratiche adottate.
- **Supporto della leadership:** è estremamente importante il ruolo assunto dalla leadership per la buona riuscita dell'Agile adoption. Infatti, i leader sono chiamati a supportare il cambiamento, prima di tutto in questa fase iniziale, ma anche successivamente e costantemente giorno dopo giorno promuovendo una cultura aperta all'innovazione e all'ascolto delle proposte di tutti. È bene che i leader

introducano tutto ciò che è in loro potere per far sì che gli ostacoli e tutti gli elementi che mettono a rischio il successo dell'adozione vengano rimossi tempestivamente in modo proattivo. Per favorire il miglioramento continuo si devono accertare che i membri del team sperimentatore non si limitino perché preoccupati di commettere degli errori durante la sperimentazione. Il clima che si deve creare dovrà permettere di effettuare scelte sbagliate senza conseguenze mirate a colpevolizzare una persona piuttosto che l'altra, bensì a valorizzare le azioni correttive che sono state intraprese.

Bisogna considerare che una delle maggiori difficoltà che si potranno incontrare riguarderà la resistenza al cambiamento da parte delle persone coinvolte. In contesti aziendali più tradizionali è difficile convincere che l'abbandono di pratiche ormai consolidate nel tempo possa apportare dei vantaggi concreti. Da un lato questo potrebbe essere causato dal fatto che non si riesca inizialmente a fidarsi del fatto che una nuova metodologia possa essere applicata realmente in quel determinato contesto aziendale, in quanto si riescono solo a evidenziare tutte le criticità che potrebbero emergere senza fare un passo in più e concentrarsi sul come esse possano essere risolte. La formazione qui gioca un ruolo fondamentale perché è proprio dando una visione corretta e ribaltando certi punti di vista che è effettivamente possibile convincere le persone a sperimentare nuovi approcci. Un'interpretazione troppo rigida della metodologia Agile, che la vede come una serie di regole rigide da seguire, è quindi un'altra possibile causa della scarsa volontà di abbracciare il cambiamento ed è per questo che è bene promuovere prima di tutto una nuova cultura e questo può essere fatto ancora prima di anche solo nominare Agile. Si pensi, per esempio, all'importanza che si potrebbe dare sin da subito alla comunicazione tra colleghi e alle azioni che possono essere intraprese per migliorare il clima e l'umore di tutti i dipendenti dell'azienda. In questi casi non si starebbe introducendo una vera e propria metodologia nuova, ma si starebbero comunque preparando le basi per un'accoglienza meno restia di un cambiamento perché sono già state messe in moto alcune accortezze che favoriscono la nascita di una nuova cultura aziendale e la successiva adozione di una nuova metodologia Agile e snella. Un altro aspetto da considerare

riguarda i disallineamenti che si creeranno tra i team che adottano ancora pratiche tradizionali e quelli che adottano le nuove pratiche. Inevitabilmente, in una realtà complessa le diverse progettualità tendono a intrecciarsi e a influenzarsi le une con le altre. La diversa modalità di lavoro potrebbe creare dei conflitti se non risulta ben chiara una visione condivisa da tutti. È ovvio che in un primo momento ciò non sarà semplice né da gestire né da vivere, ma ancora una volta saranno i leader e i coach che dovranno essere di grande aiuto e supporto per l'intera azienda.

4.1.1 ROADMAP

Quanto detto finora, trova la sua formalizzazione all'interno dell'articolo *Agile Adoption Roadmap* scritto da Mario Moreira e pubblicato in *The Agile Journal* nell'edizione di novembre 2010³⁰. Si tratta di un insieme di linee guida per l'adozione efficace delle metodologie Agile all'interno delle aziende, andando a suddividere il processo di adozione in tre fasi principali così come rappresentato in Figura 15.

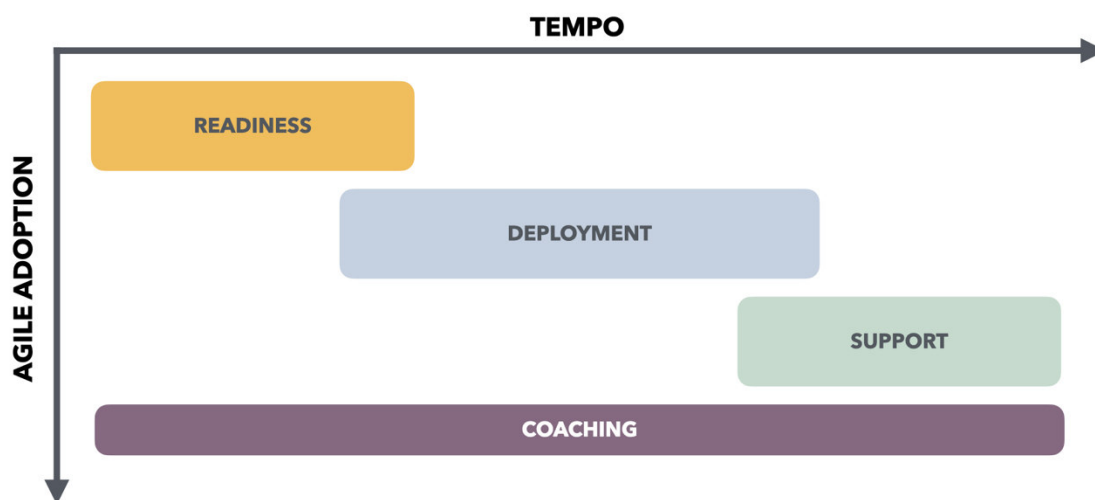


Figura 15 - Agile Adoption roadmap.

READINESS

Questa fase mira a valutare lo stato attuale del team di prodotto e a stabilire quelle che sono le aspettative per il cambiamento, definendo quindi una strategia complessiva per l'adozione di Agile. In questa fase ci si concentra molto sul Pensare Agile. Questa fase può durare dagli uno ai tre mesi e le attività principali da svolgere sono:

- **Determinare l'idoneità del prodotto:** identificare un prodotto che si presti meglio di altri per trarre il massimo vantaggio dal processo di adozione Agile.

³⁰ Moreira M., "Agile Adoption Roadmap", in Agile Alliance, *The Agile Journal*, november 2010, disponibile online all'indirizzo <https://www.agilealliance.org/wp-content/uploads/2016/01/Agile-Adoption-Roadmap-1110-AgileJournal-Mario-Moreira.pdf> (ultimo accesso in data 15/11/2024).

- **Valutare la volontà del team:** potrebbe essere necessario riformulare il team di lavoro comprendendo quali membri risulterebbero favorevoli e quali contrari o incerti rispetto ai cambiamenti.
- **Valutare le capacità del team:** bisogna identificare le competenze e la conoscenza di Agile dei membri del team così da pianificare l'opportuna formazione.
- **Valutare le pratiche ingegneristiche e la mentalità Agile del team:** potrebbe essere che vi siano alcune pratiche già adottate che in qualche modo risultino efficaci e che quindi potrebbero essere preservate.
- **Identificare un coach Agile:** è fondamentale la presenza di un coach Agile che guidi il team a partire dalla fase di preparazione e durante tutto il processo di adozione Agile.
- **Stabilire una strategia e un piano per l'implementazione di Agile:** come detto precedentemente, il processo di adozione Agile è un progetto a tutti gli effetti ed è quindi opportuno definire una lista di priorità di attività da svolgere per implementare Agile sulla base degli obiettivi. L'adozione deve avvenire in modo efficace.
- **Stabilire riunioni periodiche di monitoraggio:** un po' come i Daily, fissare delle riunioni regolari permette di valutare il progresso delle attività e di mantenere il focus sul processo di adozione.
- **Definire ruoli e organizzazione Agile:** i ruoli dei membri del team definiti nel capitolo 2 dovranno essere assegnati in modo appropriato.

- **Valutare il supporto degli stakeholder:** identificare gli sponsor Agile tra gli stakeholder permette di comprendere il livello di supporto e di prevedere eventuali rischi causati dalla mancanza di sostegno.
- **Stabilire framework e pratiche Agile:** il team deve formulare una struttura iniziale da migliorare nel tempo selezionando un set di pratiche Agile, come l'utilizzo di Sprint, degli eventi Scrum, e un framework che più si adatti alle esigenze aziendali.
- **Considerare i tool Agile necessari:** bisogna individuare e iniziare a conoscere alcuni tool Agile che supportino il processo. Si potrebbero quindi valutare e installare i moduli appositamente sviluppati per la gestione Agile di strumenti software già utilizzati in azienda come ServiceNow oppure Microsoft Project. Si possono utilizzare anche strumenti fisici come delle lavagne o delle pareti condivise.
- **Focalizzarsi sulla mentalità Agile e sul cambio culturale:** come già evidenziato più volte in questa tesi è fondamentale che si crei una cultura Agile aziendale, promuovendo l'empowerment, la servant leadership e un atteggiamento proattivo, senza perdere di vista l'Agile mindset prima ancora degli strumenti.

DEPLOYMENT

Durante questa fase, il team inizia la vera e propria applicazione sul campo delle pratiche Agile per lo svolgimento di un progetto. In questa fase si passa effettivamente ad Agile Agile. Anche questa fase può durare da uno a tre mesi e può essere implementata in modo incrementale. Le attività da svolgere indicate da Moreira sono:

- **Impostare lo strumento di pianificazione Agile:** lo strumento Agile che è stato scelto nella fase precedente va installato (se si tratta di uno strumento software) oppure preparato (in caso di strumento fisico come una parete) per renderlo

operativo. In questo modo tutti i membri del team sono pronti a utilizzarlo senza perdere di vista il lavoro da svolgere.

- **Fornire formazione Agile:** la formazione specifica in base alle scelte che sono state fatte nella fase precedente è molto importante. In questo modo ogni membro riceve una formazione adatta al ruolo ricoperto nel team e sugli strumenti e metodologie scelte durante la preparazione. Per esempio, un Product Owner riceverà una formazione specifica sulla gestione del backlog e sulla scrittura delle funzionalità, uno Scrum Master potrebbe seguire un corso e ottenere una certificazione apposita, e così via. È importante anche una formazione di gruppo sul lavoro in team e sulle responsabilità di ogni membro. È bene che tutti i membri del team possano effettuare costantemente domande per ricevere dei chiarimenti a ogni dubbio che emerge durante l'implementazione della nuova metodologia.
- **Implementare la metodologia e le pratiche Agile:** a questo punto in cui il progetto è avviato, il coach Agile aiuta il team a mettere in pratica le metodologie e le pratiche Agile offrendo supporto per almeno i primi tre Sprint così da dare consigli preziosi che aiutino a entrare nell'ottica corretta e a sfruttare al meglio gli strumenti del framework scelto (es. come gestire gli eventi Scrum).
- **Riunioni periodiche per monitorare l'implementazione:** in questa fase centrale è bene che si partecipi alle brevi riunioni periodiche pianificate per assicurarsi che le attività di implementazione siano in linea con gli obiettivi e che non si perda il focus sul percorso di adozione Agile.

SUPPORT

L'obiettivo di questa fase è quello di fornire il supporto necessario al team che sta implementando Agile. In tal senso, si verifica che effettivamente le pratiche vengano adottate correttamente e si applicano dei miglioramenti ove necessario. A differenza delle fasi

precedenti, questa dura di più delle altre, dai sei ai nove mesi. Al termine di questa fase si presuppone che emergano delle figure di riferimento che hanno ormai appreso e interiorizzato Agile, diventando quindi delle guide interne per il team senza più necessitare del costante affiancamento di un coach Agile. In questa fase, Moreira consiglia di:

- **Fornire coaching e mentoring continuo:** ogni problema e ogni sfida che emerge durante l'esecuzione del progetto è l'occasione per ricevere supporto. In questo modo il team ha un punto di riferimento, il coach Agile, che può dare chiarimenti e aiutare il team a superare gli ostacoli. È importante notare che l'interazione costante del coach Agile con il team permette anche di identificare tempestivamente situazioni pericolose per la riuscita del processo di adozione Agile. Per esempio, alcuni membri potrebbero voler tornare a un approccio più tradizionale e il coach dovrà occuparsi di riportare il focus nella giusta direzione.
- **Validazione sul campo delle pratiche:** è importante che il coach Agile partecipi occasionalmente agli eventi Scrum, come lo Sprint Planning o le Sprint Review, per validare l'applicazione delle pratiche Agile da parte del team. Grazie allo sguardo di un osservatore esterno è possibile aiutare i membri del team a comprendere se stanno applicando correttamente Agile e se ci sono delle azioni di miglioramento e adattamento.
- **Riunioni periodiche per monitorare l'adozione Agile:** bisogna incontrarsi periodicamente per discutere i progressi nell'adozione Agile. La frequenza di queste sessioni può diminuire con il tempo quando il team ha abbracciato con sempre maggiore successo la nuova cultura Agile.
- **Valutazioni periodiche sull'adozione Agile:** in questo modo è possibile valutare quale sia lo stato attuale di adozione di pratiche in linea con i principi Agile e il livello di mentalità Agile. È possibile confrontare ogni valutazione con le

precedenti per comprendere il livello di adozione raggiunto e identificare ulteriori miglioramenti.

- **Ulteriore attenzione agli strumenti e all'automazione:** è bene verificare che gli strumenti utilizzati siano sufficienti e qualora fosse necessario provare a introdurre di nuovi o a richiedere moduli aggiuntivi di quelli già esistenti. Per esempio, potrebbe essere molto importante puntare sull'automazione per velocizzare alcune operazioni di gestione del team.
- **Formare riferimenti Agile interni:** è importante che tutti i componenti di un team Agile collaborino con i coach Agile. Questo permette di verificare che Scrum Master, Product Owner e Developers abbiano una vera mentalità Agile e che stiano applicando le pratiche in modo corretto ed efficace. In tal senso, diventa molto importante identificare alcuni membri appassionati di Agile che hanno risposto positivamente e rapidamente al cambiamento e che abbiamo raggiunto un grado di autonomia e conoscenza delle pratiche Agile tale per cui possano aiutare gli altri. Questo è fondamentale perché, una volta terminata la fase di Support, saranno loro che internamente continueranno a mantenere il focus del team, in quanto i coach Agile avranno lasciato l'azienda.

È interessante notare come in questa roadmap per l'Agile Adoption ci sia un bilanciamento tra mentalità e strumenti. In un certo senso, notiamo che non è possibile operare un'adozione Agile se tutte le persone coinvolte non comprendono e non fanno propri prima di tutti i principi e poi gli strumenti. L'Agile Coach è una figura molto importante che è presente per tutto il processo di adozione; verifica che non ci si allontani da quella che è la corretta implementazione della metodologia Agile e il corretto utilizzo degli strumenti scelti. Al termine della roadmap di Agile Adoption, il coach Agile lascia in eredità ai membri del team selezionati il ruolo di guida per il gruppo.

4.1.2 EVITARE LO SCRUMBUT

Il framework Scrum è sicuramente molto efficace per iniziare a lavorare con dei team che applicano la metodologia Agile. Come abbiamo visto, grazie a una guida ben dettagliata e all'aiuto di coach Agile il processo di Agile Adoption potrebbe portare già i suoi frutti in meno di un anno dall'avvio dell'adozione di una nuova metodologia. I modelli e i framework a disposizione sono molteplici e ci siamo concentrati su Scrum analizzando nel dettaglio gli strumenti che mette a disposizione dei team che intraprendono un percorso di implementazione di una nuova metodologia che sostituisca quella tradizionale. Si è più volte evidenziato come non esista un modello univoco e prestabilito da adottare in qualsiasi realtà aziendale, ma è fondamentale effettuare degli adattamenti sulla base del contesto in cui si applicano i principi Agile e lean. In tal senso, è quindi normale che ogni azienda abbia necessità diverse, adattando le pratiche di Scrum al proprio contesto specifico. Tuttavia, è proprio durante questo processo di adattamento che si rischia di incorrere in errori, motivo per cui è fondamentale il supporto costante di una persona esterna che controlli l'avanzamento dell'implementazione di Agile. Uno degli errori più comuni quando si sceglie di adottare il framework Scrum è il cosiddetto *Scrum-But*. Questa situazione anomala indica chiaramente che non si stanno seguendo le corrette pratiche previste dal framework, ma si stanno introducendo delle eccezioni che modificano la struttura fondamentale. In altre parole, si dice di fare Scrum, ma non lo si applica completamente. In una pagina del sito web ufficiale di Scrum³¹, si presenta lo schema ricorrente degli ScrumButs:

(Scrum ma) (Motivazione) (Soluzione alternativa)

Di seguito sono riportati alcuni esempi di ScrumButs.

1. *"(We use Scrum, but)
(having a Daily Scrum every day is too much overhead,
(so we only have one per week.)"*

³¹ Scrum.org, *What is ScrumBut?*, disponibile online all'indirizzo <https://www.scrum.org/resources/what-scrumbut> (ultimo accesso in data 16/11/2024).

2. *"(We use Scrum, but)*

(Retrospectives are a waste of time,)

(so we don't do them.)"

3. *"(We use Scrum, but)*

(we can't build a piece of functionality in a month,)

(so our Sprints are 6 weeks long.)"

In pratica quello che succede è che quando emergono delle problematiche, invece di risolverle trovando delle soluzioni ricercando la causa alla radice e poi agendo di conseguenza, si preferisce applicare una modifica al framework come soluzione alternativa considerandola migliore. Bisogna quindi prestare molta attenzione perché si potrebbe pensare che questi compromessi possano essere necessari per adattare il framework al contesto aziendale, ma in realtà rischiano di compromettere i benefici attesi. Se non si seguono con attenzione i ruoli, le regole e le tempistiche di Scrum si rischia di incorrere in situazioni spiacevoli, dove le inefficienze e i risultati insoddisfacenti non rispecchiano le aspettative dell'adozione di un approccio Agile.

Lo ScrumBut si verifica spesso quando un'azienda o un singolo team si avvicina al framework senza aver compreso interamente i principi e pensando di doverlo adattare alle proprie abitudini consolidate negli anni. In questo modo però si perde completamente di vista il senso di adozione di una nuova metodologia, perché si rischia di aver fatto uno sforzo invano e di aver completamente snaturato Scrum ritornando al punto di partenza. Infatti, si rischia di continuare a lavorare con la stessa metodologia precedente avendola potenzialmente complicata con l'aggiunta di elementi fortemente maneggiati e snaturati del nuovo framework. Tra le cause più comuni possiamo trovare:

- **Resistenza al cambiamento:** è difficile adottare una nuova metodologia se non si accetta di dover cambiare l'approccio tradizionale. Alcune aziende tendono a mantenere pratiche consolidate negli anni perché preoccupate di dover

abbandonare la loro zona di comfort. Questo lo si può notare in quei contesti aziendali in cui i gruppi di lavoro e i manager sono abituati a metodi di lavoro lineari e gerarchici. La presenza di processi rigidi e di gerarchie non favorisce l'applicazione dei principi di Scrum che richiedono una forte autonomia, trasparenza e delle iterazioni rapide, in quanto potrebbero essere percepiti come una minaccia al controllo tradizionale. Un esempio tipico di ScrumBut diretta conseguenza di questa causa è:

Usiamo Scrum ma,
non possiamo prendere decisioni autonome,
quindi attendiamo l'approvazione del manager.

In questo esempio, la mancata decentralizzazione del potere decisionale non permette al team di lavoro di muoversi autonomamente, ritrovandosi bloccato ogni qual volta si renda necessaria l'approvazione del manager, con la probabile formazione di un collo di bottiglia.

- **Mancanza di formazione:** come abbiamo visto è molto importante che tutti i membri del team siano formati e conoscano al meglio i principi e gli strumenti della metodologia Agile. Inoltre, è molto importante che la suddivisione dei ruoli e le relative responsabilità di ogni membro del team siano chiare e assegnate correttamente. Per esempio, non si può immaginare che un Project Manager che lavora con scadenze rigide e vincoli ben precisi, possa ricoprire il ruolo di Scrum Master senza l'adeguata formazione e l'opportuna ridefinizione delle proprie responsabilità e dell'approccio nel suo lavoro quotidiano. In più, se il team non conosce l'importanza e il significato degli eventi Scrum si rischia che questi non vengano presi in considerazione perché ritenuti una pura e semplice perdita di tempo. Un esempio di ScrumBut che scaturisce da questa causa è:

Usiamo Scrum ma,
i Daily Scrum durano 45 minuti,
quindi ne facciamo solo uno alla settimana.

In questo esempio, non conoscendo il reale obiettivo di un Daily Scrum, si sfrutta l'incontro per affrontare anche nel dettaglio problematiche che una volta presentate potrebbero essere analizzate in un secondo momento. Siccome risulta poco efficiente impegnarsi ogni giorno per ben 45 minuti per un evento che non dovrebbe superare i 15 minuti allora si è optato per ridurre a una volta alla settimana la frequenza dei Daily, perdendo completamente di vista il significato e l'importanza di tale evento.

- **Tempistiche serrate e pressioni sulle consegne:** lavorare sotto pressione per ottenere dei risultati rapidi può portare a tagliare e a trascurate aspetti essenziali di Scrum. La necessità di dover implementare in fretta una grande quantità di nuove funzionalità senza gestire correttamente le richieste assegnando una priorità chiara perché tutte hanno la stessa priorità e generalmente tutte sono urgenti, rischia di rendere disorganizzato il lavoro del team che invece di concentrarsi sugli obiettivi a breve termine stabiliti per un determinato Sprint, si ritrova a gestire molte più richieste di quelle presenti nello Sprint backlog. Ciò comporta un aumento del carico di lavoro che non è più bilanciato come previsto dalla pianificazione iniziale e in alcuni casi si opta per risparmiare del tempo annullando gli incontri previsti dal framework. Un esempio di ScrumBut causato da questa situazione è:

Usiamo Scrum ma,
ogni elemento nel backlog è urgente,
quindi iniziamo nuovi sviluppi senza le retrospettive.

In questo caso, la mancata prioritizzazione delle attività da svolgere ha portato il team a non disporre del tempo necessario per comprendere come migliore negli Sprint successivi e ciò comporterà che difficilmente il lavoro del team potrà affinarsi e migliorare continuamente nel tempo, Sprint dopo Sprint.

- **Struttura organizzativa rigida:** in alcune aziende la struttura organizzativa esistente non si adatta facilmente all'implementazione di Scrum in quanto presenta

una suddivisione in silos funzionali. Ciò significa che vi è un team solo per lo sviluppo, un team solo per i test, uno solo per l'architettura, ecc. rendendo quindi difficile la creazione di team cross-funzionali così come previsto da Scrum. Inoltre, il forte collegamento e le dipendenze che si creano tra i diversi gruppi rischiano di rallentare l'implementazione. Un esempio di ScrumBut per questa casistica è:

Usiamo Scrum ma,
sviluppiamo diversi prodotti in parallelo,
quindi abbiamo un reparto separato con i tester.

In questo esempio, il team si dovrà interfacciare con il reparto dei tester che qualora risultassero impegnati potrebbero ritardare il flusso di lavoro e rappresentare un collo di bottiglia nell'intera catena.

- **Ambiguità sui ruoli e sulle responsabilità:** una causa comune dello ScrumBut è l'assenza di chiarezza sui ruoli chiave di Scrum, come quello dello Scrum Master o del Product Owner. Si pensi a uno Scrum Master che non si occupa di rimuovere i possibili impedimenti e non facilita quindi il lavoro del team perché in realtà svolge attività più legate al ruolo di Project Manager. Un esempio di ScrumBut per questa causa è:

Usiamo Scrum ma,
abbiamo molti task di sviluppo,
quindi il nostro Scrum Master sviluppa software nel team.

In questo esempio, l'azienda chiede allo Scrum Master di assumere anche il ruolo di sviluppatore per supportare il team. Tuttavia, il suo ruolo è quello di garantire che il framework Scrum venga seguito correttamente e potrebbe non disporre del tempo necessario per supportare adeguatamente il team se impegnato anche in un altro ruolo.

Il fenomeno dello ScrumBut ci permette di comprendere quanto sia importante prestare costantemente attenzione durante l'adozione della metodologia Scrum, assicurandosi

che i valori e i principi su cui si fonda il framework vengano rispettati. Adottare questa nuova metodologia comporta un impegno non indifferente e un atteggiamento proattivo da parte di tutte le persone coinvolte per evitare di cadere in errore, accettando compromessi che potrebbero ridurre l'efficacia e trasformare Scrum in una versione ibrida poco strutturata. Il valore di ogni pratica proposta e la sua applicazione nel contesto aziendale deve essere chiaro e compreso da tutti, mantenendo un equilibrio tra la flessibilità e la necessità di seguire alcune logiche aziendali non ancora modificate. La presenza di un coach esterno che aiuti a comprendere come meglio adattare il framework alle proprie esigenze senza snaturarlo è fondamentale e consente anche di facilitare il processo di adozione, formando nuovi punti di riferimento sulla metodologia Agile tra i membri del team. Lo ScrumBut è uno dei possibili errori comuni che possono essere commessi durante l'adozione della metodologia Agile, ma è importante essere consapevoli che possono emergere altre situazioni che potrebbero minare il processo di adozione. Per esempio, alcuni errori comuni riguardano l'assenza di un supporto adeguato da parte della leadership, la mancanza parziale o la totale assenza di un reale cambiamento culturale profondo, l'incapacità di fornire obiettivi chiari e misurabili e la mancata riorganizzazione delle strutture organizzative che non si adattano al nuovo modello collaborativo. È essenziale che durante il monitoraggio del processo di adozione si verifichi l'insorgere di tali errori e si intervenga tempestivamente per riuscire a godere in breve tempo dei benefici della metodologia Agile, diventando uno strumento efficace per affrontare le sfide quotidiane e per promuovere l'innovazione dell'azienda.

4.2 AGILE TRANSFORMATION

Se l'Agile Adoption potrebbe rappresentare un ottimo punto di partenza per introdurre nuove metodologie all'interno di un'azienda, concentrandosi su specifici team e progetti, non bisogna trascurare quanto detto sulle realtà aziendali più articolate e complesse. La necessità di applicare un modello che scali i principi di Scrum all'intera organizzazione aziendale comporta uno sforzo molto maggiore che ha un impatto in qualsiasi area in cui l'azienda è suddivisa. Quando si parla di *Agile Transformation* si punta a ridefinire l'intera organizzazione, creando un sistema integrato in cui i principi e i valori Agile e lean si inseriscono in ogni livello aziendale, dal lavoro svolto quotidianamente alle strategie che definiscono obiettivi a lungo termine. La trasformazione Agile è quindi un processo molto complesso e ha una durata molto lunga in quanto richiede tempo, risorse e soprattutto una volontà da parte di tutta l'azienda di abbracciare un cambiamento radicale sia culturale, sia organizzativo. Si tratta di un cambiamento che ha impatti sull'intera azienda, dalla leadership alla struttura dei team di ogni direzione. I processi decisionali e le modalità di pianificazione e gestione delle risorse cambiano e ogni elemento viene ridefinito per creare un ambiente di lavoro snello e flessibile. A differenza dell'Agile Adoption, il cambiamento di una trasformazione Agile modifica non solo l'approccio lavorativo e la mentalità di singoli gruppi, ma modifica completamente l'azienda nel profondo.

Così come l'adozione Agile presenta alcune sfide importanti, anche l'Agile Transformation presenta molti punti di attenzione e numerose difficoltà che la rendono ancora più impattante e delicata. Il cambiamento di mentalità a livello aziendale richiede di superare resistenze interne che in questo caso non possono essere mitigate dalla scelta di persone più propense al cambiamento, ma riguardano tutti i dipendenti, compresi coloro che hanno una mentalità fissa e non sono favorevoli al cambiamento perché legati a modelli rigidi e tradizionali di gestione. Per questo è necessario un investimento significativo nella formazione di tutti i dipendenti lavorando a stretto contatto con ognuno di loro e prevedendo delle sessioni dedicate proprio a quei gruppi di persone che di base potrebbero rifiutare più facilmente il cambiamento, diventando un ostacolo dell'intera trasformazione. Anche la leadership gioca un ruolo fondamentale perché non dovrà applicare

un nuovo approccio solo con un numero ristretto di team di lavoro, bensì dovrà guidare il cambiamento radicale con grande determinazione, comunicando costantemente la vision e i benefici della nuova metodologia per tutti i livelli dell'organizzazione. I leader di ogni area dell'azienda hanno quindi un ruolo fondamentale e devono muoversi all'unisono per promuovere una nuova cultura che guidi le scelte quotidiane e il lavoro di tutti i team. Questo è davvero importante perché una prima possibile criticità è proprio rappresentata dal rischio di una trasformazione incompleta, dove solo alcune aree aziendali adottano i principi Agile, mentre altre rimangono ferme sui modelli tradizionali. Questo potrebbe essere dovuto al mancato desiderio del leader di queste aree di abbracciare il nuovo modello in quanto preoccupato di perdere il controllo e il ruolo avuto in precedenza. Per questo è bene che la leadership si muova unita in ogni fronte così da evitare la generazione di frizioni interne che ridurrebbero inevitabilmente l'efficacia complessiva della trasformazione.

Il processo di Agile Transformation è quindi molto complesso e il cambiamento proposto non può avvenire in modo frammentario o superficiale. Trattandosi di un lungo percorso è necessario che si mantenga alta l'attenzione per evitare che le resistenze o le difficoltà organizzative ne compromettano l'efficacia o ne rallentino il completamento. Deve essere definito un piano strategico di trasformazione che definisca con attenzione il modo in cui le diverse aree aziendali saranno via via coinvolte e riorganizzate, favorendo un cambiamento graduale che non interrompa l'operatività e che garantisca una transizione graduale. È fondamentale promuovere una comunicazione costante e trasparente tra tutti i livelli organizzativi, valorizzando le diversità di tutte le aree aziendali. Vanno previsti momenti di verifica e di aggiustamento del percorso intrapreso, così da poter raccogliere dei feedback che permettano di identificare tempestivamente le criticità e le possibilità di miglioramento. Il processo di trasformazione va considerato come un progetto molto articolato e complesso. Vi è quindi la necessità di considerare l'azienda come un sistema che riceve delle modifiche progressive. Grazie a un approccio iterativo che faccia progredire la trasformazione un passo alla volta è possibile monitorare la risposta del sistema ai cambiamenti introdotti man mano e reagire alle evidenze emerse con consapevolezza e determinazione. La creazione di indicatori chiave di performance (KPI)

specifici per il processo di trasformazione permette di monitorare i progressi e di valutare l'impatto delle singole iniziative intraprese di volta in volta per garantire che l'intero processo rimanga allineato con gli obiettivi prefissati.

Durante il processo di trasformazione è importante identificare delle figure interne all'azienda che possano diventare il punto di riferimento per tutti i colleghi del cambiamento Agile. Si tratta di veri e propri ambasciatori di Agile che non solo promuovono costantemente i principi della nuova metodologia, ma chiariscono i dubbi dialogando costantemente con tutti i colleghi. In aziende complesse e articolate andrebbero identificati un paio di ambasciatori del cambiamento per ogni area in modo tale che tutti i dipendenti possano fare riferimento a delle persone con cui hanno già lavorato in passato e di cui si fidano maggiormente. Gli ambasciatori non devono essere per forza i leader delle diverse aree, i quali già sono invitati a promuovere e a favorire la nuova cultura aziendale, ma possono essere quei dipendenti che si distinguono per la loro proattività e per il loro entusiasmo con cui accolgono le sfide quotidiane e le opportunità di un cambiamento. In questo modo essendoci un rapporto alla pari tra gli ambasciatori e gli altri colleghi si riduce il rischio che il cambiamento venga vissuto come una imposizione calata dall'alto e favorisce quindi un'adesione spontanea e convinta, garantendo un'adozione più organica e condivisa.

Prima di esplorare nel dettaglio la roadmap proposta per l'implementazione di SAFe è bene chiarire quali sono le differenze principali tra l'Agile Adoption e l'Agile Transformation. Sebbene l'Agile Adoption potrebbe sembrare un primo passo verso una completa trasformazione, bisogna comprendere al meglio le caratteristiche di una e dell'altra. In questo modo non si rischia di muoversi in modo confusionario e di compiere una trasformazione parziale o addirittura di credere che l'adozione Agile in diverse aree aziendali sia lo stadio finale di una trasformazione completa dell'organizzazione. Di seguito sono presentate le principali differenze così come mostrate in Tabella 5.

- **Agile Way of Working:** l'adozione Agile prevede che il team di lavoro incominci a familiarizzare e a utilizzare gli strumenti a supporto per l'implementazione della metodologia Agile. Si può dire quindi che è invitato ad Agire Agile. In questo

senso potrebbe succedere che il lavoro limitato ad alcuni gruppi non permetta di vivere a pieno una nuova mentalità e si rischia di limitarsi all'azione e non a un vero e proprio cambio di mentalità. A causa anche del fatto che gli altri team non oggetto di adozione continuano ad applicare le stesse metodologie tradizionali, la mentalità di controllo non cambia radicalmente e quindi si rischia di svolgere alcune attività ancora allo stesso modo, con solo alcuni spiragli di novità. Invece, nella trasformazione Agile, si sviluppa prima di tutto la mentalità Agile a livello aziendale, comprendendo a fondo le motivazioni che portano ad agire in modo Agile. A tutti gli effetti, grazie a un piano strutturato e a un processo lento, si può affermare che l'obiettivo finale sarà quello di Essere Agile a tutti gli effetti.

- **Rapidità del cambiamento:** adottare le pratiche di Agile è generalmente abbastanza rapido. Subito dopo l'opportuna formazione il team può già iniziare a seguire le pratiche di Scrum, assegnando i ruoli ai membri del team, pianificando gli eventi e lavorando in Sprint. È quindi un punto di partenza vantaggioso e più immediato e permette di avere un riscontro positivo in poco tempo sulla nuova metodologia. Tuttavia, la situazione cambia quando si parla di Agile Transformation. Qui la velocità del cambiamento è molto più lenta e richiede un approccio

	AGILE ADOPTION	AGILE TRANSFORMATION
AGILE WAY OF WORKING	Agire Agile	Essere Agile
RAPIDITÀ DEL CAMBIAMENTO	Più veloce	Più lento
FOCUS	Sulle pratiche	Sul risultato
TEMA	Il team	L'intera organizzazione
AZIONE PRINCIPALE	Adottare pratiche conosciute e affermate	Creare nuove pratiche adatte al proprio contesto
OUTPUT	Produttività del team	Consegna del valore

Tabella 5 - Agile Adoption e Agile Transformation: un confronto.

molto più strutturato che coinvolga pesantemente la leadership. I benefici offerti sono maggiori di una semplice adozione, ma i risultati arriveranno dopo un periodo più lungo e intenso. Si tratta di un processo che mira ad avere un ritorno a lungo termine sull'organizzazione.

- **Focus:** durante l'adozione Agile il team si concentra sulla corretta comprensione delle nuove pratiche da utilizzare già sperimentate in altri contesti Agile. Si comincia a utilizzare i tool scelti per gestire al meglio il lavoro e si cerca di riorganizzarsi sulla base delle nuove pratiche. Tuttavia, quello che manca nel processo di adozione è una visione a più ampio raggio e lungo termine su ciò che cambierà una volta che le nuove pratiche saranno adottate, su come si misurerà il cambiamento e su quali saranno i rischi che dovranno essere mitigati. In una trasformazione Agile ci si concentra quindi anche su tutti quegli aspetti che condizionano e valutano il reale risultato finale.
- **Tema:** questo aspetto è molto importante in quanto il tema dell'adozione Agile è la trasformazione di un team specifico, mentre il tema di una trasformazione è l'intera organizzazione. In poche parole, l'adozione si concentra maggiormente sul team che può essere composto da una decina di membri e che ha come obiettivo quello di adottare con successo le pratiche Agile, mentre la trasformazione Agile si pone come obiettivo di scalare Agile in tutta l'organizzazione. Non si tratta quindi solo di estendere a tutti i team IT la nuova metodologia, ma anche a tutti i team di qualsiasi area aziendale indipendentemente dalla funzione. Quindi la trasformazione Agile riguarda indistintamente le direzioni IT, del marketing, delle vendite, dei legali, ecc. Solo in questo modo tutta l'azienda potrà affermare di essere Agile.
- **Azione principale:** durante l'adozione Agile si può prendere come punto di riferimento alcuni dei modelli di successo di altre aziende. Nella maggior parte dei casi questo è sufficiente per iniziare a comprendere la nuova modalità di lavoro e per

beneficiare dei primi risultati positivi. Tuttavia, è importante ricordarsi che ogni azienda va trattata come un caso a sé e le aziende che realmente sono Agile cambiano costantemente il proprio modello, aggiornandolo in un miglioramento continuo in linea proprio con la definizione stessa di Agile. Quindi è bene che se da un lato una prima adozione di Agile possa prevedere l'utilizzo di pratiche conosciute, dall'altro la trasformazione Agile lavori in una direzione che permetta la definizione di un modello Agile appositamente plasmato sulla realtà aziendale con maggiore consapevolezza e con l'impegno di migliorarlo costantemente.

- **Output:** grazie all'adozione Agile si aumenta la produttività, migliorando il modo in cui i team lavorano, riducendo gli sprechi nei processi. Diventando quindi più efficienti si è in grado di produrre di più e con una qualità maggiore. Sebbene questo sia molto importante, bisogna ricordarsi che è fondamentale fornire sempre più un valore maggiore al cliente finale. In tal senso, il processo di trasformazione Agile si concentra principalmente su massimizzare il valore fornito al cliente con un ritorno benefico più ampio sul business e sulla competitività nel mercato.

4.2.1 ROADMAP

Per procedere all'effettiva realizzazione di una trasformazione Agile bisogna definire un piano strategico dettagliato e si deve disporre di una sequenza ordinata di attività che coinvolgano tutte le aree e i livelli dell'azienda. Si tratta di un percorso molto complesso in quanto non si limita all'introduzione di metodologie, ma implica la creazione di un ambiente in cui i valori e i principi lean e Agile vengano vissuti quotidianamente, dall'operatività alla strategia aziendale. Per formulare una roadmap efficace bisogna identificare le necessità specifiche dell'azienda e adattarsi al contesto e agli obiettivi che si intende raggiungere a lungo termine. Vi sono diversi elementi fondamentali e tappe di una trasformazione Agile: l'adozione di un lean and agile mindset, la creazione di team cross-funzionali, l'ottimizzazione dei flussi di lavoro, la definizione di una vision strategica ben chiara. Non bisogna dimenticare che oltre all'azienda, anche gli stakeholder vanno coinvolti nella trasformazione rimuovendo tutte le barriere organizzative e promuovendo una leadership che guidi con determinazione e coerenza il complesso e lungo cambiamento. Una roadmap efficace deve prevedere diverse fasi successive ognuna delle quali permette di fare un passo avanti verso l'obiettivo finale, prevedendo un monitoraggio costante che garantisca che ogni step stia realmente portando al successo della trasformazione. In tal senso, attraverso l'implementazione del framework SAFe presentato nel secondo capitolo, essendo pensato proprio per scalare i principi di Scrum

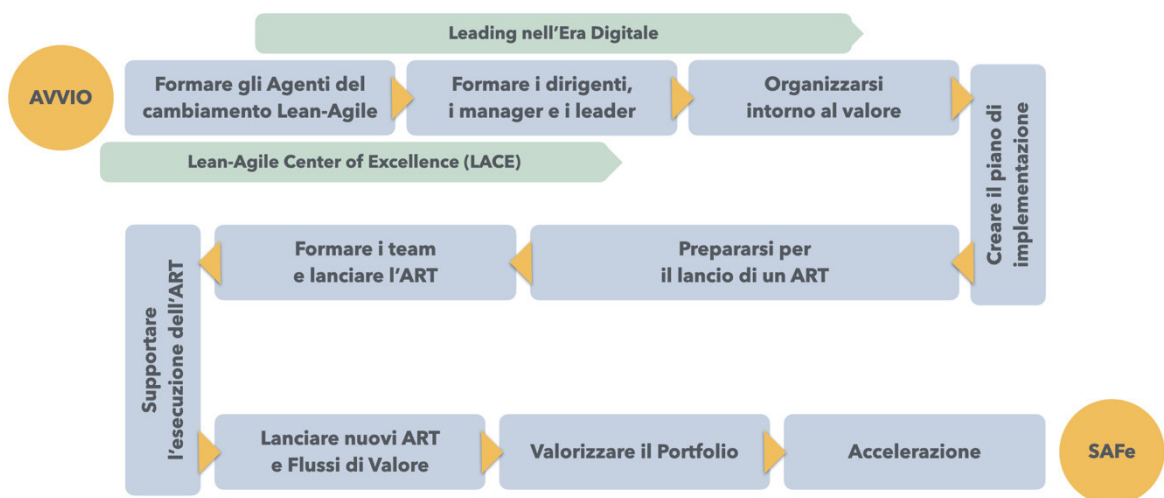


Figura 16 - SAFe Implementation Roadmap semplificata.

all'intera organizzazione, è possibile compiere una vera e propria trasformazione aziendale. Sul sito web ufficiale di SAFe è possibile trovare la *SAFe Implementation Roadmap* con una guida che spiega nel dettaglio quali sono le diverse fasi che un'azienda dovrebbe seguire per implementare efficacemente SAFe³².

In Figura 16 è rappresentata una versione semplificata della SAFe Implementation Roadmap. Si tratta di un percorso suddiviso in quattordici tappe per guidare le aziende nella trasformazione lean e Agile dell'intera organizzazione mediante l'implementazione del framework SAFe. Di seguito si presentano le tappe della roadmap con una breve descrizione dei punti centrali di ognuna.

1. Raggiungere il punto di svolta

La trasformazione Agile inizia quando un'azienda raggiunge il cosiddetto *punto di svolta*. Questo può avvenire sia perché la leadership riconosce che vi sia la reale necessità di adottare nuove metodologie diverse da quelle tradizionali per affrontare la concorrenza globale e i rapidi cambiamenti del mercato di riferimento, sia perché vi sono dei dati oggettivi che dimostrano come l'azienda necessiti di un cambiamento. Per esempio si possono notare dei cali nelle vendite, ritardi costanti nelle progettualità, notevoli inefficienze o difficoltà nel rispondere ai bisogni effettivi dei clienti. Diventa quindi importante condividere queste informazioni all'interno dell'azienda per sensibilizzare i dipendenti e creare un senso di urgenza e la necessità di un cambiamento importante. Si pensi, per esempio, a una azienda che non riesce a rilasciare aggiornamenti alla propria applicazione mobile con una frequenza equiparabile a quella di altre aziende dello stesso settore, dando ai clienti un servizio inferiore.

2. Formare gli Agenti del Cambiamento Lean-Agile

Raggiunta la consapevolezza della necessità di un cambiamento, diventa molto importante formare un gruppo di persone che diventino i punti di riferimento

³² Scaled Agile Inc., *SAFe Implementation Roadmap*, disponibile online all'indirizzo <https://scaledagile-framework.com/implementation-roadmap/> (ultimo accesso in data 21/11/2024).

della trasformazione. Bisogna individuare all'interno dell'azienda alcune persone per le quali si investirà in corsi di formazione e certificazioni appositamente pensati per formare gli Agenti del Cambiamento. Questo è molto importante perché gli agenti avranno il compito fondamentale di guidare e di supportare l'intera organizzazione durante il percorso di trasformazione. Dovranno affrontare al meglio tutte le sfide e le resistenze che si presenteranno durante il cambiamento e promuovere una cultura lean e Agile in tutta l'azienda formando a loro volta i membri dei team. Sono i diretti responsabili della creazione di una mentalità Agile e di un ambiente favorevole per una trasformazione di successo. Non è sempre necessario che si tratti di figure interne all'azienda: infatti, ci si può affidare anche a consulenti esterni certificati SAFe.

3. Creare un Centro di Eccellenza Lean-Agile (LACE)

Per supportare e gestire al meglio il processo di trasformazione, è consigliabile istituire un gruppo apposito chiamato LACE (Lean-Agile Center of Excellence). Si tratta di un team multifunzionale e i membri che lo compongono devono provenire dai diversi dipartimenti dell'azienda. Il LACE funge da punto focale per tutte le iniziative Agile offrendo supporto strategico e operativo durante tutto il percorso di cambiamento. Fornisce formazione continua a tutti i dipendenti e si occupa anche del monitoraggio costante dell'andamento della trasformazione. È sua responsabilità verificare che le pratiche lean e Agile siano implementate nel modo corretto e secondo le linee guida SAFe, assicurandosi che tappa dopo tappa l'organizzazione rimanga allineata agli obiettivi strategici. Inoltre, si occupa di garantire l'allineamento tra i team e la leadership. Il LACE può essere formato sia da dipendenti che hanno ricevuto l'adeguata formazione specializzante su Agile, sia da consulenti esterni.

4. Formare i dirigenti, i manager e i leader

Come abbiamo detto precedentemente, la leadership ha un ruolo cruciale nel successo di una trasformazione Agile. Non si può pensare che l'intera azienda

adotti un radicale cambiamento senza che i dirigenti, i manager e i leader ricevono una formazione specifica sul framework e su quelli che sono i principi lean e Agile. Bisogna che la leadership sia pronta a sostenere attivamente il cambiamento, prendendo decisioni informate, proponendosi come modelli di comportamento per l'intera organizzazione. La formazione per i dirigenti, i manager e i leader deve concentrarsi principalmente su:

- i principi lean e Agile e sulla Lean-Agile Leadership,
- le strategie per comunicare efficacemente la vision e i benefici della trasformazione,
- le tecniche di gestione del cambiamento.

Per esempio, i manager della direzione IT&Operations seguono dei corsi specializzati che li preparano sulle nuove modalità di coordinamento e di gestione di team autonomi e che lavorano con un approccio iterativo.

5. Leading nell'Era Digitale

Essere leader nell'era digitale è una grande sfida. Bisogna adattarsi rapidamente alle nuove tecnologie e alle esigenze mutevoli del mercato. Per questo motivo è molto importante che chi è alla guida di intere aree aziendali accresca una mentalità orientata all'innovazione, promuovendo la sperimentazione di nuove tecnologie e incoraggiando l'apprendimento continuo. In tal senso, i leader sono quindi chiamati a:

- incentivare la sperimentazione e l'innovazione,
- creare un ambiente che incoraggi la collaborazione,
- creare un ambiente che favorisca il miglioramento continuo,
- supportare l'adozione di nuove tecnologie che migliorino i processi.

I leader, quindi, non devono temere l'insorgere di nuove tecnologie e la possibilità di ascoltare i suggerimenti e i consigli dei propri team che per primi possono venire a conoscenza di nuove strade da percorrere e di nuovi strumenti da adottare per restare al passo con l'evoluzione rapida del mondo tecnologico. Per esempio, in un contesto bancario l'implementazione di automatizzazioni

potrebbe semplificare molti processi e la sperimentazione dell'utilizzo dell'intelligenza artificiale potrebbe rivelarsi una scelta vincente, con una riduzione del carico di lavoro degli uffici impattati.

6. Organizzarsi intorno al Valore

Aspetto molto importante del framework SAFe, così come presentato nel capitolo 2, è che l'intera azienda si riorganizzi attorno ai flussi di valore. Bisogna quindi compiere un lavoro attento di analisi per identificare tutti i flussi di valore attualmente presenti in azienda in modo tale da avere una mappatura ben precisa di tutti i Value Stream. Ciò permette di organizzare i team di lavoro in modo tale che si concentrino sulla creazione di valore per il cliente finale andando così a massimizzare l'efficienza e la reattività. In questo modo, sarà possibile facilitare una consegna rapida e continua di prodotti di qualità appositamente costruito intorno alle richieste e alle necessità del cliente. In questa tappa è quindi importante che:

- si identifichino i flussi di valore principali dell'azienda,
- si riducano le dipendenze tra i team,
- si creino dei team cross-funzionali che possano autonomamente generare valore in ogni fase di lavorazione.

I membri dei team tradizionali dedicati a funzioni specifiche vengono suddivisi in modo tale da creare nuovi team attorno ai prodotti principali dell'azienda. Per esempio è possibile creare un team cross-funzionale che si dedichi unicamente al Prestito Personale, un altro team cross-funzionale che si dedichi unicamente al Leasing, ecc. È importante però che i team cross-funzionali siano indipendenti da altri team e quindi possano apportare valore al proprio prodotto senza dover dipendere dal lavoro di altri team. Potrebbe quindi essere necessaria della formazione mirata che fortifichi le competenze dei membri del team, ognuno nella sua specializzazione. È molto importante prestare massima attenzione alla creazione di questi team. Bisogna assicurarsi che posseggano tutte le competenze necessarie per gestire autonomamente un flusso di valore. In altre parole, non esisterà

più un team di tutti i Product Owner, un team di tutti gli Sviluppatori, un team di tutti gli Architetti, un team di Marketing, ecc., ma in base alle competenze richieste per un determinato flusso di valore costruito attorno a un prodotto, ci sarà il team di quel prodotto, con il proprio Product Owner, i propri sviluppatori, il proprio tester, il proprio specialista di marketing, e così via.

7. Creare il piano di implementazione

Una volta che i Value Streams sono stati ben identificati e analizzati, l'azienda è ora pronta a implementare un piano dettagliato per iniziare a lavorare con una nuova metodologia. Nel piano è importante definire le fasi, le tempistiche e le risorse necessarie per avviare la vera e propria implementazione di SAFe. Sarà necessario pianificare anche le modalità di monitoraggio dei progressi per avere sempre una chiara immagine sull'andamento del processo di trasformazione anche durante i primi passi più applicativi e pratici del framework.

8. Prepararsi per il lancio di un ART

Prima di poter avviare un ART (Agile Release Train), così come presentato nel capitolo 2, bisogna:

- definire attentamente e chiaramente i ruoli e le responsabilità,
- identificare i team coinvolti e prepararli con una formazione specifica,
- pianificare i dettagli per il primo evento di Program Increment Planning.

Grazie a una preparazione meticolosa, il lancio dell'ART potrà avvenire con meno problematiche e con una migliore collaborazione tra i diversi team. Per esempio, un'azienda specializzata nello sviluppo software forma i suoi sviluppatori, i tester e gli analisti funzionali per il lancio del primo ART con dei workshop specializzati sul lavoro iterativo, sulla collaborazione e sulla gestione delle priorità e del backlog.

9. Formare i team e lanciare l'ART

Ha finalmente inizio il primo Program Increment e si lancia il primo ART. I team vengono formati costantemente per gestire al meglio il lavoro man mano che lo svolgono, così da avere anche degli esempi pratici sui quali basarsi per apprendere più efficacemente e rapidamente. Inizia quindi il ciclo di sviluppo incrementale con un focus particolare sul rilascio continuo di valore. Per esempio una banca potrebbe lanciare il primo ART con un PI Planning durante il quale si definiscono quelli che sono gli obiettivi per il primo trimestre.

10. Supportare l'esecuzione dell'ART

Questa tappa successiva al lancio del primo ART è fondamentale perché permette di fornire supporto continuo ai team attraverso un'attività di coaching continua. In questo modo è possibile identificare in anticipo eventuali ostacoli e aggirarli tempestivamente o comunque trovare una soluzione che li risolva e che consenta al team di restare in linea con gli obiettivi dell'ART definiti durante il PI Planning. Il supporto continuo permette anche un apprendimento costante e favorisce l'adozione di un'ottica di miglioramento continuo. I coach sono quindi tenuti a supportare:

- la risoluzione di ostacoli
- l'implementazione di best practices
- la gestione di eventuali dipendenze tra team

11. Lanciare nuovi ART e flussi di valore

Dopo i primi mesi di lavoro, grazie all'esperienza acquisita durante l'esecuzione del primo ART, è possibile espandere la trasformazione dell'azienda e l'implementazione di SAFe avviando nuovi ART e Value Streams. L'obiettivo è quello di estendere i benefici e di scalare le pratiche Agile a livello aziendale, allineando nuovi ART ai flussi di valore prioritari. In questo caso sarà importante assicurarsi che i nuovi ART non entrino in conflitto con quelli esistenti sfruttando al meglio quelli che sono gli eventi messi a disposizione dal framework SAFe.

12. Migliorare il Portfolio

Per poter incrementare la trasparenza e l'efficienza del portfolio aziendale e massimizzare il valore generato per il cliente mediante delle decisioni sempre più informate, è importante che il portfolio aziendale venga gestito costantemente apportando un miglioramento continuo. Bisogna quindi allineare saldamente e costantemente i flussi di valore con quelli che sono gli obiettivi strategici aziendali, migliorando i processi di allocazione delle risorse economiche tra i diversi flussi. In tal senso, il Lean Portfolio Management (LPM) di SAFe è appositamente progettato per aiutare le aziende a gestire in modo efficace più Value Streams, allineando la strategia con l'esecuzione e i risultati di business. È molto importante il monitoraggio e l'utilizzo di metriche basate sui KPI per valutare i progressi sulla base di dati concreti che influenzano i processi decisionali. Il LPM permette di adattare e ridistribuire le risorse o di modificare gli obiettivi in base alle performance e all'andamento del mercato, consentendo una maggiore flessibilità nell'adattarsi ai cambiamenti esterni.

13. Accelerazione

Benché in questa fase ormai l'implementazione di SAFe ha raggiunto un ottimo livello, è proprio in questa tappa che hanno inizio nuovi percorsi per espandere maggiormente le potenzialità del framework. Infatti, l'obiettivo ora è quello di aumentare l'efficacia e la velocità dei cambiamenti organizzativi, andando a costruire una cultura aziendale incentrata sull'apprendimento e il miglioramento continuo. I team vengono incentivati a sperimentare nuove idee e nuovi processi raccogliendo costantemente i feedback, godendo anche del continuo supporto della leadership. Bisogna coinvolgere attivamente tutti i dipendenti di tutti i livelli dell'organizzazione per creare un ambiente favorevole all'innovazione. Solo così sarà possibile elevare l'azienda a un grado superiore rendendola più reattiva ai

cambiamenti interni ed esterni, grazie anche a una maggiore coesione tra i team e al miglioramento continuo delle tecnologie, dei processi e dei risultati.

Rispetto alla roadmap vista per l'Agile Adoption sicuramente questa è molto più complessa e articolata, così come l'obiettivo finale è molto più ampio e a lungo termine. Si vuole ottenere un vero e proprio cambiamento completo dell'intera azienda verso una nuova metodologia e mentalità. Ancora una volta la formazione e il supporto costante di coach appositamente specializzati è vitale per la buona riuscita della trasformazione e il sostegno durante le tappe più critiche. Non si può pretendere di vedere dei risultati nell'immediato né tanto meno di voler accelerare i tempi prima ancora di aver raggiunto un'esperienza e una maturità tale da essere lanciati verso le fasi successive. È importante che si dedichi il giusto tempo per ogni tappa della roadmap costruendo così una base solida su cui fondare il cambiamento. L'Agile Transformation è quindi un percorso molto complesso che richiede molto impegno e collaborazione da parte di tutti i dipendenti dell'azienda a ogni livello dell'organizzazione, dalla leadership ai vari team di ogni area. I benefici a lungo termine lo rendono però un'ottima scelta strategica per affrontare al meglio le sfide del mercato moderno.

4.2.2 IL “BIG BANG” DI PAYPAL

A conclusione del discorso intrapreso in questo capitolo sull'Agile Transformation, è interessante soffermarsi su un esempio concreto di trasformazione Agile avvenuta con successo: il cosiddetto *Big Bang di Paypal*. L'azienda, fondata nel 1998 con l'obiettivo di offrire una modalità innovativa e sicura per effettuare i pagamenti online, non ha bisogno di molte presentazioni. Quando l'azienda è stata acquistata da eBay nel 2002 stava rapidamente sviluppando delle soluzioni di pagamento innovative portandola a posizionarsi come leader nel settore dei pagamenti digitali. La comparsa sul mercato di concorrenti validi e la nascita di startup richiedeva che i prodotti di Paypal venissero costantemente migliorati e che l'azienda cercasse una costante innovazione. Tuttavia, entro il 2008, l'innovazione ha cominciato a rallentare e il tempo impiegato per il rilascio di nuovi prodotti è aumentato drasticamente, rischiando quindi di non restare al passo con la concorrenza. È in questo contesto che l'azienda ha deciso di intraprendere un cambiamento radicale della propria organizzazione, abbracciando la metodologia Agile e scalandola a livello aziendale. Nel maggio del 2013, Paypal ha avviato una trasformazione Agile che ha chiamato *Big Bang* così come viene descritto in un paper pubblicato da Paypal nel 2015 e intitolato *PayPal Enterprise Transformation*³³.

È curioso vedere nel dettaglio quelli che sono stati identificati come i sei punti critici che insieme concorrevano ai rallentamenti nei rilasci, così come sono stati identificati all'avvio del processo di trasformazione.

- **Documenti con requisiti dettagliati:** per consentire una pianificazione annuale i progetti erano definiti e documentati in appositi documenti chiamati Product Requirements Document (PRD) molto prima dell'avvio dei lavori. Potevano volerci mesi per crearli e spesso risultavano obsoleti prima ancora che iniziassero gli sviluppi.

³³ PayPal, *Paypal Enterprise Transformation*, 2015, Sept 18. Retrieved from [Paypa-objects.com: https://www.paypalobjects.com/webstatic/en_US/mktg/pages/stories/pdf/paypal_transformation_whitepaper_sept_18_2015.pdf](https://www.paypalobjects.com/webstatic/en_US/mktg/pages/stories/pdf/paypal_transformation_whitepaper_sept_18_2015.pdf) (ultimo accesso in data 22/11/2024).

- **Pianificazione trimestrale complessa:** per cercare di dare maggiore flessibilità nella pianificazione, è stato istituito il processo Technology Quarterly Planning (TQP) che ha introdotto però anche più lavoro e complessità a un processo già complicato.
- **Colli di bottiglia nei domini:** per creare competenze più approfondite e migliorare la qualità del codice, Paypal ha suddiviso l'ambiente in 85 domini ognuno dei quali è stato assegnato a sviluppatori specifici con la conseguenza indesiderata di creare dei colli di bottiglia nei domini chiave che erano essenziali per tutti i progetti, come per esempio rischio, compliance e core payment.
- **Cambi di contesto per gli sviluppatori:** per far sì che in un determinato trimestre si avviassero più progetti, il processo di pianificazione spesso assegnava gli sviluppatori a più progetti contemporaneamente con una riduzione della loro produttività dovendo spostarsi da un progetto all'altro perdendo il focus.
- **Sviluppo tradizionale Waterfall:** benché vi fossero dei team che avevano implementato autonomamente alcune iniziative agili, l'azienda utilizzava un modello a cascata per facilitare il coordinamento tra team e domini, ed erano quindi costretti a sottostare comunque alle fasi del modello tradizionale.
- **Cicli di test di integrazione:** ogni rilascio prevedeva test rigorosi che con il passare degli anni diventavano sempre più lunghi da effettuare raggiungendo anche per la più piccola modifica le sei settimane di test di integrazione e di non regressione.

In questo contesto è semplice immaginare che il clima in azienda non fosse dei migliori. Da un lato gli sviluppatori si trovavano coinvolti sempre in più progetti perché mentre alcuni erano fermi venivano già assegnati ad altri senza mai vedere il risultato del proprio lavoro. I Product Manager passavano molto più tempo a documentare ciò che andava realizzato rispetto al reale tempo dedicato all'implementazione e il team di prodotto

invece di dedicarsi ai clienti era spesso impegnato su altre attività interne per cercare di portare a termine i lavori. In generale, era calata notevolmente la fiducia nella capacità dell'organizzazione tecnologica di rispettare le scadenze. Insomma, non è proprio la situazione ideale per effettuare dei rilasci rapidi e costanti che permettano di mantenere alto il livello di competitività dell'azienda sul mercato. Basti pensare a come una startup di giovane creazione possa godere di una struttura organizzativa molto più snella garantendo dei rilasci più frequenti e in linea con le funzionalità in costante aggiornamento nel mondo dei pagamenti online.

È in questo contesto che nel 2012 David Marcus venne promosso a nuovo Presidente e iniziò sin da subito a concentrarsi sull'esperienza del cliente e sulla necessità di innovare i prodotti introducendo un nuovo team di leader per la trasformazione radicale di Paypal. Nello stesso anno Kristen Wolberg si unì a Paypal come responsabile delle Technology Business Operations con anche l'incarico di capire come semplificare il lavoro nell'azienda. K. Wolberg aveva avuto esperienza già in passato con la metodologia Agile e quindi collaborò con l'intera azienda per definire un percorso verso Agile. Grazie alla direzione visionaria di Marcus e al supporto esecutivo della Wolberg in Paypal si intraprese una trasformazione Agile. Viene quindi istituito un team dedicato formato da leader senior di organizzazioni di Prodotto e di Tecnologia per prendere le decisioni chiave alla base della trasformazione. È interessante notare che invece di introdurre gradualmente la nuova metodologia, è stato deciso un cambiamento drastico, un Big Bang appunto. Questo perché vi era una forte interdipendenza tra i team e vi era la necessità di stabilire un framework operativo aziendale per la gestione delle risorse, dei rilasci e della governance.

È a questo punto che si inizia a definire il modo in cui saranno organizzati i nuovi team di lavoro, i quali non saranno più concentrati su un solo dominio specifico, ma saranno cross-funzionali, con le competenze necessarie sia in ambito prodotto sia di dominio tecnico. Si sceglie di adottare il framework Scrum e di fornire l'adeguata formazione sulle pratiche Agile e sulla Customer Driven Innovation ai team, con una formazione aggiuntiva specifica per gli Scrum Master e i Product Owner. Una volta prese le decisioni a livello di team è stato necessario scalare a livello aziendale le pratiche Agile mediante un

adattamento dei framework SAFe e Large Scale Scrum. Come abbiamo visto uno degli aspetti fondamentali per l'Agile Adoption è il supporto di coach Agile durante tutto il periodo di cambiamento. In tal senso, Paypal ha esteso a venti il numero totale di coach per garantire che tutte le linee di prodotto ne avessero almeno uno responsabile del loro successo a seguito del lancio ufficiale della trasformazione avvenuto l'8 maggio 2013. Tutti i progetti che erano già in corso sono terminati con le modalità con cui erano stati avviati, ma da quel giorno tutti i team hanno avviato il loro primo Sprint. In Paypal è stato deciso di pianificare Sprint della durata di due settimane per evitare di farli durare per più tempo rischiando di fare solo con maggiore frequenza tanti piccoli progetti Waterfall. Si è scelto di far ricadere l'inizio e la fine degli Sprint in mezzo alla settimana così da dare la possibilità a tutti di essere presenti in quelle occasioni e ogni team è sincronizzato con gli altri.

Un altro aspetto molto importante riguarda la scelta dei tool. Siccome alcuni team già utilizzavano diversi strumenti sia fisici come delle lavagne, sia software, è stato più semplice avere dei feedback su quali strumenti avrebbero permesso di pianificare, gestire e rendicontare al meglio il lavoro senza appesantire l'attività dei team e automatizzando il più possibile. Ciò che è bene notare è che con più di 300 team è stato fondamentale definire un tool a livello aziendale, che fosse lo stesso per tutti, in modo tale anche da favorire la gestione delle interdipendenze tra i team. In più, tutti i dipendenti sono stati formati appositamente sull'utilizzo del tool e hanno ricevuto supporto dedicato, aspetto molto importante da un lato per imparare a usare il tool, dall'altro per standardizzare e uniformare il modo in cui l'azienda ha stabilito che fosse meglio utilizzarlo.

Per quanto riguarda il controllo costante dell'andamento della trasformazione, in Paypal è stata prevista un'apposita dashboard di monitoraggio e una serie di metriche per comprendere quanto efficacemente i team stessero adottando le nuove pratiche e per garantire che i manager in tutta l'azienda stessero formando e gestendo i team in modo da massimizzare il successo. Inoltre, i team dovevano completare delle valutazioni periodiche in modo tale che tutti questi dati messi insieme formassero un punteggio che desse a colpo d'occhio alla leadership una visione sullo stato e sui possibili ostacoli o sulle pratiche inefficienti.

Il processo di trasformazione Agile ha portato Paypal a essere completamente diversa e già nel 2014 era possibile vedere i benefici:

- **Focus sulle roadmap di prodotto:** grazie a una pianificazione strutturata i leader si potevano concentrare su obiettivi strategici.
- **Sostituzione dei *Dev Days* con i *Team Sprints*:** per la stima e la gestione del lavoro invece di considerare i giorni di lavoro si considera il numero di Sprint.
- **Maggiore stabilità dei team:** collaborazione tra i membri del team e rapida adattabilità ai cambiamenti delle priorità del prodotto.
- **Utilizzo di pratiche Agile Scrum:** la pianificazione e il completamento del lavoro sono basati sul backlog di prodotto e non più su lunghissimi documenti funzionali.
- **Integrazione dei feedback dei clienti:** le idee raccolte dai clienti possono essere introdotte in soluzioni iterative e dinamiche durante gli sviluppi, non facendo più riferimento a documenti statici.
- **Rilasci più rapidi:** il codice viene rilasciato più rapidamente grazie agli Sprint di due settimane.

Non bisogna però immaginare che tutto questo sia stato di semplice implementazione o che non vi siano state situazioni di resistenza al cambiamento. Per esempio, nel paper si legge di alcune difficoltà con alcuni team che avevano già adottato autonomamente alcune pratiche Agile e che quindi hanno impiegato più tempo per apprezzare il valore di uno standard aziendale in quanto abituati da sempre a utilizzare il proprio modello, non più accettato.

Il percorso intrapreso da Paypal a partire dal 2012 ha fatto sì che l'azienda sia ancora oggi leader nel settore dei pagamenti digitali e sia riconosciuta come un esempio di successo dell'Agile Transformation. La trasformazione di Paypal è un esempio chiaro di come un'azienda che si trova in una situazione di stallo e in cui vi sono una molteplicità di criticità possa procedere a un'attenta riorganizzazione e affermare la sua presenza e competitività sul mercato grazie all'adozione di pratiche Agile su scala aziendale.

4.3 DALL'ADOPTION ALLA TRANSFORMATION IN SANTANDER CONSUMER BANK

Intraprendere un percorso di cambiamento, evolvendo da una struttura tradizionale verso un modello più flessibile e integrato, capace di rispondere con maggiore rapidità ed efficienza alle esigenze del mercato e agli interessi dei clienti, è un momento cruciale per qualsiasi azienda. Vi sono sicuramente molti segnali che possono far capire che effettivamente un cambiamento sarebbe qualcosa di benefico per l'intera organizzazione, ma non è facile prendere una tale decisione ed essere pronti a sostenere un periodo delicato di transizione. È in questo contesto che Santander Consumer Bank Italia si trova di fronte a una grande sfida che da un lato può essere preoccupante, ma dall'altro potrebbe apportare notevoli benefici, anche a lungo termine. Dall'esperienza di tirocinio sono emerse numerose lessons learned che mettono in evidenza come un approccio tradizionale presenti alcune criticità non trascurabile e allo stesso tempo come l'introduzione di una mentalità e una metodologia Agile e lean porterebbe con sé non poche opportunità. Data la complessa struttura organizzativa e la grande quantità di prodotti è chiaro che un cambiamento radicale necessiterebbe di un grande lavoro da parte di tutta l'azienda e molto probabilmente un approccio graduale permetterebbe di affinare al meglio gli strumenti sulle reali necessità, introducendo ed espandendo in tutte le aree aziendali una nuova mentalità di crescita, orientata al cambiamento e al miglioramento continuo. Proprio parlando di mentalità abbiamo evidenziato come non tutte le persone approcciano i cambiamenti e le sfide quotidiane allo stesso modo; perciò, non tutti reagirebbero ugualmente e unitamente a favore del cambiamento. Sicuramente, muoversi gradualmente e con un gruppo di persone di base più aperte a esplorare nuovi orizzonti getta le basi verso un'espansione scalata a livello aziendale di nuove pratiche e metodologie. Se da un lato l'Agile Adoption, che, come si è visto nei capitoli precedenti, si concentra maggiormente sul cambiamento nel piccolo, a livello di singolo team o di specifico progetto, può essere un ottimo punto di partenza per un cambiamento più ampio, è con l'Agile Transformation che si raggiunge il punto di arrivo e che si integrano i principi e i valori lean e Agile in ogni area e livello aziendale. Si tratta di un cambiamento strategico

che necessita di una forte leadership e di una pianificazione accurata che coinvolga tutti i dipendenti aziendali, ma anche tutti i fornitori esterni. Come si è visto per il caso Paypal una trasformazione nasce sempre da una consapevolezza chiara dello stato attuale e una rilevazione accurata delle criticità che devono fungere da carburante del cambiamento, diventando delle opportunità di crescita e di miglioramento.

L'avvio di una trasformazione Agile di successo deve prendere in considerazione sia interventi immediati che affrontino quelle che sono le criticità emerse nel breve termine, sia una strategia più articolata e strutturata con una visione a lungo termine. Gli interventi mirati e l'adozione di alcune best practices possono apportare benefici tangibili che siano un concreto esempio per il futuro dell'intera azienda. In tal senso, l'introduzione di alcuni consigli basati sulle evidenze apprese durante il tirocinio permetterebbe a Santander Consumer Bank di godere di alcuni benefici sin da subito, senza dover necessariamente avviare una trasformazione radicale repentina. Gli interventi mirati possono migliorare la gestione operativa e favorire un approccio più efficiente anche senza stravolgere l'intera struttura tradizionale in quanto agiscono sulle dinamiche attuali come, per esempio, la gestione dei fornitori esterni e l'importanza di un maggior coinvolgimento degli utenti durante l'intero progetto. Implementando correttamente le best practices è possibile ottenere un significativo miglioramento della qualità dei prodotti e nella riduzione dei tempi di consegna di nuove funzionalità in linea con le richieste iniziali.



Figura 17 - Azioni suggerite sulla base delle lessons learned in Santander Consumer Bank Italia.

In Figura 17 e di seguito sono descritti alcuni suggerimenti tattici forniti come azioni da implementare sulla base delle lessons learned analizzate durante l'esperienza di tesi in azienda e di tirocinio da Product Owner.

INCENTIVARE LA COMUNICAZIONE TRA I FORNITORI

Lesson learned: il gran numero di fornitori coinvolti genera una frammentazione e la mancanza di un coordinamento efficace rallenta il processo di integrazione dei moduli sviluppati da ognuno aumentando i ritardi nei SIT.

Action: è molto importante che il Product Owner tra le sue attività incentivi il dialogo con i fornitori e soprattutto tra i fornitori stessi. È bene che tutti dispongano delle informazioni necessarie e che si muovano autonomamente durante gli sviluppi per garantire che i propri singoli moduli del sistema funzionino correttamente una volta integrati. In questo modo sarà possibile ridurre le modifiche richieste durante i SIT che causano rallentamenti e rischiano di far posticipare gli UAT e conseguentemente la data di go-live. Sarebbe bene che i fornitori partecipassero a meeting periodici, anche convocati e gestiti autonomamente, in modo tale da poter discutere delle problematiche, dei progressi e dei miglioramenti che sono alla fine comuni a tutti. In questo modo, non solo si possono ridurre i disallineamenti, ma si riducono anche i tempi di risoluzione di una problematica grazie a un confronto diretto e immediato tra tutti i fornitori impattati.

COINVOLGERE GLI UTENTI DURANTE GLI SVILUPPI

Lesson learned: durante gli UAT emergono spesso delle richieste di modifiche da parte degli utenti, segno di un disallineamento tra quanto richiesto e quanto realmente sviluppato.

Action: è bene che gli utenti siano coinvolti durante l'intero ciclo di vita del progetto. Sarebbe quindi fondamentale che non si limitasse il loro coinvolgimento alle sole fasi di raccolta dei requisiti e di validazione finale. Si potrebbe, invece, prevedere degli incontri

regolari in cui gli utenti possano già verificare alcune implementazioni parziali e fornire dei feedback che guidino gli sviluppatori durante l'implementazione delle funzionalità richieste. Non si tratta di incontri per conoscere in generale lo stato di avanzamento dei lavori, ma un vero e proprio incontro di condivisione di dubbi, perplessità o di richieste in un dialogo prezioso tra i fornitori e l'utente. Il Product Owner si occuperebbe di convocare le riunioni e di presidiare assicurandosi che gli sviluppatori stiano comprendendo le richieste degli utenti e che non vi siano delle discrepanze tra i requisiti previsti all'inizio del progetto e le richieste formulate a seguito di questi incontri. Per esempio, si potrebbero pianificare delle review ogni settimana o ogni due settimane, in base alla durata della fase degli sviluppi, in cui eventualmente il fornitore mostri delle demo di anche solo minime parti implementate che consentano di identificare ancora prima della fase degli UAT possibili disallineamenti, prima ancora che diventino problematiche più grandi. In questo modo si riduce il rischio di incorrere in modifiche costose e lunghe in una fase intermedia del progetto e non nelle fasi avanzate e più delicate. In più, questo rapporto costante con gli utenti aiuta ad accrescere in loro la fiducia verso i fornitori e di conseguenza potrebbero essere più propensi ad accettare compromessi qualora alcune funzionalità non potranno essere pienamente in linea con le richieste.

STANDARDIZZARE

Lesson learned: spesso i problemi di integrazione derivano dal modo in cui vengono gestiti i dati tra i vari fornitori che implementano i moduli del sistema, risultando in formati incompatibili o alle volte totalmente assenti.

Action: standardizzare a livello aziendale il formato dei dati e le modalità di stesura dei sequence in modo tale che si riducano le incertezze e la possibilità di introdurre disallineamenti. La definizione di standard permetterebbe inoltre di fornire delle linee guida univoche per ogni fornitore sulla gestione degli errori e dei relativi messaggi d'errore in modo tale che gli interventi in risoluzione delle segnalazioni siano più puntuali e tempestivi. Potrebbe essere utile delineare un approccio comune e verificare che venga rispettato da tutti utilizzando strumenti come OpenAPI e Swagger per la definizione e la

documentazione delle API. Qualora vi fossero dei moduli e dei servizi che non sono allineati al nuovo standard è opportuno prevedere un piano di riallineamento così che si potrà beneficiare dei vantaggi della standardizzazione anche su quanto è già stato sviluppato in passato, integrandosi alla perfezione con le future implementazioni. Ciò non solo migliora l'efficacia degli sviluppi e riduce le tempistiche dei SIT, ma fa sì che eventuali disallineamenti emergano già durante le analisi tecniche perché è più facile rilevare già a monte eventuali discrepanze con gli standard previsti. Inoltre, si consiglia come detto precedentemente, un dialogo costante tra i fornitori già durante gli sviluppi per verificare la correttezza dei sequence e del flusso dei dati ancora prima di procedere all'integrazione completa che introduce molte più possibilità di errore rendendo più complessa l'investigazione. Invece, se i singoli moduli venissero testati a uno a uno le variabili in gioco sarebbero limitate e di più facile analisi e risoluzione. Solo successivamente si potrà verificare nei SIT che l'intera struttura sia robusta e funzioni unitamente senza errori. Quindi oltre a standardizzare è bene anche avere un approccio orientato ai test sin dai primi sviluppi, in un dialogo costante tra tutti i fornitori che dovrebbero allineare il proprio piano di lavoro in modo tale da prevedere dei checkpoint intermedi gestiti autonomamente.

COMUNICARE REQUISITI COMPLETI E CHIARI

Lesson learned: passando molto tempo tra la definizione dei requisiti iniziali, la loro implementazione e la validazione, spesso non risultano sufficientemente chiari e i documenti delle analisi funzionali non sono aggiornati.

Action: siccome per la raccolta dei requisiti sono già presenti dei workshop dedicati, in tal senso sarebbe opportuno pensare a degli strumenti e delle modalità che possano renderli più efficaci. Aumentare il focus nella fase di definizione dei requisiti iniziali permetterebbe di prevenire fraintendimenti e spiacevoli disallineamenti. Durante i workshop si potrebbero utilizzare delle tecniche di brainstorming e di visualizzazione delle idee così da renderle più chiare e comprensibili. È fondamentale che sin da subito non vengano coinvolti solo gli utenti, ma anche i fornitori così che entrino più concretamente

nell'ottica di quello che andrà implementato, ragionando sulle tecnologie coinvolte e sulle possibili soluzioni da adottare per essere innovativi e al passo con l'evoluzione tecnologica. Tutte le persone coinvolte e che dovranno prendere parte ai workshop sono invitate a partecipare attivamente, proponendo le proprie idee, discutendo sui requisiti richiesti ed esponendo le proprie perplessità in un clima sereno. In questo modo, si è maggiormente concentrati e l'unione di tutti fa sì che nulla sia lasciato al caso e che si crei una base solida approvata all'unanimità dopo ogni incontro. È bene che i fornitori preparino dei prototipi che durante i workshop permettano di avere una visione più chiara sia sulle richieste. Inoltre, è opportuno introdurre un sistema che permetta la tracciabilità per ogni requisito, collegandolo direttamente alla sua implementazione tecnica e ai relativi test, consentendo un opportuno monitoraggio durante tutte le fasi, utile ad identificare rapidamente eventuali mancanze o modifiche.

FORMAZIONE DEDICATA

Lesson learned: a volte la comunicazione è scarsa e non sempre sono chiari i ruoli e le responsabilità delle diverse figure coinvolte in una iniziativa, sia perché non sempre sono coinvolti tutti gli attori (es. a volte le iniziative sono gestite dal Product Owner senza un Project Manager che come abbiamo visto sono due figure distinte con responsabilità diverse) sia perché il panorama delle iniziative è molto variegato e articolato.

Action: è importante che prima di avviare una nuova iniziativa, si svolga un incontro in cui tutte le risorse necessarie si conoscano e prendano consapevolezza delle rispettive responsabilità, indipendentemente dal grado di esperienza pregressa. Infatti, è opportuno che ogni volta si analizzi il caso specifico e si delinei il campo di azione di ognuno. Questo permette di stabilire una base comune per la comunicazione evitando sovrapposizioni o lacune operative. Anche se alcune figure, come per esempio il Product Owner, ricoprono spesso responsabilità simili in ogni progetto, è utile ribadire tali aspetti, sottolineando quale sarà il contributo di ognuno nelle diverse fasi del progetto. In tal senso, l'introduzione di workshop dedicati per i vari ruoli, che forniscano una visione dettagliata delle responsabilità e che ribadiscano i punti fondamentali del lavoro da svolgere

ricoprendo quel determinato ruolo, aiuta a mantenere alto il livello di professionalità di tutti. La formazione costante garantisce che vi sia anche un aggiornamento continuo delle metodologie applicate e che si conoscano anche nuovi strumenti che potrebbero essere utili da adottare iniziativa dopo iniziativa. Se da un lato si possono prevedere dei corsi di formazione con soggetti esterni specializzati, dall'altro è bene considerare anche dei momenti di condivisione tra persone che ricoprono lo stesso ruolo in diverse progettualità. Esporre le proprie perplessità e venire a conoscenza delle sfide affrontate dai colleghi nello stesso ambito permette di scoprire altri punti di vista e altre pratiche che potrebbero essere un valore aggiunto e prese da esempio per migliorare il lavoro svolto quotidianamente. Inoltre, è bene promuovere iniziative che rafforzino la collaborazione. Non trovandosi a lavorare sempre con le stesse persone, creare un gruppo coeso può essere facilitato tramite dei momenti di team building che migliorano l'allineamento e favoriscono una comunicazione efficace durante tutte le fasi del progetto.

AGILE ADOPTION

Lesson learned: i progetti gestiti con il modello Waterfall incontrano spesso dei ritardi e provocano molteplici criticità dovute alla rigidità delle diverse fasi. È difficile rispettare la pianificazione iniziale e garantire che si raggiungano le date di go-live senza che vi siano ancora delle segnalazioni da risolvere. Alcuni progetti, per loro natura, potrebbero non essere adatti per essere gestiti con metodologie tradizionali e si rischia di creare delle forzature e delle inefficienze cercando di incastrarli in un modello meno affine.

Action: in questo caso vale la pena valutare la possibilità di adottare la metodologia Agile. Sicuramente esistono alcune iniziative che possono essere gestite con il modello Waterfall, ma in ambito dei progetti IT il modello Agile si presta molto più efficacemente. In tal senso, sarebbe opportuno introdurre un processo di valutazione iniziale per identificare la presenza di progetti che potenzialmente potrebbero essere gestiti con una metodologia Agile, con particolare attenzione per quei progetti più complessi e che generalmente dall'esperienza pregressa presentano un'alta variabilità dei requisiti. Il suggerimento è quindi quello di avviare una adozione Agile che permetta di introdurre per la

prima volta un nuovo approccio di gestione delle progettualità dell'azienda. Di seguito si indicano alcuni punti chiave per facilitare l'Agile Adoption:

- **Progetto pilota:** identificare e selezionare un progetto pilota che per le sue caratteristiche risulta più adatto alle logiche e in linea con i principi di Agile. È molto importante che lo sponsor del progetto sia disposto a sperimentare le opportunità e motivato al cambiamento. Si consiglia un progetto che sia contenuto nella complessità e nelle attività in modo tale che si possa famigliarizzare gradualmente con la metodologia Agile. Prima ancora di avviare il progetto pilota, è bene selezionare una serie di strumenti da utilizzare come il framework Scrum e Kanban così da sperimentare diverse soluzioni e lasciare che sia il team a comprendere la migliore nel mentre che la si utilizza.
- **Team pilota:** formare un team di persone motivate e aperte al cambiamento, disposte a sperimentare la nuova metodologia e a mettere in discussione le proprie convinzioni per gestire con un nuovo approccio il progetto pilota. Se si sceglie di adottare il framework Scrum, prevedere almeno un Product Owner e uno Scrum Master interni all'azienda. Per quanto riguarda i Developers, invece, lavorando solo con fornitori esterni è importante che siano loro stessi a identificare alcune persone disposte ad affrontare la sfida entrando a far parte del team pilota.
- **Predisposizione ambiente:** vista la situazione nuova e particolare, è prima di tutto importante predisporre un ambiente di lavoro dedicato nel quale il team pilota possa lavorare comodamente e possa sfruttare gli spazi per favorire la collaborazione e la comunicazione trasparente. Sarebbe quindi consigliato che anche gli sviluppatori dei diversi fornitori siano frequentemente presenti direttamente nella sede aziendale a stretto contatto con i membri interni del team. Anche la predisposizione di un sistema di videoconferenza opportunamente

progettato potrebbe facilitare il lavoro di squadra anche quando la presenza di tutti non può essere garantita.

- **Formazione e coaching:** fornire a tutti i membri del team la corretta formazione iniziale che non sia una semplice introduzione al mondo dell'Agile e agli strumenti, ma che siano una serie di sessioni in cui è richiesto un coinvolgimento pratico e in cui si facciano delle simulazioni guidate di ciò che potrebbero dover gestire durante lo svolgimento del progetto pilota. È importante anche il supporto costante durante tutto il progetto grazie a coach esterni che chiariranno dubbi e perplessità che emergeranno durante lo svolgimento delle attività e che formeranno uno dei membri che diventerà un esperto interno di Agile, punto di riferimento per il team e per le iniziative future.
- **Pianificazione iterativa:** è importante che si introducano dei cicli di lavoro brevi, in particolare gli Sprint se si adotta Scrum, per garantire una maggiore adattabilità ai cambiamenti e un continuo miglioramento grazie alla continua verifica del progresso e alla valutazione del lavoro svolto. È importante che tutti gli eventi previsti da Scrum vengano rispettati con particolare attenzione a tutti i punti di contatto con gli utenti per garantire il dialogo costante e la trasparenza. I cicli di lavoro brevi permetteranno di effettuare test più frequenti e più puntuali aumentando la qualità del software rilasciato e raggiungendo le aspettative degli utenti e dei clienti finali. Sarà opportuno disporre di tool adeguati alla realizzazione del backlog e alla sua gestione.
- **Monitoraggio continuo:** bisogna considerare che l'Agile Adoption è fondamentale per comprendere quali sono i vantaggi e le sfide nell'adozione della metodologia Agile in azienda, anche solo come punto di inizio per poi valutare l'avvio di una vera e propria Agile Transformation. Quindi è fondamentale che vengano implementati degli strumenti di monitoraggio per valutare l'efficacia del nuovo approccio e identificare che il team stia seguendo le corrette linee guida e che

l'adozione stia avvenendo con successo. Il monitoraggio continuo permettere di rilevare aree di miglioramento e di utilizzare le lessons learned per affinare il lavoro in corso.

- **Feedback a livello aziendale:** una volta terminata l'esperienza di adozione Agile per il progetto pilota, bisognerà effettuare degli incontri dedicati per una retrospettiva a più ampio raggio ripercorrendo tutte le difficoltà incontrate e le diverse azioni intraprese. In questo modo si potranno tirare le somme della sperimentazione e preparare una presentazione conclusiva di tutto il percorso. Sarà molto importante prevedere un incontro aziendale in cui esporre l'esperienza del team pilota e dei risultati ottenuti, promuovendo in questo modo la metodologia Agile verso sempre più aree della banca. In questa occasione si potrebbe richiedere ai partecipanti di esprimere una propria preferenza sull'essere coinvolti o meno in successivi progetti che si svolgeranno con metodologia Agile, identificando i membri di nuovi team pilota cercando di scalare sempre di più all'intera organizzazione la nuova metodologia, abbandonando progressivamente i modelli tradizionali.

Quelle presentate finora sono delle azioni concrete e immediate che mirano a migliorare le criticità riscontrate durante l'esperienza di tesi in azienda e di tirocinio. Queste soluzioni possono apportare benefici tangibili nel breve termine, ma è comunque molto importante considerare che un reale cambiamento strutturale e culturale sarebbe più strategico in quanto apporterebbe dei vantaggi anche a lungo termine. L'Agile Transformation, infatti, non si limita alla sola risoluzione delle singole criticità esistenti, ma è revisione completa della mentalità, dell'intero modello organizzativo e dei processi aziendali. Nel contesto di Santander Consumer Bank un cambiamento di questo tipo rappresenta un'opportunità unica e che consentirebbe di integrare a tutti i livelli le metodologie e i valori Agile e lean, gettando le basi a un'organizzazione più flessibile e innovativa, orientata al miglioramento continuo e attenta alle esigenze in continuo cambiamento dei clienti finali. Come abbiamo visto il processo di trasformazione è molto lungo e

complesso, richiede un grande coinvolgimento da parte di tutte le aree aziendali e sono quindi necessarie alcune linee guida che consentano un cambiamento di successo.



Figura 18 – Alcuni elementi chiave per una trasformazione Agile in Santander Consumer Bank Italia.

In Figura 18 sono riportati alcuni elementi chiave per una trasformazione Agile di successo in Santander Consumer Bank Italia, così come dettagliato di seguito.

SERVANT LEADERSHIP

La struttura organizzativa di Santander Consumer Bank Italia è articolata e basata su processi tradizionali. Uno dei primi passi per avviare una trasformazione Agile è quello di avere un forte supporto dalla leadership. Sono i leader *in primis* che devono essere promotori di una nuova cultura aziendale, mostrando il loro impegno verso i valori della trasparenza, della collaborazione e del miglioramento continuo. Si parla di *servant leadership* proprio perché i leader sono chiamati a non concentrarsi sul potere o sul controllo, ma sul servizio dato agli altri come loro principale obiettivo. È bene quindi che pongano al centro quelle che sono le esigenze e la crescita delle persone all'interno dell'azienda, costruendo un ambiente adeguato in cui si possa dialogare liberamente promuovendo l'empatia e l'ascolto attivo. I leader fungono da motivatori e sostengono il benessere di tutti i dipendenti dell'azienda, supportandoli e aiutandoli nella crescita personale e professionale. Questo è un aspetto molto importante perché getta delle solide basi e crea un ambiente di lavoro più armonioso e di conseguenza più produttivo. Non è

semplice entrare in quest'ottica soprattutto se per anni si è applicato un approccio più tradizionale, ma vi sono alcuni suggerimenti che se applicati possono agevolare la transizione verso una servant leadership. La leadership dovrebbe:

- dedicare del tempo prestando attenzione a quelli che sono i bisogni, le preoccupazioni e le idee del team,
- incoraggiare e accettare i feedback e le opinioni condivise creando uno spazio preposto,
- coltivare l'empatia comprendendo le prospettive degli altri e interessandosi al loro reale benessere,
- dimostrare fiducia verso i collaboratori, delegando le responsabilità in modo strategico, assegnando attività che consentano una loro crescita personale e professionale,
- ottenere la fiducia dai collaboratori agendo con onestà, trasparenza e coerenza,
- verificare che l'ambiente di lavoro metta a proprio agio tutti i dipendenti, assicurandosi che tutti siano rispettati e ascoltati,
- dare la propria disponibilità a supportare e aiutare i collaboratori quando si trovano in difficoltà nell'affrontare le sfide e nel raggiungere gli obiettivi.

In tal senso risulta molto importante organizzare dei workshop appositi per il top management che, oltre a introdurre e illustrare i benefici dell'Agile e del lean thinking, formino i manager al ruolo di servant leader. I leader sono anche i diretti responsabili di definire una vision chiara e condivisa sulla trasformazione che avverrà comunicando efficacemente gli obiettivi e i vantaggi di cui beneficerà l'intera azienda.

CENTRO DI ECCELLENZA LEAN E AGILE

Dato che la trasformazione richiede un grande sforzo da parte di tutta l'azienda, è consigliabile formare un gruppo specifico che si concentri sulla strategia di cambiamento e sul coordinamento delle diverse azioni da compiere. Il team dovrà disporre di competenze multidisciplinari e di un chiaro mandato per guidare la trasformazione, sostenendo l'adozione dei principi lean e Agile in tutta l'azienda. Si tratta quindi di un punto di riferimento

sia per la definizione delle strategie, sia per la selezione degli strumenti e delle pratiche da utilizzare. Vista la complessità della realtà di Santander Consumer Bank Italia, è consigliato formare un gruppo che non includa solo le persone interne provenienti dalle diverse direzioni e aree, ma anche di professionisti esterni, esperti delle metodologie Agile e degli strumenti messi a disposizione dai framework. SAFe prevede diversi corsi e certificazioni che sarebbero molto preziosi per tutti i componenti del team, in linea con l'importanza data alla formazione costante che, come si è già detto, è sempre molto importante. Il Centro di eccellenza Lean e Agile si occuperà di:

- definire la strategia di trasformazione in linea con gli obiettivi aziendali e sulla base delle evidenze emerse da un'analisi attenta delle criticità aziendali,
- definire le linee guida e scegliere gli strumenti Agile più opportuni che dovranno essere adottati dall'intera azienda,
- coordinare le diverse azioni e iniziative di cambiamento, evitando che si creino conflitti e disallineamenti,
- dare supporto ai diversi team aziendali nelle diverse direzioni e aree così da aiutarli ogni qual volta emergano dei dubbi o delle perplessità,
- promuovere la cultura Agile all'interno dell'azienda,
- monitorare il progresso della trasformazione basandosi su metriche chiare,
- rimuovere eventuali ostacoli identificati che impediscono il progredire pianificato del cambiamento.

Il Centro di eccellenza Lean e Agile sarà alla base della trasformazione Agile, garantendo che il cambiamento avvenga senza ostacoli mediante una transizione efficace e sostenibile verso un modello più flessibile e innovativo. In tal senso, la roadmap dell'implementazione di SAFe è sicuramente uno strumento molto utile a cui aderire o anche solo ispirarsi per adattarla alle necessità dell'azienda.

PROGETTO PILOTA

Vista la complessità della struttura presentata nel sottocapitolo 3.1 di Santander Consumer Bank Italia, sarebbe meglio procedere in modo graduale verso una trasformazione

Agile completa. Quanto detto precedentemente per le lessons learned e l'Agile Adoption trova il suo significato anche in questo contesto. Un approccio incrementale, che all'avvio identifica alcuni progetti pilota che impattano solo alcune aree dell'azienda, consente di identificare sin da subito eventuali criticità che si riscontrerebbero anche successivamente scalando il nuovo modello a tutti i livelli dell'azienda. In questo modo diventa più semplice iniziare a sperimentare e a monitorare la risposta da parte dei team all'utilizzo di nuove pratiche. In più, circoscrivere l'introduzione di Agile a piccoli team e progetti pilota riduce il rischio di una trasformazione, consentendo di fallire o di commettere errori senza che questi abbiano impatto sull'intera organizzazione con conseguenze disastrose. Solo successivamente, una volta affinata la tecnica si potrà allargare progressivamente il perimetro della trasformazione senza esporre l'intera azienda a un rischio troppo elevato. Man mano che le nuove pratiche applicate nei progetti pilota mostreranno i propri risultati e benefici, le persone coinvolte si faranno promotrici con i colleghi, motivando le altre aree aziendali e promuovendo la cultura Agile. In questa fase di Agile Adoption il supporto del Centro d'eccellenza Lean e Agile è essenziale sia per la formazione che per il coaching continuo garantendo che gli standard definiti e gli strumenti selezionati siano compresi e applicati correttamente. È bene quindi prevedere dei cicli di lavoro brevi che consentono revisioni periodiche frequenti alle quali sarebbe utile invitare anche alcuni colleghi non coinvolti nel progetto pilota per iniziare a scoprire il nuovo modo di lavorare, favorendo l'apprendimento collettivo e l'engagement dell'intera organizzazione. Per garantire un avvio graduale basato su progetto pilota si consiglia di:

- identificare un'area a basso rischio,
- selezionare pochi progetti di dimensioni ridotte e molto specifici, con impatti trasversali ridotti,
- formare un team di persone motivate e favorevoli al cambiamento, con conoscenze dei principi Agile e lean e che si appassionano al proprio lavoro,
- applicare un framework come Scrum e utilizzare Kanban per organizzare il lavoro del team e monitorare il progresso dei progetti pilota,
- comunicare gli avanzamenti e le evidenze emerse con l'intera organizzazione,

- usare le lessons learned apprese dai progetti pilota per migliorare l'implementazione progressiva verso le altre aree aziendali.

Un approccio pilota consentirebbe quindi di muoversi con cautela e di imparare dall'esperienza sviluppando competenze fondamentali che saranno preziose per supportare l'adozione su larga scala.

NUOVA STRUTTURA ORGANIZZATIVA

Dopo aver sperimentato in un perimetro ristretto, è però importante prevedere una riorganizzazione a livello aziendale. Uno dei punti critici della struttura attuale di Santander Consumer Bank Italia è la suddivisione in silos funzionali che rallenta notevolmente il flusso di lavoro, limitando la collaborazione e creando dei colli di bottiglia indesiderati. Per questo motivo è importante riorganizzare le aree in modo tale che si creino dei team cross-funzionali non più suddivisi per funzionalità, ma costruiti attorno ai prodotti e quindi a quelli che abbiamo definito i flussi di valore. Il Centro d'eccellenza Lean e Agile sarà responsabile di identificare tutti gli Stream Values dell'azienda e di prevedere le risorse necessarie per creare i nuovi team trasversali a tutte le aree e direzioni attualmente presenti, favorendo l'efficienza operativa e un approccio orientato ai risultati. La nuova struttura dell'organizzazione dovrà essere formata da team eterogenei con competenze diverse che consentano di gestire autonomamente l'intero ciclo di vita di un prodotto senza dipendere da altri reparti. È importante che i team siano costruiti con particolare attenzione ai processi che portano al cliente finale uno specifico prodotto o un servizio dell'azienda, facendo particolare attenzione a identificare i flussi principali e fondamentali per garantire un'elevata competitività sul mercato. Ad ogni team dovranno essere assegnati degli obiettivi chiari e specifici del proprio flusso di valore con indicatori di performance specifici. Il management dovrà comunicare la vision e la mission aziendale in modo trasparente ed efficace in modo tale da garantire che tutti i team siano allineati e lavorino insieme per raggiungere gli obiettivi comuni a lungo termine, anche grazie all'utilizzo degli eventi SAFe pensati appositamente per garantire l'allineamento tra team. Effettuare una riorganizzazione della struttura complessa dell'azienda richiede un grande sforzo, ma porta a grandi vantaggi come:

- la riduzione delle dipendenze tra le diverse aree e i passaggi tra team, consentendo tempi di consegna più rapidi e la riduzione dei ritardi,
- l'aumento della collaborazione tra team e del senso di appartenenza e di valore nell'azienda,
- una maggiore attenzione al cliente finale, organizzandosi attorno ai flussi di valore in modo tale che si apporti un reale miglioramento dell'esperienza del cliente,
- una risposta più rapida ai cambiamenti nelle priorità aziendali e nelle normative o alle esigenze del mercato.

L'elemento atomico della nuova struttura sono i team cross-funzionali che permettono di rimuovere le inefficienze della struttura rigida a silos funzionali e promuove un ambiente di lavoro maggiormente collaborativo, dinamico e orientato al valore per il cliente finale.

CONTRATTI AGILE

Un altro aspetto critico di Santander Consumer Bank Italia riguarda il gran numero di fornitori esterni coinvolti sulle diverse progettualità. È fondamentale quindi che, per supportare efficacemente la trasformazione Agile, venga rivista la gestione dei fornitori. Bisogna prevedere di coinvolgerli attivamente nel processo di trasformazione e questo mediante l'adozione dei contratti Agile che superano i limiti dei tradizionali contratti a prezzo fisso in favore di una maggiore flessibilità e collaborazione che apporti un vantaggio a entrambe le parti. In questo modo i fornitori diventano dei veri e propri partner che lavorano insieme al team interno per raggiungere gli obiettivi comuni. Il successo di Santander permette anche il successo dei fornitori e con questa modalità di contratto i ritardi e le problematiche bloccanti avranno un peso non solo sull'azienda, ma anche su tutti i fornitori, incentivando un lavoro di qualità e la massima professionalità e serietà. Grazie a meccanismi per revisioni frequenti e pianificazioni iterative è possibile misurare costantemente l'avanzamento dei lavori incentivando un miglioramento continuo e la qualità attraverso metriche di performance concordate. Tra le diverse strutture contrattuali Agile, si consiglia il contratto SAFe Managed-Investment per le motivazioni

affrontate nel sottocapitolo 3.5. Tra i vantaggi della sottoscrizione di contratti Agile per la gestione dei fornitori troviamo:

- la possibilità di rispondere rapidamente ai cambiamenti del mercato e delle strategie aziendali,
- la riduzione del rischio di implementare funzionalità non in linea con le richieste iniziali e di introdurre ritardi, accorgendosene in tempo grazie alle revisioni frequenti,
- una maggiore integrazione dei fornitori con i team interni avendo un obiettivo comune più marcato e allineato,
- un clima più sereno nei rapporti committente-fornitore minimizzando le controversie legate a specifiche contrattuali rigide e inadeguate.

Essendo quindi le collaborazioni esterne cruciali per tutte le progettualità di Santander Consumer Bank Italia, l'adozione di contratti Agile migliorerebbe significativamente la qualità delle collaborazioni esterne, disponendo di partner reattivi e capaci di sostenere efficacemente la trasformazione Agile.

MONITORAGGIO

Infine, bisogna ricordare che la trasformazione Agile è un processo continuo e che richiede un monitoraggio costante per effettuare costantemente degli adattamenti durante tutto il percorso di cambiamento. In altre parole, la trasformazione Agile in Santander Consumer Bank Italia dovrà essere gestita come un progetto a tutti gli effetti, applicando i principi Agile e lean con un approccio iterativo e dando molta importanza alla revisione periodica e frequente degli avanzamenti, rilevando punti di forza, debolezze e opportunità. È vero che il Centro d'eccellenza Lean e Agile adotterà diverse pratiche per pianificare al meglio l'avvio della trasformazione, ma sarà fondamentale che venga fatto un monitoraggio costante in corso d'opera a tutti i livelli riorganizzati della nuova struttura aziendale articolata in team costruiti intorno ai flussi di valore. Bisognerà quindi pianificare degli incontri regolari per esaminare i progressi e identificare gli ostacoli che andranno rimossi e prevedere un adattamento delle strategie definiti prima dell'avvio. Un monitoraggio efficace si avvale di metriche di performance specifiche come il time-to-

market, la qualità degli sviluppi, il grado di collaborazione e la soddisfazione dei clienti interni ed esterni. Si consiglia la creazione di una dashboard apposita che permetta a colpo d'occhio di avere un'immagine sullo stato della trasformazione e sui benefici che sta apportando, rendendo i dati di monitoraggio accessibili in modo trasparente a tutti i livelli aziendali. Bisognerà dare molta importanza ai feedback raccolti dai team, dai manager e dai clienti per avere un riscontro chiaro e significativo su come le nuove pratiche vengono percepite e sui risultati positivi o meno che stanno producendo. Promuovere quindi la condivisione delle opinioni e privilegiare l'ascolto continuo permetterà di sviluppare una cultura che faccia del miglioramento continuo basato sull'apprendimento dai successi e dagli errori il punto di forza dell'organizzazione. Andranno identificati dei tool digitali per gestire la grande quantità di feedback necessari e per tracciare le attività efficacemente, proponendo dei sondaggi periodici durante tutto il processo di trasformazione per misurare il grado di soddisfazione dei team e identificando aree di miglioramento trasformando le critiche in opportunità di crescita. Il monitoraggio costante permetterà di:

- rispondere rapidamente ai cambiamenti e agli ostacoli che si incontreranno durante il percorso di cambiamento,
- apprendere dall'esperienza e ottimizzare le pratiche,
- aumentare il coinvolgimento e la motivazione dei membri dei team che si sentiranno valorizzati dall'attenzione posta ai loro feedback,
- garantire che tutte le attività siano sempre allineate per il corretto raggiungimento degli obiettivi aziendali.

Per garantire una trasformazione Agile di successo in Santander Consumer Bank Italia il monitoraggio dovrà assumere un ruolo centrale come componente essenziale del cambiamento. L'azienda potrà garantire che la trasformazione rimanga in linea con le aspettative, adattandosi ai cambiamenti e imparando dall'esperienza vissuta, diventando un'azienda di successo con una nuova cultura collaborativa e orientata al valore e che sostiene tutti i dipendenti e la loro crescita personale e professionale, favorisce l'innovazione e risponde con agilità alle esigenze del mercato. Sebbene si tratti di un lungo

processo di cambiamento, l'azienda beneficerà a lungo termine di un consolidamento della propria posizione competitiva, diventando il fiore all'occhiello del settore bancario e dei prodotti di credito al consumo, capace di ispirare fiducia nei clienti e di attrarre i migliori talenti da integrare nella propria organizzazione.

CONCLUSIONE

Il percorso compiuto per la realizzazione di questa tesi ha permesso di esplorare l'applicazione dei principi Agile e lean partendo da un'attenta analisi teorica fino ad arrivare all'esperienza pratica vissuta in prima persona durante i sei mesi di tesi in azienda e di tirocinio presso Santander Consumer Bank Italia. Il punto di partenza di questo percorso ha permesso di analizzare e presentare le origini del lean thinking e della metodologia Agile andando a comprendere quanto sia importante la diffusione di una nuova cultura aziendale per l'avvio di un cambiamento significativo. L'introduzione di una nuova metodologia, prima ancora dell'adozione di nuovi strumenti, è un cambiamento nel modo di pensare di tutte le persone che fanno parte di una azienda. Questo è da considerarsi valido indipendentemente dal ruolo o dalla mansione svolta e trova nella leadership e nel top management un punto di riferimento essenziale per guidare una trasformazione che altrimenti rischia di rimanere confinata a piccoli gruppi di persone senza generare un vero e proprio cambiamento significativo dell'intera organizzazione. Una volta analizzati i presupposti per Pensare Agile si sono approfonditi i framework Scrum e SAFe che sono tra i più diffusi e utilizzati per l'implementazione dei principi Agile e lean nelle aziende. Se da un lato Scrum ha permesso di comprendere come si possano organizzare al meglio i membri all'interno di un team di lavoro e su come sia importante avere un approccio iterativo orientato all'analisi dei risultati e al miglioramento continuo, dall'altro SAFe ha permesso di fare chiarezza su come si possano gestire in modo flessibile e snello anche quelle realtà aziendali più articolate e complesse dove sembrava molto difficile poter implementare una metodologia Agile. I modelli possono quindi essere di grande aiuto per l'implementazione di una nuova metodologia aziendale a supporto del cambiamento culturale, fornendo una base concreta su cui costruire una trasformazione di successo. L'analisi del modello Spotify ha permesso di notare come sia importante che un'azienda scelga gli strumenti più opportuni a supporto della nuova metodologia e faccia un lavoro attento di adattamento al proprio contesto valutando le specificità della propria struttura organizzativa. È stato possibile quindi capire che per Agire Agile le aziende hanno a disposizione diversi strumenti tra cui scegliere e che questi

vanno non solo perfezionati inizialmente, ma costantemente aggiornati per restare sempre al passo con le nuove esigenze, l'evoluzione tecnologica e i cambiamenti del mercato di riferimento.

Il confronto tra le metodologie Agile e l'esperienza reale vissuta in Santander Consumer Bank Italia ha evidenziato che un approccio più tradizionale della gestione dei progetti con modelli rigidi come quello Waterfall può presentare criticità significative in un contesto altamente dinamico e competitivo come quello bancario e dei prodotti di credito al consumo. Inoltre, entrare in contatto quotidianamente con il contesto aziendale è stato molto importante per notare da vicino delle problematiche che effettivamente sono in linea con quelli che sono stati i presupposti teorici che hanno dato origine al lean thinking e al Manifesto Agile, confermandone la rilevanza e l'attualità anche in settori diversi da quelli di origine. L'occasione di gestire diverse situazioni in autonomia durante il tirocinio ha permesso di mettere in pratica alcuni degli aspetti fondamentali delle metodologie Agile, rafforzando la consapevolezza che una mentalità di crescita combinata a un approccio flessibile che non si limita a seguire schemi prefissati può portare a risultati soddisfacenti anche in poco tempo. In tal senso, le lessons learned raccolte durante il tirocinio unite alle basi esplorate del pensiero e dell'agire Agile hanno permesso di formulare delle Actions mirate a migliorare i processi e l'efficienza aziendale.

Il passo finale del percorso teorico e pratico ha portato a concentrarsi sulle possibili azioni concrete che le aziende possono intraprendere per migliorare i propri processi organizzativi per affrontare con successo le sfide quotidiane. Si è visto come sia possibile adottare alcune soluzioni a breve termine che mirano ad apportare dei benefici immediati, come l'Agile Adoption, ma la messa a terra di una strategia con obiettivi a lungo termine, come l'Agile Transformation, rappresenta la vera e propria svolta per trasformare completamente l'organizzazione a ogni livello e in ogni area per far sì che l'azienda sia competitiva e in linea con il mercato e i bisogni del cliente finale. Analizzare uno dei casi di Agile Transformation di successo come quello di Paypal ha permesso di conoscere alcune delle possibili sfide e delle opportunità di un cambiamento consistente e radicale in una grande azienda. In questo modo è stato possibile formulare alcune best practices

e alcuni consigli informati per l'avvio di una trasformazione Agile in Santander Consumer Bank Italia in aggiunta alle azioni immediate proposte sulla base delle lessons learned. È molto interessante evidenziare come questo percorso ha permesso di mettere in luce che un approccio tradizionale presenta numerose limitazioni soprattutto nell'ambito dei progetti IT nel settore bancario e dei prodotti di credito al consumo. Queste limitazioni emergono dalla rigidità intrinseca di modelli come quello a cascata che fatica ad adattarsi facilmente a un contesto dinamico e in continua evoluzione. Le pianificazioni iniziali dettagliate e le fasi in una sequenza rigida non consentono di rispondere rapidamente e con semplicità ai cambiamenti delle esigenze dei clienti o alle inevitabili problematiche tecniche che emergono durante lo sviluppo di progetti articolati e complessi. In più, possono verificarsi dei ritardi sostanziosi che si propagano lungo tutte le fasi, rischiando di ricadere in costi aggiuntivi e in prodotti finali che dal momento in cui diventano disponibili sul mercato sono già stati superati e necessitano di miglioramenti. Per questo motivo la metodologia Agile permetterebbe di ovviare alla rigidità dei metodi tradizionali rappresentando una valida soluzione delle loro limitazioni proponendosi come un'alternativa che consenta di gestire i progetti in modo più efficiente anche quando la complessità aumenta e l'incertezza introduce ostacoli da superare tempestivamente grazie alla flessibilità. Per questo motivo, la trasformazione Agile per Santander Consumer Bank Italia rappresenterebbe l'opportunità di posizionarsi come un esempio innovativo nel settore bancario in tutta Italia adottando dei modelli organizzativi che le consentirebbero di rispondere con maggiore prontezza e flessibilità alla continua evoluzione del mercato. Le raccomandazioni proposte rappresentano un punto di partenza per un percorso di trasformazione che potrebbe estendersi a tutte le aree aziendali, contribuendo al raggiungimento di nuovi obiettivi strategici a lungo termine che valorizzino ulteriormente le risorse interne, migliorino l'efficienza operativa e promuovano un ambiente di lavoro collaborativo e innovativo, generando benefici concreti e durevoli nel tempo.

L'esperienza di tesi in azienda e di tirocinio è stata un'occasione unica per osservare dall'interno e concretamente le dinamiche di un'organizzazione complessa, comprendendo l'importanza di un approccio strategico all'introduzione di nuove metodologie. Tuttavia, sarebbe riduttivo sostenere che i sei mesi trascorsi finora presso Santander

Consumer Bank siano stati sufficienti per esplorare in profondità tutte le dinamiche e le sfaccettature di un'organizzazione molto articolata. Se si considerano in aggiunta le diverse collaborazioni e le varie joint venture nascono nuove sfide e nuove opportunità che necessitano di ulteriori analisi e approfondimenti. Ciò che non cambia è l'approccio che è stato seguito in questa tesi, un approccio basato sulla conoscenza approfondita di nuove metodologie e sulla successiva osservazione critica e analisi delle opportunità di un miglioramento orientato alla crescita e al cambiamento. Questo approccio potrà e dovrà essere applicato alle sfide future perfezionandosi con il tempo sulla base delle esperienze.

BIBLIOGRAFIA

Beck K. et al., "I principi sottostanti al Manifesto Agile", in *The Agile Manifesto*, Agile Alliance, 2001 <https://agilemanifesto.org/iso/it/principles.html> (ultimo accesso: 04/10/2024)

Dweck C. S., *Mindset: The New Psychology of Success*, Random House, New York, 2006

Errico P., *Agile Marketing: Fai felici i tuoi clienti in metà tempo, con risorse limitate, con team più allineati e motivati*, Edizioni LSWR, Milano, 2020.

Highsmith J., "History: The Agile Manifesto", in *The Agile Manifesto*, Agile Alliance, 2001, <https://agilemanifesto.org/history.html> (ultimo accesso: 04/10/2024)

Kniberg, H., Ivarsson, A., *Scaling Agile @ Spotify*, October 2012, <https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf> (ultimo accesso: 20/10/2024)

Krafcik J.F., "Triumph of the Lean Production System", in *Sloan Management Review*, vol. 30, n. 1, pp. 41-52, 1988

Moreira M., "Agile Adoption Roadmap", in Agile Alliance, *The Agile Journal*, november 2010, <https://www.agilealliance.org/wp-content/uploads/2016/01/Agile-Adoption-Roadmap-1110-AgileJournal-Mario-Moreira.pdf> (ultimo accesso: 15/11/2024)

Ōno T., *Toyota Production System: Oltre la produzione di massa*, Milano, Franco Angeli, 1993

PayPal, *Paypal Enterprise Transformation*, 2015, Sept 18. Retrieved from PayPalobjects.com: https://www.paypalobjects.com/webstatic/en_US/mktg/pages/stories/pdf/paypal_transformation_whitepaper_sept_18_2015.pdf (ultimo accesso: 22/11/2024)

Project Management Institute, *A Guide to the Project Management Body of Knowledge: PMBOK Guide*, 3a edizione, Newtown Square, Project Management Institute, 2004

Royce W. W., "Managing the Development of Large Software Systems", in *ICSE '87: Proceedings of the 9th international conference on Software Engineering*, pp. 328 - 338, IEEE Computer Society Press, Washington DC, 1987,
<https://dl.acm.org/doi/10.5555/41765.41801> (ultimo accesso: 30/11/2024)

Schwaber K., Sutherland J., *The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game*, 2020, <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf> (ultimo accesso: 10/10/2024)

Wakamatsu Y., *Toyota Way. Il valore delle persone nel Toyota Production System*, Franco Angeli, Milano, 2008

Womack J.P., Jones D.T., *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, Simon & Schuster, New York, 1996

Womack J.P., Jones D.T., Roos D., *The Machine That Changed the World: The Story of Lean Production*, Harper Perennial, New York, 1990

SITOGRAFIA

Banco Santander, *We are proud to be one of the 100 most valuable brands in the world*, 21 ottobre 2024, <https://www.santander.com/en/stories/we-are-proud-to-be-one-of-the-100-most-valuable-brands-in-the-world> (ultimo accesso: 02/11/2024)

Filippetti A., *Il (non) modello Spotify*, in Agile Plaza, 9 settembre 2022, <https://agileplaza.it/modello-spotify/> (ultimo accesso: 20/10/2024)

Scaled Agile Inc., *Extended SAFe Guidance*, <https://scaledagileframework.com/advanced-topics/extended-safe-guidance/> (ultimo accesso: 15/11/2024)

Scaled Agile Inc., *SAFe Implementation Roadmap*, <https://scaledagileframework.com/implementation-roadmap/> (ultimo accesso: 21/11/2024)

Scaled Agile Inc., *SAFe Lean-Agile Principles*, <https://scaledagileframework.com/safe-lean-agile-principles/> (ultimo accesso: 18/10/2024)

Scrum.org, *What is ScrumBut?*, <https://www.scrum.org/resources/what-scrumbut> (ultimo accesso: 16/11/2024)

Toyota Blog, *Jidoka – Toyota Production System guide*, 31 maggio 2016, <https://mag.toyota.co.uk/jidoka-toyota-production-system/> (ultimo accesso: 29/09/2024)

Toyota Blog, *Kanban – Toyota Production System guide*, 31 maggio 2013, <https://mag.toyota.co.uk/kanban-toyota-production-system/> (ultimo accesso: 30/09/2024)

RINGRAZIAMENTI

Vorrei ringraziare innanzitutto la mia Relatrice, Prof.ssa Nicoletta Gay, che mi ha seguito attentamente e sostenuto durante tutto questo percorso di tesi in azienda. La ringrazio per avermi consigliato durante il corso di Digital Strategy di valutare una posizione lavorativa nell'ambito della gestione dei progetti IT, facendomi scoprire un mondo meraviglioso che mi ha effettivamente appassionato molto. Ringrazio anche tutti i docenti dei bellissimi corsi che ho seguito in questi anni per avermi permesso di ampliare e consolidare le mie competenze.

Ringrazio la mia tutor aziendale, nonché la mia manager, Barbara Torricelli per avermi accolto a braccia aperte in Santander, facendomi vivere un'esperienza unica di formazione e di crescita personale e professionale. La ringrazio per avermi dato fiducia sin da subito, consentendomi di entrare immediatamente nel cuore dell'azione, prima affiancando una collega Product Owner in un progetto che è stato per me molto stimolante, poi gestendo in autonomia un nuovo progetto lasciandomi sperimentare un approccio differente da quello tradizionale. Ringrazio anche tutti i colleghi di lavoro che mi hanno aiutato ogni giorno e con cui ho avuto modo di confrontarmi spesso sulle tematiche affrontate in questa tesi.

Ringrazio caldamente i miei genitori e i miei parenti per avermi sostenuto ancora una volta anche durante questo percorso meraviglioso di laurea magistrale avendomi dato la possibilità di studiare lontano da casa, sostenendomi di fronte agli ostacoli.

Un ringraziamento di cuore va alla mia compagna Maristella che è sempre stata al mio fianco in questi anni intensi e che mi ha dato sempre una grande motivazione e un enorme sostegno sia nella quotidianità, sia nei momenti più difficili.

Un grande grazie va anche a Gian Marco per essere stato un valoroso compagno di viaggio con cui ho condiviso l'intero percorso di laurea ritrovando in lui un collega

competente e un ottimo amico. Un altro grazie va a Lisa per essere stata la Smart Assistant ideale nei diversi progetti seguiti insieme e a Diego, Silvia, Eugenio, Flavia e Anna che sono stati dei meravigliosi compagni di corso e amici. Un ultimo e sentito grazie va a Raniero, Marzia, Roberta, Duccio, Anita, Miriam, Irene, Melissa e Carlo e a tutti gli amici che in questi anni hanno creduto in me e mi hanno motivato con i loro preziosi consigli.