



UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

Corso di Laurea Magistrale in Ingegneria Gestionale

**MODELLI AGILE E PLAN-DRIVEN NELLO
SVILUPPO DI PRODOTTI TANGIBILI**

Relatore: Ch.mo Prof. Stefano Biazzo

Laureando: Edoardo Puozzo
Matricola N. 1105814 - IG

ANNO ACCADEMICO 2016-2017

RINGRAZIAMENTI

In primis volevo ringraziare il Ch.mo Prof. Stefano Biazzo per avermi seguito nella stesura del mio elaborato fornendomi, oltre all'interessante argomento, sempre nuovi spunti e critiche costruttive per poter condurre il mio lavoro al meglio.

Volevo ringraziare la mia compagna Daniela che mi ha sopportato e supportato durante il percorso che mi ha portato a laurearmi in Ingegneria Gestionale.

Ringrazio la mia famiglia che mi ha sempre spronato e sostenuto in tutto il mio percorso.

Ultima ma non ultima, ringrazio Whitford s.r.l. per avermi dato l'opportunità di poter continuare il mio percorso di studi.

SOMMARIO

INTRODUZIONE	1
1 MODELLO DI SVILUPPO DI PRODOTTO STAGE-GATE	3
1.1 Fasi tipiche del processo Stage-Gate	5
1.2 Evoluzione della metodologia Stage-Gate	9
2 METODOLOGIE AGILE	11
2.1 Lo Scrum	14
2.1.1 I fondamenti teorici	14
2.1.2 I ruoli principali	16
2.1.3 Gli eventi Scrum	20
2.1.4 Gli artefatti Scrum	25
2.1.5 Monitorare l'avanzamento verso un Goal	27
3 DIFFERENZE E LIMITI DEI MODELLI PLAN-DRIVEN E AGILE	29
3.1 Limitazioni dei modelli Plan-Driven	29
3.2 Limitazioni delle metodologie Agile	33
4 DA MODELLI SEPARATI A MODELLI IBRIDI	35
4.1 Integrazione Agile nei processi di sviluppo software	35
4.2 Agile-Stage-Gate	37
4.2.1 Gli stadi che interessano la metodologia Agile-Stage-Gate	39
4.2.2 Lo Scrum nell'Agile-Stage-Gate	40
4.2.3 Lo Stage-Gate nell'Agile-Stage-Gate	48
5 GRANDEZZE UTILIZZATE PER LA CLASSIFICAZIONE	51
5.1 Rigidità dei Gate	51
5.2 La presenza di feedback	52
5.3 Classificazione dei modelli presenti in letteratura	54
5.3.1 Modello Stage-Gate	54
5.3.2 Modello Agile (Scrum)	54

5.3.3	Modello ibrido Agile-Stage-Gate	55
5.4	Matrice Rigidità dei Gate/Presenza di feedback	56
6	CASI DI STUDIO	57
6.1	Caso Wikispeed	57
6.1.1	L'organizzazione	58
6.1.2	Lo sviluppo di prodotto	59
6.1.3	Vantaggi e sfide	61
6.1.4	Classificazione del caso di studio	62
6.2	Caso ThermoValves	64
6.2.1	Lo sviluppo di prodotto	64
6.2.2	Vantaggi e sfide	66
6.2.3	Classificazione del caso di studi	67
6.3	Caso Siemens Westinghouse Power Generation	69
6.3.1	Lo sviluppo di prodotto	71
6.3.2	Vantaggi e sfide	74
6.3.3	Classificazione del caso di studio	75
6.4	Caso Marel GRB	77
6.4.1	Lo sviluppo di prodotto	78
6.4.2	Framework di sviluppo di prodotto	79
6.4.3	Vantaggi e sfide	84
6.4.4	Classificazione del caso di studio	86
6.5	Caso Andritz Hydro AB Svezia	88
6.5.1	L'organizzazione	89
6.5.2	Lo sviluppo di prodotto	91
6.5.3	Vantaggi e sfide	96
6.5.4	Classificazione del caso di studio	98
6.6	Caso Compagnia A	99

6.6.1	Lo sviluppo di prodotto	99
6.6.2	Vantaggi e sfide	102
6.6.3	Classificazione del caso di studio	104
6.7	Caso LEGO Education	106
6.7.1	Lo sviluppo di prodotto	107
6.7.2	Vantaggi e sfide	109
6.7.3	Classificazione del caso di studio	110
7	ANALISI DELLA CLASSIFICAZIONE DEI CASI DI STUDIO	111
7.1	Posizionamento dei casi di studi nella matrice	113
7.2	Caratteristiche per diversi livelli di definizione dei prerequisiti	117
7.2.1	Alta definizione dei requisiti iniziali	118
7.2.2	Media-Alta definizione dei requisiti iniziali	119
7.2.3	Media incertezza dei requisiti	120
7.2.4	Media-Bassa definizione dei requisiti iniziali	120
7.2.5	Bassa definizione dei requisiti iniziali	121
	CONCLUSIONI	123
	BIBLOGRAFIA	127

INTRODUZIONE

Lo scopo di questo lavoro è di fornire una panoramica sulle diverse metodologie di sviluppo di prodotto ed una serie di metodologie modificate dipendenti dalla classificazione dei diversi progetti di sviluppo.

I primi capitoli di questo elaborato presentano le metodologie di sviluppo di prodotto formalizzate in letteratura.

Verranno esposte le metodologie Plan-Driven, in particolare la metodologia Stage-Gate, nella quale la rigidità dei Gate gioca un ruolo fondamentale per far rispettare una pianificazione minuziosa, derivata dal congelamento dei requisiti prima dell'inizio di ogni progetto.

Alle metodologie Plan-Driven, si contrappongono quelle Agile che nascono per lo sviluppo di software in cui i requisiti per alcuni progetti sono in continuo mutamento per seguire le richieste dei clienti e del Product Owner. La metodologia Agile più usata è lo Scrum. Esaminando i punti di forza e debolezza delle due metodologie, Cooper e Sommer nel 2016 hanno formalizzato un modello ibrido chiamato modello Agile-Stage-Gate che cerca di unire i vantaggi di entrambi i modelli.

Partendo dalle caratteristiche delle metodologie presenti in letteratura, si formalizza una classificazione qualitativa utilizzando una matrice le cui grandezze saranno la rigidità dei Gate e la presenza dei feedback da parte dei clienti.

Sono stati analizzati e classificati sette casi di studio presenti in letteratura, utilizzando le grandezze prese in considerazione per questo elaborato.

Dalla posizione dei vari casi di studio all'interno della matrice, si ricavano diverse caratteristiche dipendenti dalla completezza dei prerequisiti congelati all'inizio dello sviluppo di prodotto.

1 MODELLO DI SVILUPPO DI PRODOTTO STAGE-GATE

Il processo di sviluppo di prodotto Stage-Gate è la metodologia più diffusa e più usata nello sviluppo di prodotti tangibili.

Questa metodologia, mostrata in Figura 1.1 è chiamata anche a cascata, o ciclo di vita.

Il sistema Stage-Gate è un modello Plan-Driven in quanto l'obiettivo del progetto consiste nel migliorare l'efficienza e l'efficacia di un prodotto esistente o di sviluppare un prodotto con determinate caratteristiche. Si tratta di un modello sia concettuale sia operativo che permette di giungere alla realizzazione di un prodotto partendo dalla nascita dell'idea ed arrivando al suo lancio sul mercato [1], [2]. Il modello Stage-Gate consiste in una serie di Stage (o fasi) dove la squadra di sviluppatori lavora sul prodotto, ottiene le informazioni necessarie, integra i dati e ne fa l'analisi. Dopo ciascuna fase si ha un Gate all'interno del quale viene presa la decisione di continuare ad investire in tale progetto o meno (decisione Go/Kill) [2]. In ogni Stage ci sono delle attività lavorative che sono necessarie per proseguire con il progresso del progetto ed entrare così negli Stage successivi.

Gli Stage secondo Cooper [2] hanno la particolarità di essere progettati con lo scopo di ridurre l'incertezza e il rischio relativo al progetto, ottenendo continue informazioni per procedere allo sviluppo del prodotto. Inoltre la squadra che lavora al prodotto deve avere competenze cross-funzionali, così da riuscire a portare avanti attività in parallelo, permettendo un aumento della velocità di sviluppo del nuovo prodotto.

Al termine di ogni Stage si ha un Gate che funge da punto di controllo relativo al lavoro svolto nella fase precedente.

La metodologia di sviluppo di prodotto Stage-Gate è composta da diversi Gate ed è caratterizzata da un continuo input di requisiti che vengono forniti dal Gate precedente allo Stage successivo. Al termine di uno Stage, il Gate associato deve revisionare il lavoro fatto per garantire la corretta implementazione dei requisiti nel prodotto e decidere se il progetto può accedere o meno allo Stage successivo [1], [2].

Se la revisione del progetto dà esito positivo, il lavoro procede alla fase successiva; in caso contrario, esso continua nella stessa fase finché il problema non viene risolto.

Le frecce inverse o le iterazioni raffigurate nella Figura 1.1, indicano che è possibile ritornare alla fase precedente e apportare modifiche, ma ciò è difficile e spesso costoso. Questi grandi e generalmente inaspettati circuiti di retroazione vengono accettati se necessari, ma sono generalmente limitati a fasi adiacenti per ridurre al minimo il costo dei feedback in varie fasi. Le iterazioni all'interno di ogni fase sono presenti, ma non pianificate [3].

1.1 Fasi tipiche del processo Stage-Gate

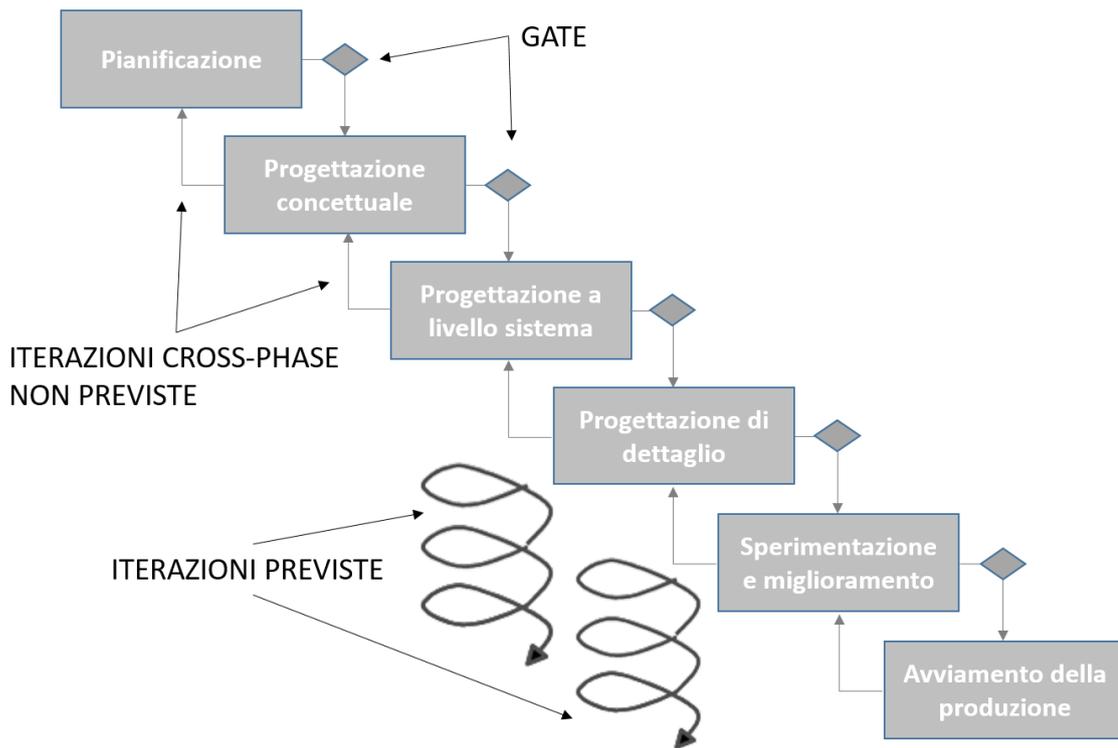


Figura 1.1 Processo di sviluppo di prodotto Stage-Gate [4]

Nella Figura 1.1 è rappresentato un esempio di processo Stage-Gate strutturato in sei fasi. Il processo di sviluppo di prodotto inizia con una fase di pianificazione del progetto, il cui output è la dichiarazione d'intenti (mission statement), che è il documento necessario nella fase di sviluppo concettuale e serve per guidare il Team di sviluppo nel raggiungimento degli obiettivi prefissati come requisiti di prodotto. La conclusione dell'intero processo di sviluppo avviene quando il prodotto è pronto per il lancio sul mercato.

Il processo può essere raffigurato come un imbuto nel quale all'inizio si hanno molti concetti di prodotto, tutte alternative potenzialmente di successo sul mercato, che vanno valutate. Con il procedere dello sviluppo, esse si riducono a causa dell'aumento del livello di definizione delle caratteristiche, fino a quando il prodotto può essere realizzato in maniera ripetibile dal sistema produttivo (Figura 1.2).

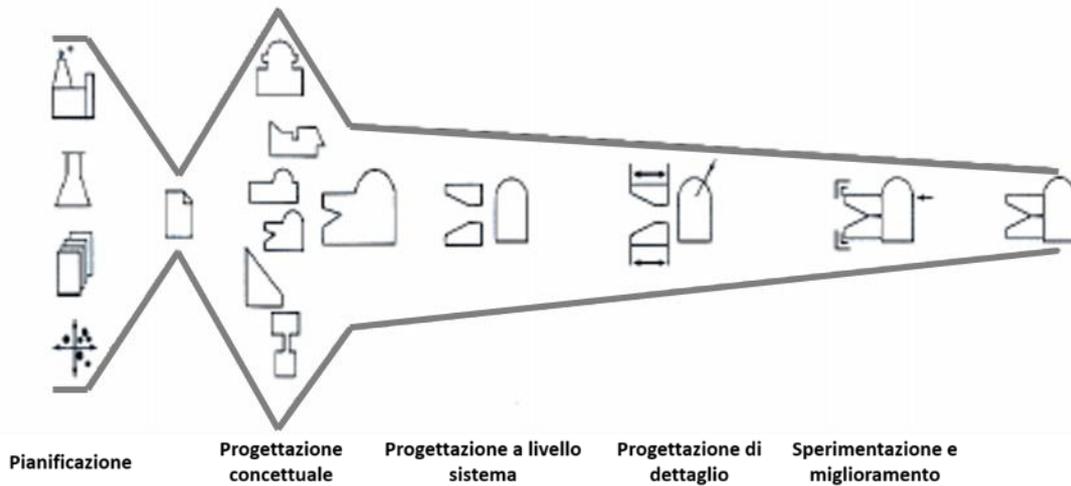


Figura 1.2 Processo tipo di sviluppo di prodotto [3]

Le tipiche fasi di un processo di sviluppo sono sei [3]:

0. **Pianificazione.** Si fa spesso riferimento all'attività di pianificazione come "fase 0" perché precede l'approvazione del progetto e l'inizio vero e proprio dell'attività di sviluppo del prodotto. Questa fase raccoglie la valutazione degli sviluppi tecnologici e degli obiettivi di mercato confrontandoli con altri progetti e con la strategia aziendale. Il risultato dell'attività di pianificazione è la dichiarazione d'intenti (mission statement) del progetto che specifica il mercato del prodotto, gli obiettivi del prodotto, le ipotesi fondamentali e i vincoli.
1. **Progettazione concettuale.** In questa fase, si analizzano i prerequisiti e i bisogni del cliente. Si generano e vengono valutati i possibili concetti di prodotto analizzando le potenzialità di ognuno nella generazione di un prodotto di successo sul mercato e selezionando i migliori per il proseguimento dello sviluppo. Il concetto di prodotto è la descrizione delle funzioni, delle caratteristiche e delle specifiche di un prodotto, accompagnato da un'analisi di prodotti simili o con applicazioni simili esistenti nel mercato che giustifichino lo sforzo economico dello sviluppo del prodotto.
2. **Progettazione a livello sistema (pre design).** Nella fase di progettazione a livello di sistema si definisce l'architettura di prodotto, il numero di sottosistemi e componenti. Inoltre vengono identificate le interfacce. Questo è importante per la fase successiva di sviluppo per completare i requisiti esterni, come i bisogni dei clienti, con i requisiti interni di modularità dei prodotti per poter così semplificare poi la fase di produzione. L'output di questo Stage è solitamente l'architettura

(layout) del prodotto, una specifica funzionale per ciascuno dei sotto sistemi del prodotto e un diagramma di flusso preliminare per il processo produttivo.

3. **Progettazione di dettaglio.** Questo Stage è la fase di progettazione vera e propria, nella quale si ha la definizione della geometria, dei materiali utilizzati, delle tolleranze delle singole parti del prodotto e l'identificazione di tutti i componenti o le parti del prodotto che verranno acquistate dai fornitori. Si definiscono, inoltre, il piano di produzione e le lavorazioni di ciascun componente da fabbricare all'interno del sistema produttivo aziendale. Il risultato di questa fase è una documentazione tecnica con disegni e file che descrivano i componenti del prodotto e le sue lavorazioni, le specifiche delle parti da acquistare e la pianificazione del processo di fabbricazione e assemblaggio del prodotto.
4. **Sperimentazione e miglioramento (testing and refenemet).** La fase di sperimentazione e miglioramento consiste nella costruzione e nel test dei vari prototipi di prodotto per verificarne l'aderenza ai requisiti presenti nel mission statement. La costruzione dei prototipi iniziali è realizzata con processi differenti rispetto alla fabbricazione dei prodotti che avverrà nel prossimo Stage e solitamente viene effettuata con metodi più artigianali. I prototipi iniziali sono sottoposti a test per determinare se il prodotto ha il funzionamento previsto e se soddisfa i principali bisogni dei clienti. I prototipi successivi sono solitamente costruiti con parti ottenute dai processi di produzioni prescelti, ma possono non essere assemblati con i processi di assemblaggio previsti. I prototipi beta vengono valutati all'interno dell'azienda e, tipicamente, vengono anche provati dai clienti nel loro ambiente d'uso. Lo scopo dei prototipi beta è solitamente quello di fornire risposte a domande sulle prestazioni e l'affidabilità al fine di identificare quali modifiche è necessario apportare al prodotto finale
5. **Avviamento della produzione e lancio.** Nello Stage di avviamento della produzione il prodotto viene fabbricato e assemblato utilizzando il sistema produttivo previsto. Lo scopo della fase d'avviamento è quello di istruire il personale nel migliore dei modi per costruire il prodotto ed inoltre si cerca di illustrare i possibili problemi nella fase di produzione ed il metodo per risolverli. I prodotti ottenuti durante questa fase di avviamento sono talvolta forniti a utenti selezionati i quali li esaminano accuratamente per individuare qualsiasi difetto residuo. Il passaggio dalla fase di avviamento alla produzione vera e propria è di solito graduale. Giunti ad un certo grado di avviamento della produzione, si ha il

momento del lancio del prodotto sul mercato. Questo lo rende disponibile a tutti gli utenti.

Segue l'analisi dei risultati e delle problematiche che seguono il lancio del prodotto sul mercato.

È interessante far notare che prima del Gate “progettazione di dettaglio”, all'interno del quale viene definitivamente deciso l'aspetto e le caratteristiche del prodotto, vi sono tre Stage.

Questo sottolinea la minuziosità nella valutazione e decisione per l'implementazione di un nuovo prodotto [1], [2].

Negli ultimi decenni, il sistema Stage-Gate ha subito alcune modifiche in quanto si è evoluto per fornire una metodologia efficace al mercato odierno.

Il cambiamento ha adattato la metodologia al concetto di open innovation, infatti le aziende hanno ridefinito il loro processo per lo sviluppo di nuovi prodotti, gestendo all'interno dell'organizzazione i flussi di idee, le tecnologie ed i prodotti già sviluppati provenienti da fonti esterne [5].

Per adattare la metodologia di sviluppo di prodotto Stage-Gate a questa gestione del flusso di idee e di nuovi network di alleanze con partner e fornitori esterni, sono stati modificati diversi Stage tra cui [2]:

- **Generazione di idee e pianificazione:** in questa fase le aziende non fanno solo riferimento alle idee provenienti dall'interno, ma anche all'ambiente esterno sia per risolvere i problemi legati ai consumatori, sia per entrare in contatto con partner e altre fonti di tecnologie disponibili che possono essere utilizzati come base per lo sviluppo interno.
- **Progettazione di dettaglio:** per la risoluzione di problematiche relative alla tecnologia da utilizzare per lo sviluppo del prodotto, le aziende possono affidarsi anche ad esperti esterni alla società, o acquistare le licenze e il know-how necessario per utilizzare una tecnologia già presente nel mercato.
- **Lancio e commercializzazione:** le società possono, al termine dello sviluppo del prodotto, vendere la licenza dell'uso di una determinata tecnologia o di un prodotto ad altre aziende.

1.2 Evoluzione della metodologia Stage-Gate

Negli ultimi anni la velocità e la richiesta di aderenza ai bisogni del cliente è talmente elevata che la metodologia di sviluppo di prodotto formalizzata da Cooper nel 1990 e implementata nel 2008 fatica ad essere competitiva. Lo Stage-Gate è stato criticato e definito un modello troppo lineare, rigido e programmato, inadatto alla gestione dei progetti di sviluppo di nuovi prodotti più innovativi e dinamici [6]. A fronte di queste critiche, Cooper [6] ha ideato il nuovo sistema idea-to-launch, che nasce dalla definizione di una panoramica delle possibili direzioni per lo sviluppo del sistema Stage-Gate.

Le caratteristiche principali che contraddistinguono questo sistema sono [6]:

- **Adattabilità e flessibilità:** il nuovo sistema riesce a modellarsi secondo i diversi progetti. Mentre il tradizionale Stage-Gate richiedeva che il progetto fosse definito e congelato prima di entrare nella fase di sviluppo, nell'idea-to-launch si ha un'evoluzione dei requisiti, adattandolo così alle continue richieste dei clienti e dei mutamenti del mercato. Questa metodologia crea nel sistema una spirale di sviluppo iterativo che permette di aggiornare costantemente i requisiti sulla base di feedback esterni. L'adattabilità del modello può essere vista nel fatto che questo comprende tre versioni del modello originale Stage-Gate dipendenti dalla rischiosità del progetto. Un altro aspetto che rende il modello flessibile sono le azioni presenti in ogni Stage e gli obiettivi da raggiungere in ogni Gate che sono unici per ogni progetto in relazione al contesto di mercato e alle esigenze di sviluppo del prodotto. Allo stesso modo il principio Go/Kill di ogni Gate non è più costituito da criteri standard e universali, ma è anch'esso flessibile e collegato alla gestione dell'impresa.
- **Agilità:** questo aspetto è il risultato dell'incorporazione dei principi Agile [7] nella metodologia di sviluppo tradizionale Stage-Gate.
- **Accelerazione:** il nuovo sistema idea-to-launch è focalizzato nell'accelerare il processo di sviluppo. Per le imprese è sempre più importante ridurre i tempi di sviluppo per ridurre il rischio di mercato, di lanciare un prodotto non più aderente ai bisogni del cliente. Per questo le aziende impiegano una serie di metodi per ridurre i tempi di sviluppo e tra questi: l'esecuzione simultanea di Stage diversi in sovrapposizione; la creazione di team dedicati che seguano il progetto progressivamente; la chiara definizione a priori dei possibili rischi nei quali può

incorrere il progetto; le tecnologie che può coinvolgere; l'utilizzo di software automatizzati.

Il nuovo sistema idea-to-launch è il risultato di uno studio empirico condotto da Cooper [6] per indicare la direzione che le imprese stanno prendendo per reinventare i loro sistemi Stage-Gate ed individuare quindi le prossime tendenze in materia [6].

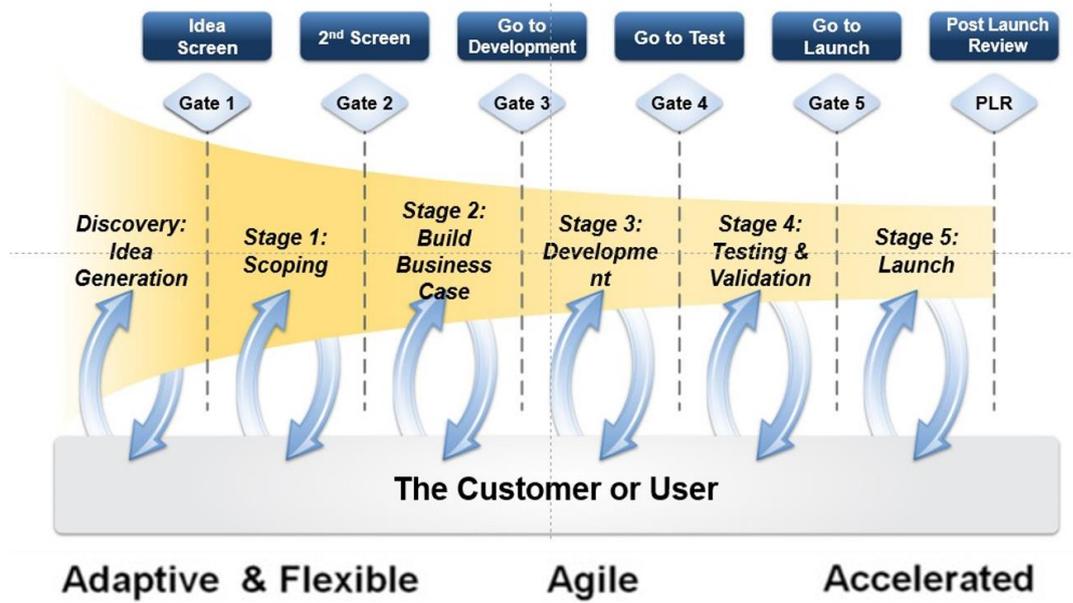


Figura 1.3 Sistema di nuova generazione idea-to-launch [6]

In questo modello Cooper, oltre a rendere lo sviluppo più Agile e avere un continuo feedback da parte dei clienti, come mostrato in Figura 1.3, propone tre versioni di Stage-Gate che si adattano al livello di rischio del progetto. Come si vede in Figura 1.4 le 3 versioni diventano più corte mano a mano che il rischio diminuisce. Questa versione di Stage-Gate è molto flessibile, dipendentemente dal prodotto e dal rischio del mercato [6].

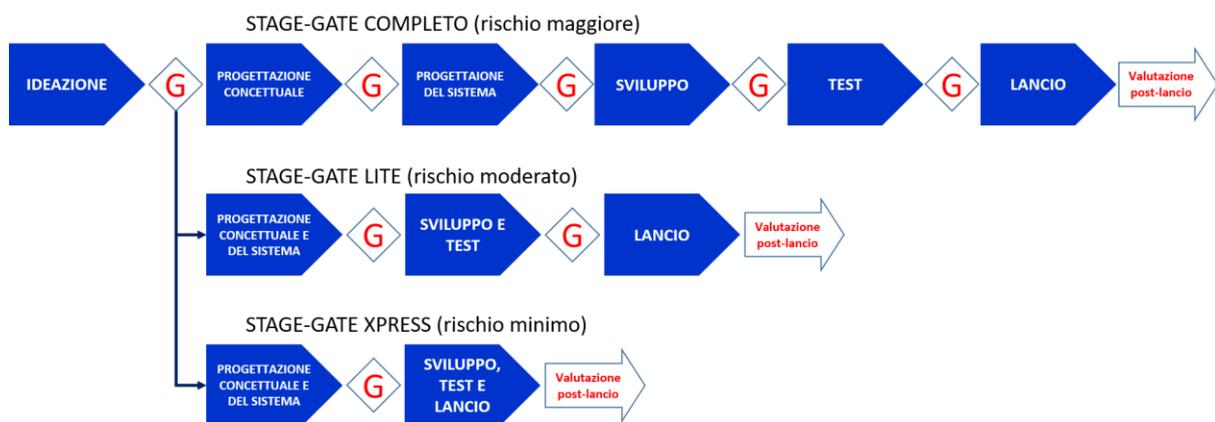


Figura 1.4 Sistema Stage-Gate scalabile [6]

2 METODOLOGIE AGILE

Nel febbraio del 2001, diciassette esperti dello sviluppo software alla ricerca di alternative alle metodologie di sviluppo tradizionale, si incontrarono in una località sciistica nelle Wasatch Mountains nello Utah [7]. Tutti i partecipanti erano concordi nel ritenere che le metodologie derivate dal modello a cascata non rappresentavano la soluzione più efficace come metodologia di sviluppo di prodotto, in un'ambiente in cui è importante possedere una certa "agilità" nell'affrontare lo sviluppo di sistemi software. Nella maggior parte delle situazioni i clienti non hanno chiaro ciò che desiderano e per questo vanno seguiti con particolare attenzione per cercare di far loro capire i requisiti di un prodotto che riesca a essere aderente ai loro bisogni. Per raggiungere questo obiettivo, bisogna costruire con i clienti un forte legame di collaborazione [7].

Uno dei punti principali, condiviso da tutti i partecipanti, risultava essere che tali requisiti non dovevano rimanere immutati durante tutto lo sviluppo di prodotto, ma subivano indubbiamente modifiche lungo il processo di realizzazione. Era necessario quindi prevedere questo cambiamento invece di controllarlo e sfruttarlo per permettere al cliente di ottenere una soluzione più compatibile alle nuove esigenze emerse, siano esse di tipo tecnologico, manageriale, progettuale o temporale [7].

Avere una metodologia di sviluppo Agile significa alleggerire i processi, come ad esempio, una riduzione significativa della documentazione prodotta in ogni fase del processo. Nei metodi tradizionali, lo sviluppo di prodotto può essere descritto formalmente in ogni aspetto e questo permette ad un nuovo membro la completa familiarizzazione con il processo insieme a una conoscenza dello stato di avanzamento del progetto. Questo però produce una quantità tale di documentazione che può essere un ostacolo in situazioni d'urgenza, in cui anche pochi giorni di ritardo fanno la differenza. Un componente del Team che necessita di essere istruito o aggiornato sul lavoro svolto in corso, può impiegare troppo tempo per essere informato sull'intero progetto [7].

Il risultato fu la redazione del Manifesto Agile che ha definito un set comune di 12 valori e principi generali per tutte le singole metodologie Agile [7]:

1. La nostra massima priorità è soddisfare il cliente per mezzo di tempestivi e continui rilasci di software di valore.

2. Siano benvenuti i cambiamenti nelle specifiche, anche a stadi avanzati dello sviluppo. I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente.
3. Rilascia software funzionante frequentemente, da un paio di settimane a un paio di mesi, con preferenza per i periodi brevi.
4. Manager e sviluppatori devono lavorare insieme quotidianamente durante tutto il progetto.
5. Basa i progetti su individui motivati. Dai loro l'ambiente e il supporto di cui necessitano e confida nella loro capacità di portare il lavoro a termine.
6. Il metodo più efficiente ed efficace di trasmettere informazione verso e all'interno di un Team di sviluppo è la conversazione faccia a faccia.
7. Il software funzionante è la misura primaria di progresso.
8. I processi agili promuovono uno sviluppo sostenibile. Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere un ritmo costante.
9. L'attenzione continua per l'eccellenza tecnica e la buona progettazione esaltano l'agilità.
10. La semplicità, l'arte di massimizzare l'ammontare di lavoro non svolto, è essenziale.
11. Le migliori architetture, specifiche e design emergono da Team auto-organizzati.
12. A intervalli regolari il Team riflette su come diventare più efficace, dopodiché mette a punto e aggiusta il suo comportamento di conseguenza.

I principi su cui si basa una metodologia leggera che segue i punti indicati dall'Agile Manifesto, sono quattro [7]:

- **Individui e reciproche interazioni:** le relazioni e la comunicazione tra gli attori di un progetto software sono la miglior risorsa del progetto.
- **Distribuzione di un software funzionante:** bisogna rilasciare nuove versioni del software a intervalli frequenti e bisogna mantenere il codice semplice e avanzato tecnicamente, riducendo la documentazione al minimo indispensabile.
- **Collaborazione del cliente:** la collaborazione diretta offre risultati migliori dei rapporti contrattuali.

- **Risposta al cambiamento:** bisogna essere pronti a rispondere ai cambiamenti più che aderire al progetto. Il Team di sviluppo dovrebbe essere autorizzato a suggerire modifiche al progetto in ogni momento.

I firmatari del “Manifesto Agile” si auto definirono, in quella occasione, la “Agile Alliance” il cui scopo era diffondere le idee per nuovi e più realistici approcci alla realizzazione di prodotti software.

Inizialmente, i modelli Agile avevano il nome di “light weight” ma, con la pubblicazione del Manifesto Agile nel febbraio del 2001, viene introdotto per la prima volta il termine “Agile”.

2.1 Lo Scrum

Scrum è uno dei framework di processo più utilizzati per gestire lo sviluppo di prodotti complessi [8].

Non è un processo o una tecnica per costruire prodotti, ma è piuttosto un framework all'interno del quale è possibile utilizzare vari processi e tecniche. Lo Scrum chiarisce l'efficacia relativa del proprio Product Management e delle proprie pratiche di sviluppo così da poterle migliorare [9].

Questa metodologia fu sviluppata da due dei diciassette membri che scrissero il manifesto agile nel 1995, sei anni prima del meeting nella località sciistica nelle Wasatch Mountains nello stato dello Utah.

Jeff Sutherland e Ken Schwaber presentarono per la prima volta un articolo che descriveva la metodologia Scrum. Negli anni successivi Sutherland e Schwaber ne continuarono lo sviluppo, integrandolo con le loro esperienze lavorative e alcune best-practice dell'industria.

Il termine "Scrum" fu introdotto dai giapponesi Takeuchi e Nonaka che nel 1986 [10] descrissero un nuovo approccio allo sviluppo di prodotti commerciali che avrebbe aumentato la velocità e la flessibilità. I due autori la definirono come una strategia flessibile e olistica allo sviluppo di un prodotto, dove il Team di sviluppo lavorava come un'unica entità per raggiungere un obiettivo comune, in opposizione all'approccio sequenziale delle metodologie tradizionali.

Lo Scrum è la fase di mischia nel gioco del rugby, usata per riprendere il gioco dopo un'infrazione. L'analogia con la metodologia di Agile Software Development risiede nel fatto che il Team di sviluppo deve lavorare come una squadra che cerca di raggiungere l'obiettivo come unità, passando la palla avanti e indietro [10].

2.1.1 I fondamenti teorici

Lo Scrum si basa sul controllo empirico dei processi e parte dall'idea che la conoscenza derivi dall'esperienza e che le decisioni si basino su ciò che si conosce. Lo Scrum utilizza un metodo iterativo ed un approccio incrementale per ottimizzare la prevedibilità ed il controllo del rischio [9].

I principi alla base di questa metodologia sono:

- **Trasparenza.** Le parti del processo più significative devono essere visibili ai responsabili e a chi usufruisce del risultato finale. La trasparenza richiede che quegli aspetti siano definiti da uno standard comune in modo tale che gli osservatori condividano una comune comprensione di ciò che viene visto.
- **Ispezione.** Chi utilizza lo Scrum deve controllare frequentemente gli artefatti (modi di documentare il lavoro) previsti dalla metodologia di sviluppo di prodotto e l'avanzamento verso un obiettivo con lo scopo di rilevare le eventuali deviazioni indesiderate. Queste ispezioni non devono essere tanto frequenti da intralciare il lavoro. I controlli sono più utili se eseguiti con particolare attenzione da parte di chi ha l'abilità e la competenza necessaria a effettuarli.
- **Adattamento.** Se chi ispeziona il processo di sviluppo di prodotto verifica che uno o più aspetti del processo sono al di fuori del range assegnato, deve subito segnalarlo perché sia implementato per rispondere alle esigenze dello sviluppo del prodotto. Se questo non avvenisse, il prodotto finale potrebbe non essere aderente ai requisiti che sono stati chiesti. Inoltre il cambiamento del processo dovrà essere effettuato nel minor tempo possibile per ridurre al minimo ulteriori sprechi di tempo e risorse.

Il framework Scrum è costituito dai ruoli, eventi, artefatti e regole a essi associati [9]. Ogni parte del framework serve a uno specifico scopo ed è essenziale per il successo e l'utilizzo di Scrum. Le regole di Scrum legano insieme gli eventi, i ruoli e gli artefatti governando le relazioni e le interazioni tra essi.

Lo Scrum definisce un insieme di attività che consentono al team di offrire un prodotto di maggior valore ai clienti in meno tempo. Le attività offrono la possibilità ai clienti di influire sul lavoro del team durante lo sviluppo. Questo approccio non tenta di definire tutte le funzioni del prodotto all'inizio di un progetto. Al contrario, il team lavora in brevi iterazioni (noti anche come Sprint) perfezionando il piano man mano che il lavoro procede [9].

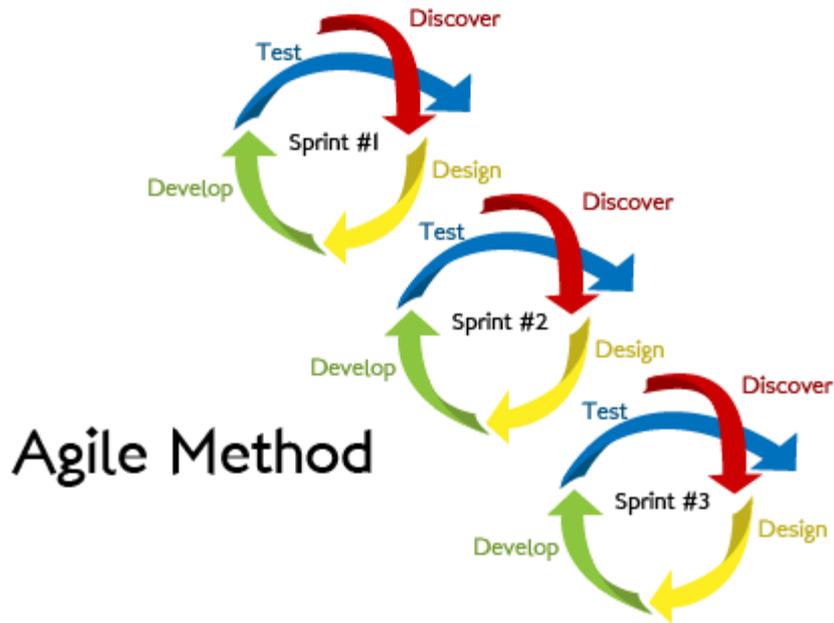


Figura 2.1 Metodologia di sviluppo di prodotto con l'uso dello Scrum [11]

2.1.2 I ruoli principali

I ruoli principali nel processo Scrum chiamati Team Scrum sono il cuore dello sviluppo del prodotto in quanto forniscono il loro know-how per realizzare il prodotto. Il Team Scrum è formato dal Product Owner, il Team di sviluppo (Development Team) e uno Scrum Master, come si nota in Figura 2.2.

I Team Scrum sono auto-organizzati e cross-funzionali. Auto-organizzandosi, scelgono come compiere il lavoro nel miglior modo possibile e pianificano, organizzano e gestiscono le loro attività autonomamente. Hanno inoltre la caratteristica di avere al loro interno tutte le competenze necessarie per realizzare il lavoro senza dover dipendere da nessuno al di fuori del team. Il modello di team in Scrum è progettato per ottimizzare la flessibilità, la creatività e la produttività. I Team Scrum rilasciano i prodotti in modo iterativo e incrementale, massimizzando le opportunità di feedback [9].

2.1.2.1 Il Product Owner

Questa figura rappresenta gli stakeholder ed è il portavoce delle richieste del cliente. Il Product Owner ha la responsabilità di massimizzare il valore del prodotto e del lavoro

svolto dal team di sviluppo, fornendo i requisiti richiesti dai clienti in forma di attività sul Backlog.

Il Product Owner, infatti, ha la responsabilità esclusiva di gestione del Product Backlog. Tale gestione prevede che [9]:

- Gli elementi del Product Backlog siano espressi in modo chiaro.
- Gli elementi del Product Backlog siano ordinati per raggiungere meglio gli obiettivi e le missioni.
- Il valore del lavoro svolto dal team sia ottimizzato.
- Il Product Backlog sia visibile, trasparente e chiaro a tutti e mostri su cosa lo Scrum Team lavorerà in seguito.
- Gli elementi del Product Backlog siano compresi al livello necessario dal Team di Sviluppo.

Il Product Owner è una persona sola, non un gruppo di persone. Può esprimere la volontà di un comitato nel Product Backlog, ma chiunque voglia cambiare l'ordine di un elemento deve rivolgersi al Product Owner [9].

2.1.2.2 Il Team di sviluppo

Il team di sviluppo è formato da professionisti che con il loro know-how e il lavoro di sviluppo creano, attraverso il completamento delle attività, di produrre un incremento "Fatto" di prodotto potenzialmente rilasciabile al termine di ogni Sprint. Lo Sprint è un'unità di base dello sviluppo nello Scrum ed è di durata fissa, generalmente da una a quattro settimane. Ogni Sprint è preceduto da una riunione di pianificazione in cui vengono identificati gli obiettivi e vengono stimati i tempi [9].

I Team di Sviluppo hanno le seguenti caratteristiche [9]:

- Sono auto-organizzati. Nessuno (neanche lo Scrum Master) dice al Team di sviluppo come trasformare il Product Backlog in incrementi di prodotto potenzialmente rilasciabili.
- Sono cross-funzionali, con tutte le competenze come Team necessarie a creare un incremento di prodotto.
- Scrum non riconosce alcun titolo ai membri del Team di sviluppo, indipendentemente dal lavoro eseguito dagli sviluppatori.

- Non contengono sotto-Team dedicati a particolari domini come il test del prodotto o il business analysis, non ci sono eccezioni a questa regola.
- I singoli membri hanno competenze specialistiche e diverse aree di competenza, ma è il Team di sviluppo nel suo complesso ad avere la responsabilità finale.

Il Team è preferibile che abbia una dimensione abbastanza piccola per rimanere agile, ma abbastanza grande per completare una certa quantità di attività in modo da avere un incremento di funzionalità significativo all'interno dello Sprint: di solito è composta da 3 a 9 persone [9].

Meno di tre persone nel Team di sviluppo diminuisce l'interazione tra i componenti e può portare ad un minore guadagno in termini di produttività. Più di nove persone nel Team di sviluppo richiede un eccessivo lavoro di coordinamento.

I ruoli del Product Owner e dello Scrum Master non sono inclusi nel conteggio, a meno che non stiano eseguendo anche loro il lavoro contenuto nello Sprint Backlog [9].

2.1.2.3 Lo Scrum Master

Lo Scrum Master ha il compito di rimuovere degli ostacoli che limitano la capacità del Team di sviluppo di completare le attività che portano al successo dello Sprint e degli obiettivi previsti. Lo Scrum Master ha anche il ruolo di controllare che lo Scrum Team aderisca ai valori, alle pratiche e alle regole dello Scrum.

Sebbene sia un ruolo manageriale, lo Scrum Master non è il Team Leader, ma piuttosto colui che facilita una corretta esecuzione del processo.

Lo Scrum Master gestisce inoltre gli interventi esterni aiutando a capire se possono essere utili o meno al team.

Lo Scrum Master rende un servizio al Product Owner in vari modi, tra cui [9]:

- Trovare le tecniche per una gestione efficace del Product Backlog.
- Aiutare lo Scrum Team a comprendere come creare gli elementi del Product Backlog in modo chiaro e conciso.
- Comprendere la pianificazione del prodotto in un ambiente empirico.
- Assicurare che il Product Owner capisca come ordinare gli elementi del Product Backlog per massimizzare il valore.
- Capire e praticare l'Agile.
- Facilitare gli eventi Scrum come richiesto e necessario.

Lo Scrum Master è di supporto anche al Team di sviluppo [9]:

- Supporta il Team di sviluppo per l'autogestione e la cross-funzionalità.
- Aiuta a creare prodotti di alto valore.
- Elimina gli ostacoli ai progressi del Team di sviluppo.
- Facilita gli eventi Scrum, se necessario.

Lo Scrum Master rende un servizio all'organizzazione in vari modi, tra cui [9]:

- Fare leading e coaching all'organizzazione per l'adozione di Scrum.
- Pianificare le implementazioni dello Scrum all'interno dell'organizzazione.
- Aiutare i dipendenti e gli stakeholder a capire e mettere in atto lo Scrum e lo sviluppo del prodotto in maniera empirica.
- Provocare il cambiamento che aumenta la produttività dello Scrum Team.
- Lavorare con altri Scrum Master per aumentare l'efficacia dell'applicazione dello Scrum nell'organizzazione.

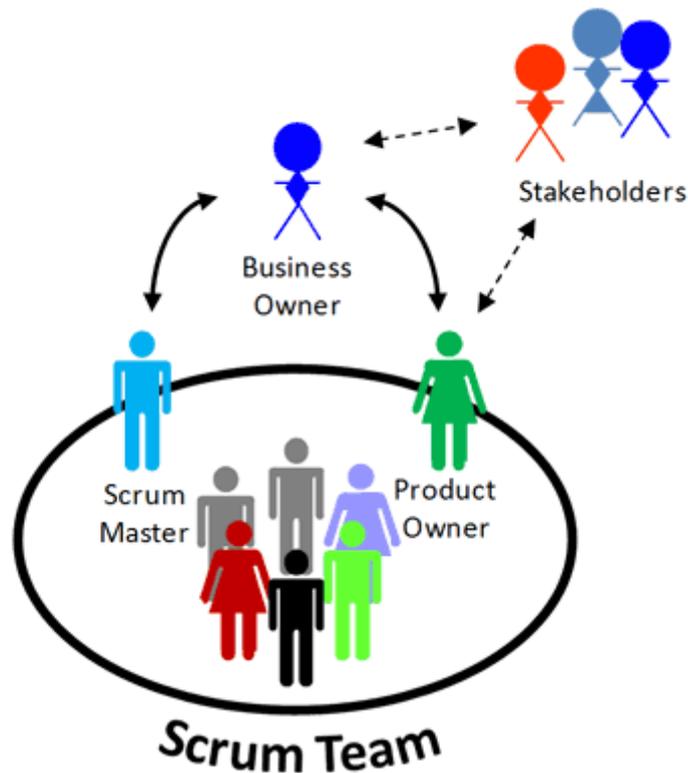


Figura 2.2 Esempio di Scrum Team [12]

2.1.3 Gli eventi Scrum

Gli eventi nella metodologia Scrum sono utilizzati per scandire il tempo dello sviluppo di prodotto, creare regolarità e ridurre al minimo la necessità di riunioni non definite dallo Scrum stesso. Lo Scrum utilizza eventi aventi un tempo definito (time-box), in modo ognuno di questi abbia una durata massima. Questo permette di evitare sprechi di tempo durante le attività, le riunioni e le pianificazioni [9].

Gli eventi nello Scrum sono progettati ad-hoc per avere la massima trasparenza critica e ispezione per ogni fase del progetto di sviluppo di prodotto. La mancanza di uno qualsiasi dei risultati riduce la trasparenza ed è un'occasione persa per ispezionare e adattarsi.

Quando uno Sprint inizia, la sua durata è prefissata e non può essere né accorciata né allungata. Gli altri eventi possono invece terminare quando è stato raggiunto lo scopo, assicurando che sia spesa l'appropriata quantità di tempo senza permettere sprechi all'interno del processo.

2.1.3.1 Lo Sprint

Lo Sprint è l'evento base dello sviluppo nello Scrum ed è di durata fissa, generalmente da una a quattro settimane. Ogni Sprint è preceduto da una riunione di pianificazione in cui vengono identificati gli obiettivi e vengono stimati i tempi per ogni attività [9].

Durante uno Sprint non è permesso cambiare gli obiettivi, quindi le modifiche sono bloccate fino alla successiva riunione di pianificazione. Durante questa riunione, infatti, gli obiettivi potranno essere inseriti e modificati per il prossimo Sprint. Ogni Sprint può essere visto come uno sviluppo di un prodotto con un tempo massimo che non può superare il mese. Come i progetti, gli Sprint sono utilizzabili per realizzare qualcosa. Ogni Sprint ha una definizione di ciò che si va a costruire, un progetto e un piano flessibile che guideranno la costruzione, il lavoro svolto e il prodotto risultante [9].

Al termine di ogni Sprint il Team di sviluppo consegna una versione potenzialmente completa e funzionante del prodotto, contenente gli avanzamenti decisi nella riunione di pianificazione. Gli Sprint migliorano la prevedibilità, assicurando che l'ispezione e l'adattamento del progresso verso un obiettivo siano effettuati almeno una volta al mese. Lo Sprint contiene altri eventi molto importanti per la metodologia quali lo Sprint Planning, il Daily Scrum, il lavoro di sviluppo, lo Sprint Review e lo Sprint Retrospective.

Uno Sprint può essere chiuso prima della scadenza del tempo massimo stabilito se lo Sprint Goal diventa obsoleto: questo potrebbe verificarsi se l'organizzazione cambia direzione o se le condizioni di mercato o della tecnologia cambiano [9].

L'unico membro dello Scrum Team che ha la possibilità di chiudere uno Sprint è il Product Owner anche se può essere influenzato dal management, dagli stakeholder, dal Team di Sviluppo o dallo Scrum Master. Le cancellazioni degli Sprint consumano risorse, poiché tutti devono partecipare di nuovo ad un altro Sprint Planning Meeting, per questo motivo sono molto rare [9].

2.1.3.2 Sprint Planning

La riunione per lo Sprint Planning è un incontro della durata massima di otto ore per uno Sprint di un mese, riducibile per Sprint più corti. Durante questo meeting lo Scrum Master, il Product Owner e il Team si confrontano su quali funzionalità del prodotto portano più valore per il cliente e su quali il Team dovrà lavorare durante quello Sprint [9]. Il risultato di questa riunione è la compilazione di uno Sprint Backlog che contiene le voci selezionate del Product Backlog e gli obiettivi per quello Sprint.

In questo meeting si utilizza il Product Backlog, analizzando le attività e le funzionalità incrementate nell'ultimo Sprint, le funzionalità ancora da implementare, la previsione di capacità del Team di Sviluppo durante lo Sprint e le performance registrate in passato del Team di Sviluppo per decidere le attività da svolgere nel prossimo Sprint. Gli elementi selezionati dal Product Backlog per lo Sprint sono definiti esclusivamente dal Team di Sviluppo.

Soltanto il Team di Sviluppo, infatti, è in grado di valutare cosa può compiere durante il prossimo Sprint. Dopo che il Team di Sviluppo ha selezionato gli elementi del Product Backlog su cui lavorerà nello Sprint, lo Scrum Team definisce gli Sprint Goal.

Lo Sprint Goal è l'obiettivo selezionato per lo Sprint che può essere raggiunto attraverso l'implementazione del Product Backlog. Esso fornisce una motivazione del lavoro sulle attività che il Team di sviluppo farà nel prossimo Sprint e sullo scopo che queste hanno nella realizzazione del prodotto finito. Questo è inoltre in grado di fornire un obiettivo comune che gli sviluppatori si impegnano a raggiungere, in modo da non avere una serie di azioni individuali senza un nesso logico [9].

Il Team di sviluppo dopo aver definito gli obiettivi dello Sprint e selezionato le voci del Product Backlog che devono essere implementate nel prodotto nello Sprint, deve decidere

come queste funzionalità di prodotto possano portare ad un incremento finito e indipendente alla conclusione di quello Sprint [9].

La pianificazione dello Sprint da parte del Team di Sviluppo inizia con la progettazione del sistema e il lavoro necessario per convertire il Product Backlog in un incremento del prodotto, avendo sempre in mente gli obiettivi dello sprint definiti come Sprint Goal. Può capitare che durante lo Sprint il lavoro necessario sia diverso da quanto il Team di Sviluppo aveva pianificato. In questo caso il Team di Sviluppo collaborerà con il Product Owner per determinare come rivedere al meglio il piano senza perdere di vista lo Sprint Goal.

Al termine della riunione di Sprint Planning, il Team di Sviluppo illustra al Product Owner e allo Scrum Master come ha intenzione di lavorare e come autogestirsi le attività e il tempo, in modo da raggiungere l'obiettivo prefissato alla fine dello Sprint e creare l'incremento previsto [9].

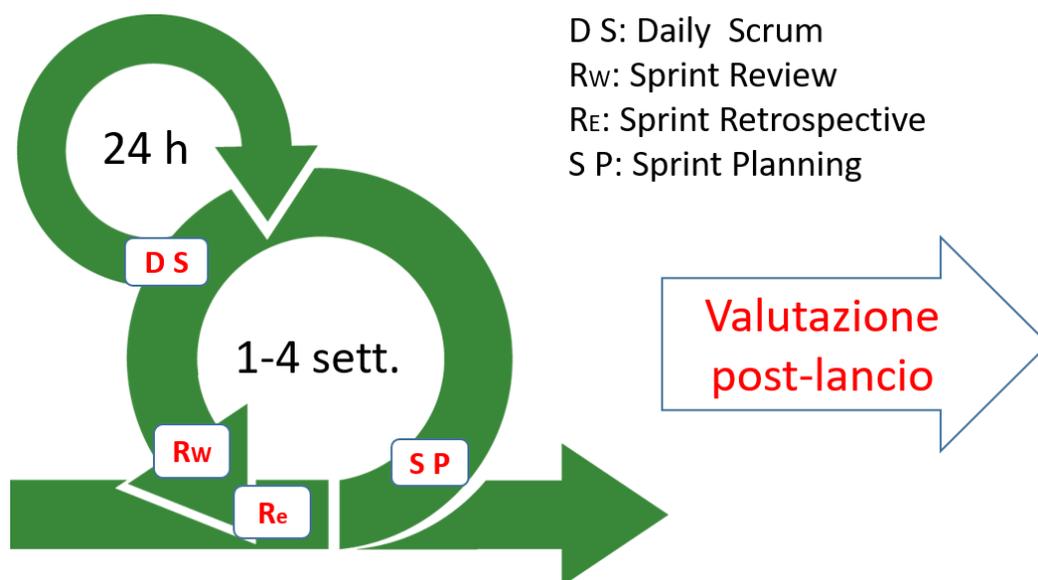


Figura 2.3 Schema della metodologia di sviluppo Scrum

2.1.3.3 Daily Scrum

Questo evento si tiene una volta al giorno durante il periodo di Sprint, di solito la mattina. In questa riunione chiamata Daily Scrum, ogni membro del Team di Sviluppo comunica agli altri membri su cosa ha lavorato il giorno prima, su cosa lavorerà quel giorno e che problemi potrebbero esserci per non portare a termine il lavoro. Questo meeting ha una serie di regole [9]:

- Tutti i membri del Team di sviluppo vengono preparati con gli aggiornamenti per la riunione.
- L'incontro inizia puntualmente anche se qualche membro del Team è assente.
- Il meeting dovrebbe avvenire ogni giorno nello stesso luogo e allo stesso tempo per ridurre la complessità.
- La durata del meeting è fissata al tempo massimo di 15 minuti.
- Si partecipa rimanendo in piedi, per non dare modo ai partecipanti di distrarsi ed isolarsi come accade nelle riunioni "tradizionali".
- Tutti sono benvenuti, ma normalmente solo i ruoli principali possono parlare.

Durante l'incontro ogni persona del Team di Sviluppo spiega:

- Cosa ha fatto ieri che ha aiutato il Team di Sviluppo a procedere verso lo Sprint Goal.
- Cosa farà oggi per aiutare il Team di Sviluppo a procedere verso lo Sprint Goal.
- Gli ostacoli che prevede di incontrare per sé stesso o per il Team di Sviluppo per arrivare allo Sprint Goal.

Lo Scrum Master si assicura che solo i membri del Team di Sviluppo partecipino al Daily Scrum. Questo meeting ha lo scopo di migliorare la comunicazione, eliminare altri incontri che sarebbero solo uno spreco di tempo, identificare e rimuovere gli ostacoli allo sviluppo, evidenziare e promuove il rapido processo decisionale e migliorare il livello di conoscenza del progetto da parte del Team di Sviluppo. Rappresenta un incontro chiave d'ispezione e adattamento.

2.1.3.4 Sprint Review

Dopo la conclusione di uno Sprint si ha la riunione chiamata di Sprint Review. Questo evento della durata massima di 4 ore, riducibile per Sprint più corti di un mese, ha la funzione di ispezionare l'incremento e adattare, se necessario, il Product Backlog a nuove funzionalità su cui si è lavorato nello Sprint precedente. Durante la riunione di Sprint Review, lo Scrum Team e gli Stakeholder discutono su ciò che è stato fatto durante lo Sprint appena terminato.

Sulla base di questo confronto e dai cambiamenti apportati al Product Backlog durante lo Sprint, i partecipanti allo Sprint Review discutono sulle prossime funzioni che potrebbero

essere implementate nel prodotto per incrementarne il valore. Si tratta di un incontro informale e la presentazione del risultato sotto forma di prodotto finito eseguito nell'ultimo Sprint, ha lo scopo di suscitare commenti e promuovere il confronto.

Lo Sprint Review include i seguenti elementi [9]:

- I partecipanti includono lo Scrum Team e i principali stakeholder invitati dal Product Owner.
- Il Product Owner identifica ciò che è stato “Fatto” e ciò che non è stato “Fatto”.
- Il Team di Sviluppo discute su cosa è andato bene durante lo Sprint, quali problemi sono stati incontrati e come questi sono stati risolti.
- Il Team di Sviluppo mostra il lavoro “Fatto” e risponde alle domande sull'incremento di sviluppo del prodotto.
- Il Product Owner discute lo stato attuale del Product Backlog. Fa una previsione sulla possibile data di completamento in base alla misura dell'avanzamento attuale se necessario.
- L'intero gruppo collabora su cosa fare dopo, così lo Sprint Review fornisce un prezioso contributo alle successive riunioni di Sprint Planning.
- Passare in rassegna come il marketplace o il potenziale utilizzo del prodotto possa avere cambiato l'elemento di maggior valore da implementare prossimamente.
- Passare in rassegna la timeline, il budget, le funzionalità potenziali e il marketplace per la prossima release del prodotto prevista.

Il risultato del meeting di Sprint Review è un Product Backlog revisionato e modificato che definisca le attività classificate secondo la priorità da dare nel prossimo Sprint, per dare il massimo valore al prodotto sviluppato nel minor tempo possibile. Il Product Backlog può anche essere adattato globalmente per venire incontro a nuove opportunità.

2.1.3.5 Sprint Retrospective

L'evento Sprint Retrospective è un meeting della durata massima di tre ore per lo Sprint di un mese; per Sprint più brevi, l'evento è di solito più breve. È un'occasione per lo Scrum Team per confrontarsi sulla metodologia di sviluppo di prodotto cercando di creare un piano di miglioramento da attuare durante il prossimo Sprint. Lo Scrum Team si riunisce per lo Sprint Retrospective dopo lo Sprint Review e prima del successivo Sprint Planning.

Lo Scrum Master ha il compito di verificare che l'evento abbia luogo e che i partecipanti ne comprendano la finalità. Lo Scrum Master, oltre a istruire i partecipanti sulla metodologia per avere un corretto svolgimento nel tempo dedicatogli, partecipa egli stesso, come componente dello Scrum Team per la responsabilità che detiene, nel far rispettare le regole della metodologia Scrum e nella responsabilità dell'eliminazione degli ostacoli nel processo di sviluppo.

Lo scopo dello Sprint Retrospective è di [9]:

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

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

2.1.4 Gli artefatti Scrum

Il ruolo degli artefatti nello Scrum è di rappresentare il lavoro, il suo valore e di fornire trasparenza delle informazioni chiave, di modo che ognuno abbia lo stesso livello di comprensione dell'artefatto.

L'implementazione delle funzionalità di un prodotto con conseguente aumento di valore devono essere fatte nella più completa trasparenza degli artefatti in modo da avere una base decisionale solida.

Nel caso in cui gli artefatti non siano completamente trasparenti, tali decisioni possono essere imperfette, il valore può diminuire e il rischio può aumentare.

Compito dello Scrum Master è quello di lavorare con lo Scrum Team e con l'organizzazione per aumentare la trasparenza degli artefatti.

2.1.4.1 Product Backlog

Il Product Backlog è un elenco ordinato di idee per il prodotto ed è l'unica fonte di requisiti per le modifiche da apportare al prodotto. Il Product Owner è il responsabile del Product Backlog, compreso il suo contenuto, la sua disponibilità e l'ordinamento dei suoi elementi.

Quando viene compilato, all'inizio del processo di sviluppo di prodotto, il Product Backlog è una lista più o meno lunga piena di funzionalità da implementare nel prodotto. All'inizio del progetto non è mai completo ed evolve insieme al prodotto. Questo artefatto è dinamico e cambia continuamente per identificare ciò che serve al prodotto per essere appropriato, competitivo e utile.

Il Product Backlog al suo interno, contiene tutte le caratteristiche, le funzioni, i requisiti, le migliorie e le correzioni che costituiscono le modifiche da apportare al prodotto nelle future versioni. Per ogni elemento dell'elenco, vengono assegnati degli attributi quali: descrizione, ordine, stima e valore.

Il Product Backlog contiene delle stime approssimative sia del valore che dello sforzo necessario a svilupparle.

Le stime aiutano il Product Owner a calcolare la timeline e possono influenzare l'ordine dei Backlog item.

Il raffinamento del Product Backlog è l'atto di aggiungere dettagli, stime e ordine agli elementi. Lo Scrum Team decide come e quando il raffinamento è completato. Il raffinamento solitamente occupa non più del 10% della capacità del Team di Sviluppo.

Questa attività comprende ma non è limitata a [9]:

- Mantenere ordinato il Product Backlog.
- Eliminare o riposizionare verso il basso gli elementi che non sono più ritenuti importanti.
- Riposizionare gli elementi, nuovi o già esistenti, che assumono un'importanza maggiore.
- Dividere gli elementi in parti più piccole.
- Stimare gli elementi del Product Backlog.

Il Team di Sviluppo è responsabile di tutte le stime. Il Product Owner può influenzare il Team di Sviluppo aiutando a capire e selezionare i compromessi, ma coloro che eseguiranno il lavoro sono gli stessi che effettueranno la stima finale.

2.1.4.2 Sprint Backlog

Lo Sprint Backlog è la lista del lavoro che il Team di sviluppo deve effettuare nel corso dello Sprint successivo. Questa lista viene generata selezionando una quantità di storie/funzionalità a partire dalla cima del Product Backlog determinata da quanto il Team di sviluppo ritiene possa realizzare durante lo Sprint: ovvero avere una quantità di lavoro tale da riempire lo Sprint.

Il Team di sviluppo modifica lo Sprint Backlog durante tutto lo Sprint e lo Sprint Backlog emerge durante lo Sprint. Ciò si verifica quando il Team di Sviluppo opera attraverso il piano e viene a conoscenza di più dettagli sul lavoro necessario a raggiungere lo Sprint Goal.

Durante lo Sprint, il Team di sviluppo si auto-organizza per produrre un incremento di prodotto corrispondente allo Sprint Backlog, come concordato durante lo Sprint Planning. Questo significa che il Team si prende la responsabilità di produrre un incremento di prodotto in accordo con gli standard aziendali.

2.1.5 Monitorare l'avanzamento verso un Goal

Il lavoro totale rimanente per raggiungere un obiettivo può essere calcolato in ogni momento. In occasione di ogni Sprint Review, il Product Owner tiene traccia del lavoro totale rimanente. Il Product Owner confronta questa quantità con il lavoro che era stato calcolato come rimanente durante le precedenti Sprint Review, al fine di valutare l'avanzamento del lavoro correntemente stimato in rapporto alla scadenza desiderata per l'obiettivo. Tale informazione è resa in maniera trasparente a tutti gli stakeholder.

Vari indicatori di tendenza, come Burn-Down Chart o Burn-up, insieme ad altre pratiche proiettive, sono stati già utilizzati per effettuare previsioni relative all'avanzamento. Tali strumenti e tecniche si sono rivelati utili, senza però sostituirsi all'importanza dell'empirismo. In ambienti complessi, non è dato conoscere quello che accadrà in futuro, solo ciò che si è già verificato può essere utilizzato per prendere decisioni in futuro.

2.1.5.1 Incremento

Ogni Sprint genera un Incremento di Prodotto, che deve avere una qualità sufficientemente elevata da consentirne il rilascio agli utenti. L'Incremento è la somma di tutti gli elementi del Product Backlog completati durante uno Sprint e durante tutti gli Sprint precedenti.

Alla fine di uno Sprint, il nuovo Incremento deve risultare “Fatto”, il che significa che deve essere utilizzabile e deve incontrare la definizione di “Fatto” data dallo Scrum Team. Deve essere utilizzabile indipendentemente dal fatto che il Product Owner decida di rilasciarlo realmente o meno.

2.1.5.2 Definizione di “Fatto”

Quando un elemento del Product Backlog o un incremento è considerato “Fatto”, tutti devono capire cosa s'intende per “Fatto”. Sebbene il significato di “Fatto” vari in maniera significativa da Team a Team, i membri dello stesso Team di Scrum devono avere una comprensione condivisa di ciò che si intenda per lavoro completo, al fine di assicurare che ci sia trasparenza. Questa è la “Definizione di Fatto” per il Team ed è utilizzata per valutare quando il lavoro è completo sull'incremento di prodotto.

La stessa definizione guida il Team di sviluppo a capire quanti elementi del Product Backlog è possibile selezionare durante lo Sprint Planning Meeting. Lo scopo di ogni Sprint è di fornire incrementi di funzionalità potenzialmente rilasciabili che aderiscono alla definizione attuale di “Fatto” per il Team.

I Team di sviluppo forniscono ad ogni Sprint un incremento di funzionalità del prodotto. Questo incremento è potenzialmente utilizzabile. Un Product Owner può scegliere se rilasciarlo immediatamente. Ogni incremento è additivo a tutti gli incrementi precedenti ed è testato, garantendo che tutti gli incrementi lavorino insieme.

Via via che gli Scrum Team maturano, ci si attende che la loro definizione di “Fatto” si espanda per includere criteri più stringenti finalizzati ad una qualità maggiore.

3 DIFFERENZE E LIMITI DEI MODELLI PLAN-DRIVEN E AGILE

3.1 Limitazioni dei modelli Plan-Driven

La differenza principale tra metodologie Plan-Driven (come Stage-Gate) e Agile è l'accettazione del cambiamento. È la capacità di rispondere alla modifica che spesso determina il successo o il fallimento di un progetto [13].

Nei metodi Plan-Driven le funzionalità del prodotto vengono congelate all'inizio dello sviluppo, impedendo il cambiamento secondo le richieste successive del cliente o del mercato. Le metodologie Agile, invece, hanno la capacità di cambiare i requisiti del prodotto in qualsiasi momento dello sviluppo.

Martin Fowler e Jim Highsmith, fondatori del Manifesto Agile, affermano che la capacità di adattamento di un prodotto alle caratteristiche richieste è fondamentale, quindi è importante utilizzare una metodologia che permetta questi cambiamenti e non che cerchi di prevederli per prevenirli [14]. Uno dei più importanti punti di forza che rendono i processi Agile importanti per il mercato, specialmente in mercati turbolenti, è la capacità di cambiare in qualsiasi fase del progetto.

Nello studio condotto da Boehm [15] e Jones [16], è stato riscontrato che in un normale processo di sviluppo di prodotto, i requisiti che i clienti richiedono al momento del lancio del nuovo prodotto sul mercato variano del 25% rispetto ai requisiti che la società identifica dai clienti all'inizio del progetto.

Uno studio di ricerca è stato condotto nel 2004 da un gruppo di Standish comprendente 365 intervistati e riguardava 8380 progetti di aziende presenti nei principali settori industriali. Dai loro risultati, il 16.2% dei progetti è stato completato rientrando nei tempi e con il budget pianificato con tutte le funzionalità e i requisiti specificati.

Il 52.7% dei progetti sono stati completati ma hanno superato il budget, oltre che essere in ritardo e il prodotto risultante dallo sviluppo offriva meno funzionalità e funzioni di quelle pianificate.

Il 31,1% dei progetti invece è stato annullato a un certo punto durante il ciclo di sviluppo [14].

Un'altra limitazione delle metodologie dei Plan-Driven è la gestione della complessità. L'approccio della pianificazione del lavoro, sia a livello strategico che a livello operativo, funziona egregiamente in ambienti stabili e poco complessi. Per ambienti grandi, complessi, e in continua evoluzione questa metodologia di sviluppo di prodotto fatica ad essere applicata.

Nelle metodologie tradizionali, si creano schemi molto complessi per cercare di far aderire la pianificazione all'ambiente in cui si sviluppa il prodotto. In ambienti in continuo mutamento è indispensabile ricercare la semplicità. Dee Hock afferma che lo scopo e i principi semplici e chiari creano comportamenti complessi e intelligenti [14], infatti alcune aziende usano regole semplici per sopravvivere a mercati complessi e turbolenti. Ad esempio, Jack Welch, CEO di General Electric (GE), ha trasformato la sua azienda di un valore di mercato di 12 miliardi di dollari, a una delle società più preziose al mondo con un valore di quasi 500 miliardi di dollari. La sua strategia per il successo secondo il professor Stanford University, Kathleen Eisenhardt, si basava su due semplici strategie [14].

In primo luogo, Welch ha sviluppato un metodo semplice per analizzare i punti di forza e le sue debolezze. Ha poi venduto o eliminato le imprese che non erano abbastanza promettenti per guadagnare in ogni settore.

Welch ha amministrato la riduzione di compiti inutili e ha introdotto un processo chiamato Work-Out, durante il quale i dipendenti di tutti i livelli si incontravano per concentrarsi su un problema o un'opportunità. Le idee migliori uscite da questa riunione, indipendentemente dalla fonte, venivano attuate rapidamente [14].

La metodologia Agile promuove le stesse idee. Fowler e Highsmith affermano che in un progetto Agile è particolarmente importante utilizzare approcci semplici, perché sono più facili da modificare [14].

È più facile aggiungere qualcosa ad un processo molto semplice, piuttosto che togliere qualcosa da un processo troppo complicato [7].

Con semplici regole, i Team di sviluppo possono migliorare continuamente i processi e i prodotti senza una guida dettagliata o una metodologia complessa [17].

Le regole da seguire, che sono molto simili alle regole del Lean Manufacturing [18] sono:

- **L'eliminazione degli sprechi.** Eliminare qualsiasi cosa che non aggiunga valore al prodotto finale. I documenti, i diagrammi e i modelli prodotti come parte dello sviluppo del software devono essere ridotti al minimo, perché una

volta che viene fornito un sistema di lavoro all'utente non interessano in quanto non fanno parte del prodotto finito.

- **Fare l'inventario è spreco.** L'inventario dello sviluppo di prodotto è la documentazione in eccesso che è uno spreco di tempo nella produzione e nella revisione dei documenti. Piuttosto che avere una specifica dettagliata di 100 pagine, scrivere un set di regole e linee guida di 10 pagine. Questo è ciò che le metodologie Agile propongono, la documentazione dovrebbe essere mantenuta al minimo.
- **Massimizzare il flusso.** Piuttosto che richiedere mesi per mostrare al cliente il prodotto finale, si utilizza uno sviluppo iterativo dove parti piccole ma complete di un prodotto sono progettate e consegnate durante tutto il ciclo di sviluppo. Questo consente al cliente di avere una migliore idea di come funziona il prodotto.
- **Definire il prodotto il più tardi possibile.** Le pratiche di sviluppo che mantengono i requisiti flessibili e li definiscono vicini alla consegna del prodotto possono offrire un notevole vantaggio competitivo in un ambiente in continua evoluzione. Le metodologie Agile sono progettate per rispondere al cambiamento e riescono a far in modo tale da prendere le decisioni il più tardi possibile.
- **Motivare i lavoratori.** È importante fornire sia gli strumenti che l'autorità agli sviluppatori in modo che siano loro a prendere le decisioni piuttosto che i responsabili.

Come descritto sopra, uno dei problemi principali delle metodologie Plan-Driven è la documentazione. A differenza della metodologia Agile, la metodologia Plan-Driven non consente agli sviluppatori di prendere decisioni durante la progettazione [18].

Lo sviluppo sequenziale rispetto allo sviluppo iterativo, è un'altra differenza tra i modelli tradizionali e il modello Agile. Le metodologie Plan-Driven, infatti, chiedono feedback ai clienti nell'ultima fase del loro ciclo di vita del progetto dopo il lancio del prodotto [14].

Le metodologie Agile sostengono che il feedback da parte dei clienti dovrebbe essere integrato nello sviluppo quotidiano. La chiave dello sviluppo iterativo secondo Fowler è quella di produrre spesso versioni di prodotto che siano composte da un sottoinsieme delle caratteristiche richieste da mostrare agli stakeholder [19].

Uno studio condotto da Alan Mac Cormack, un professore di Harvard Business School, sullo sviluppo del processo software, su 30 progetti, ha dimostrato che un rilascio anticipato del design del prodotto in evoluzione al cliente è uno dei fattori per un progetto di successo [20]. Lo studio dimostra che i processi fondati sulla metodologia Agile, con brevi sviluppi iterativi, feedback continui, test e sviluppi incrementali migliorano notevolmente la qualità del software.

Un altro limite delle metodologie Plan-Driven è che gli sviluppatori non vengono coinvolti nella gestione dello sviluppo di un prodotto. Le metodologie tradizionali vedono le persone come componenti prevedibili che seguono il processo di sviluppo. Alistair Cockburn nel suo paper, descrive i componenti del Team come il fattore più importante nello sviluppo del software [21]. Se gli sviluppatori vengono trattati come pedine di un programma e non come degli individui, il loro morale si abbassa e la loro creatività diminuisce.

Le metodologie Agile si concentrano sui talenti e sulle abilità degli individui, non come i metodi tradizionali in cui gli sviluppatori hanno un ruolo specifico che gli viene assegnato e devono svolgere un compito preciso. Le metodologie Agile affermano che le persone possono rispondere e trasferire le idee più rapidamente quando si parla faccia a faccia piuttosto che leggerle in un report o in altra documentazione come nelle metodologie Plan-Driven.

Un'altra differenza tra le metodologie di sviluppo di prodotto tradizionale e Agile è il modo di misurare il successo del progetto.

Secondo le metodologie Plan-Driven un progetto per lo sviluppo di un prodotto, lo si può considerare di successo se questo risponde a tutti i prerequisiti fissati nelle prime fasi del progetto, inoltre deve rispettare il budget e la durata pianificata senza subire ritardi [14].

Le metodologie Agile definiscono invece un progetto di successo se il cliente ha ottenuto un prodotto aderente alle proprie esigenze che risponde ai requisiti e alle funzioni di cui ha bisogno nel momento in cui utilizza il prodotto.

3.2 Limitazioni delle metodologie Agile

L'obiettivo delle metodologie Agile è quello di incrementare le funzionalità dell'oggetto in fase di sviluppo avendo, alla fine di ogni Sprint, un prodotto pronto per il lancio sul mercato. Tuttavia, come dice Constantine, non tutte le funzionalità richieste per il prodotto possono essere spezzate in parti indipendenti per subire uno sviluppo incrementale [22]. Infatti, la più grande limitazione della metodologia Agile è il modo in cui si gestiscono le squadre più grandi. Cockburn e Highsmith affermano che lo sviluppo agile è più difficile se le squadre sono grandi.

Quando le dimensioni crescono le interfacce di coordinamento diventano un problema importante [21]. I due autori affermano inoltre che anche la comunicazione faccia a faccia che contraddistingue l'Agile dalle metodologie di sviluppo tradizionali, diventa più difficile e complessa con un numero di sviluppatori superiore a venti. Al contrario, i metodi Stage-Gate, si basano su grandi progetti pianificati e grazie a questa pratica è possibile coordinare un elevato numero di sviluppatori.

Nella metodologia Agile, la documentazione è ridotta al minimo. Questa pratica è controproducente nel momento della verifica del progetto da parte di un supervisore o un manager che dovrà controllare la corretta esecuzione del lavoro.

Questo può portare ad errori nello sviluppo di un prodotto che non possono essere facilmente rilevati da revisori esterni a causa della mancanza di documentazione. Al contrario i modelli tradizionali riducono questo rischio richiedendo tempo agli sviluppatori per redare report e altra documentazione per facilitare gli esperti nelle revisioni del progetto, anche se questi piani possono essere obsoleti o costosi da cambiare se si verifica una modifica [23].

Nelle metodologie Agile, la mancanza di documentazione è sopperita dalla conoscenza tacita degli sviluppatori. L'uso del tipo di conoscenza non documentata ha il pregio di non far perdere tempo agli sviluppatori nel formalizzarla, ma ha il difetto di essere gestita bene solo dalle persone di talento in quanto sono in grado di trasferirla in modo efficiente. Questo ha portato alcuni autori a pensare se è il personale coinvolto nell'uso della metodologia ad essere di talento o l'Agile ad essere una metodologia efficiente, infatti le persone di talento sono capaci di lavorare con qualsiasi metodologia venga loro proposta. Alistair Cockburn è d'accordo con queste argomentazioni, ma spiega che se le persone del progetto sono competenti e di gran talento, possono utilizzare quasi tutti i processi e

svolgere i loro incarichi. Se non sono abbastanza brave, nessun processo riparerà la loro inadeguatezza [21].

Un altro punto a sfavore della metodologia Agile è la mancanza di documentazione durante il processo dell'individuazione delle richieste del cliente. Secondo Boehm, i metodi Agile funzionano meglio quando i clienti si dedicano con il team allo sviluppo del prodotto e hanno la conoscenza necessaria per aiutare il Team fino al completamento del progetto [24].

In questa situazione, se si hanno molti clienti ci si trova di fronte a diversi punti di vista conflittuali tra di loro. Questo rischio potrebbe essere ridotto in metodi Plan-Driven utilizzando la documentazione, la pianificazione, le revisioni del progetto e le revisioni di esperti indipendenti per compensare la negligenza dei clienti in certe materie [24].

La mancanza di documentazione ha anche l'effetto negativo di rendere difficile l'inserimento nel team o all'interno del dipartimento di sviluppo di altri sviluppatori. Infatti i nuovi membri non hanno a disposizione un archivio in cui studiare i modi di risolvere problemi comuni, come nel caso di progetti di sviluppo che utilizzano la reportistica tradizionale. [14]

4 DA MODELLI SEPARATI A MODELLI IBRIDI

4.1 Integrazione Agile nei processi di sviluppo software

Il grande successo delle metodologie Agile, ha portato grandi aziende nel campo dello sviluppo del software ad integrare nella loro metodologia di sviluppo di prodotto tipicamente Plan-Driven. Karlström & Runeson nel loro studio condotto nel 2005 [25], hanno esaminato tre grandi aziende europee quali Ericsson, ABB e Vodafone e hanno visto come le metodologie Agile e Plan-Driven siano compatibili e riescano a sopperire ai loro limiti.

ABB opera nel settore dell'automazione industriale, fornendo sistemi al mercato del settore. Ericsson Microwave opera nel settore della difesa, costruendo sistemi di sorveglianza radar. Vodafone Group fornisce servizi e prodotti di telecomunicazione a un mercato globale di circa 140 milioni di clienti in 26 paesi [25].

Nei primi due casi la metodologia Agile, in particolare l'Extreme Program, è stata applicata a livello di sviluppo di prodotto, integrandolo in un modello Stage-Gate esistente. Nel terzo caso, nello studio condotto nell'articolo [25], si è voluta studiare l'integrazione della metodologia Agile dal punto di vista della gestione dei progetti e dei prodotti.

La metodologia è stata applicata per questi progetti pilota allo sviluppo software, essendo le metodologie Agile nate per questo scopo.

L'integrazione tra le metodologie Agile e quelle tradizionali ha portato alla creazione di modelli ibridi con una serie di vantaggi [26]:

- Migliore comunicazione interna della squadra, il team si sente più unito e sotto controllo.
- Migliori metodi di misura dell'avanzamento e più intuitivi visivamente per la gestione, ad esempio, Burn-Down chart.

- Pianificazione più efficiente, basata su risposte anticipate dei clienti sui requisiti più importanti del prodotto, evitando piani rigidi che portino a ritardi nella definizione di importanti caratteristiche alla fine dello sviluppo.
- Miglioramento e continuo feedback da parte del cliente.
- Miglioramento del morale del team di progetto: gli sviluppatori sono più motivati a migliorare la comunicazione ed il controllo sul progetto.

Dallo studio emergono inoltre delle sfide che le aziende che utilizzano questo tipo di approccio devono affrontare [26]:

- Le squadre sono dedicate esclusivamente a un progetto e hanno la tendenza ad isolarsi dal resto del dipartimento di sviluppo limitando così l'interazione tra i vari Team dedicati a diversi progetti.
- La pianificazione a lungo termine tende a essere ignorata a favore di un focus sull'attuale Sprint.
- Possono nascere attriti in particolare tra i dirigenti che devono rinunciare al controllo durante gli Sprint del processo di sviluppo.

Nonostante le sfide, la metodologia Agile fornisce al modello Stage-Gate strumenti potenti per il micro planning, il controllo del lavoro quotidiano e il reporting dei progressi [25], [27].

Nello studio dei casi sopracitati, è stato notato che lo sviluppo del software non è un'attività che viene eseguita solo dagli sviluppatori, ma coinvolge un ambiente di sviluppo composto da molti dipartimenti tra cui sviluppo hardware, marketing, pianificazione della produzione ecc., che devono essere gestiti e coordinati contemporaneamente [27].

Il modello Stage-Gate fornisce un supporto non solo per la comunicazione all'interno del progetto, ma anche per i manager che lo sponsorizzano. Il modello di sviluppo Agile offre quindi maggiore efficienza e attenzione mentre lo Stage-Gate fornisce un mezzo per coordinare il Team di sviluppo con altre squadre e comunicare con funzioni come marketing e senior management.

4.2 Agile-Stage-Gate

Cooper & Sommer nel loro articolo nel 2016 [26] hanno cercato di definire un framework per una metodologia di sviluppo di prodotto.

L'Agile-Stage-Gate ha la particolarità di unire i vantaggi di una metodologia Agile (avere continui feedback dei clienti e la capacità di auto-gestione del Team di Sviluppo), con i vantaggi del modello Stage-Gate (in particolare il modello Idea-to-Launch) quali la capacità di pianificazione a lungo termine e la presenza dei Gate come punti di controllo dell'avanzamento del progetto.

Il modello è stato pensato per essere applicato sia ai prodotti software che ai prodotti fisici. Mentre delle metodologie Agile, in particolare lo Scrum, sono di facile applicazione per lo sviluppo di prodotti software, per prodotti tangibili si trovano diverse sfide. I prodotti fisici, a differenza dei software, non possono essere spezzettati in prodotti finiti potenzialmente lanciabili sul mercato [26].

Questo, oltre a non creare Sprint indipendenti fra loro, fa in modo che i risultati dopo un'iterazione di sviluppo siano di difficile valutazione da parte dei clienti. Inoltre i prodotti tangibili hanno un tempo di produzione che non può essere trascurato. Tra la fase di fine sviluppo e la fase di lancio, il prodotto deve essere lavorato e validato dal controllo qualità prima della commercializzazione. Questo tempo potrebbe influire sul gradimento del prodotto da parte dei clienti in quanto i loro bisogni nel frattempo potrebbero essere cambiati [26].

Il modello di sviluppo di prodotto Agile-Stage-Gate è composto da una serie di Stage che al loro interno utilizzano principalmente la metodologia Scrum. Come descritto da Begel e Nagappan [8], la maggior parte delle aziende usa questa metodologia per lo sviluppo dei propri prodotti. Lo Scrum nell'Agile-Stage-Gate si basa su una serie di Sprint della durata circa 2-4 settimane. Alla fine di ogni Sprint, il team di progetto produce "qualcosa di fatto" che può essere dimostrato agli stakeholder e spesso ai clienti.

Ogni Sprint è pianificato in tempo reale, quindi il processo è altamente reattivo e adattativo. Le varie versioni, sviluppate in modo rapido, iterativo e incrementale di concetti, disegni e prototipi, forniscono risposte veloci e tempestive. Questo rende il prodotto più aderente alle richieste degli utilizzatori finali e migliora le probabilità del successo sul mercato. Le preferenze dei clienti (Voce of Customers) diventano quindi un driver dinamico in tutto il progetto, in quanto i clienti possono definire meglio le proprie

esigenze attraverso il loro coinvolgimento attivo (ad esempio, tramite frequenti test utente o co-progettazione del cliente) [26].

Il team di progetto è dedicato a un unico progetto e si trova a lavorare riunito in un solo luogo, garantendo così sviluppi più rapidi e una migliore comunicazione. L'intero processo è visibile, con nuovi sistemi di avanzamento nei quali è possibile visualizzare il piano, i progressi e i risultati in tempo reale. Gli Stage e i Gate rimangono parte del processo, con i Gate come punti decisionali (Go/Kill) fondamentali per eliminare i progetti meno proficui e per focalizzare l'attenzione sulla pipeline di sviluppo.

Gli Stage offrono una panoramica ad alto livello delle principali fasi del progetto, nonché approfondimenti sulle attività richieste in ogni fase e sui risultati previsti alla fine di ogni Stage. Anche se il metodo Agile-Stage-Gate viene principalmente utilizzato nella fase di sviluppo, può essere utilizzato anche nelle fasi di pre-sviluppo per sviluppare il concetto e valutare la fattibilità. In queste prime fasi, si può utilizzare l'Agile (principalmente il metodo Scrum) per risolvere problemi legati alla mancanza di conoscenza o effettuare ricerche di mercato per cogliere le preferenze dei clienti. Con una serie di Sprint mirati, lo Scrum permette di appianare uno o una serie di divari di conoscenza [26].

L'integrazione dell'Agile nel sistema Stage-Gate per i prodotti fisici, comporta inevitabilmente delle modifiche alla metodologia sia dello Scrum sia della metodologia Stage-Gate.

Agile e Stage-Gate sono abbastanza compatibili per le aziende manifatturiere, quindi non si tratta di adottare o un metodo o l'altro: il modello ibrido integra il meglio di entrambi i sistemi. In pratica, l'Agile è tipicamente implementato in un sistema già stabile di Stage-Gate [26].

L'Agile non è una soluzione a un sistema poco progettato o in grado di fornire una risposta più rapida ai cambiamenti, ma dà al progetto una maggiore visibilità e una maggiore flessibilità.

Agile-Stage-Gate è progettato per gestire progetti di sviluppo più dinamici, che affrontano i mercati fluidi e le mutevoli esigenze e requisiti dei clienti; l'incertezza e l'ambiguità delle informazioni è ben gestita dal nuovo sistema.

Non tutti i progetti di sviluppo possono essere sviluppati utilizzando l'Agile-Stage-Gate. Questo metodo infatti richiede molte risorse dedicate e ciò ne ostacola l'applicazione. Quindi, la maggior parte delle imprese limita l'applicazione di Agile-Stage-Gate ai progetti di sviluppo più importanti, più grandi, più innovativi o più rischiosi [26].

Ad esempio in Corning, solo i progetti più importanti e critici passano attraverso il nuovo sistema agile, circa il 20%, e hanno squadre dedicate. La maggioranza dei progetti di sviluppo di nuovi prodotti di Corning ha ancora membri di team distribuiti su più progetti [6].

4.2.1 Gli stadi che interessano la metodologia Agile-Stage-Gate

L'Agile viene spesso implementato innanzitutto nelle fasi tecniche del processo Stage-Gate, ad esempio per le fasi di sviluppo e test in Figura 4.1. Molte aziende dichiarano che i risultati positivi ottenuti dall'utilizzo di Agile nei loro progetti di sviluppo software, hanno ispirato i tecnici che sviluppavano prodotti fisici a provarlo. Inoltre, dal momento che ad ogni Sprint si devono presentare versioni funzionali o dimostrative del prodotto, la metodologia si adatta bene alle fasi tecniche di un progetto di nuovo prodotto. Inoltre le persone tecniche sono più inclini a dedicarsi ad un singolo progetto piuttosto che a molti progetti insieme e questo rende più semplice l'implementazione da un punto di vista delle risorse umane [26].

La difficoltà di applicazione della metodologia e del pensiero Agile in altre aree funzionali come la parte commerciale o la produzione, è che i lavoratori dovrebbero dedicarsi ad un unico progetto per avere la massima concentrazione solo su quello.

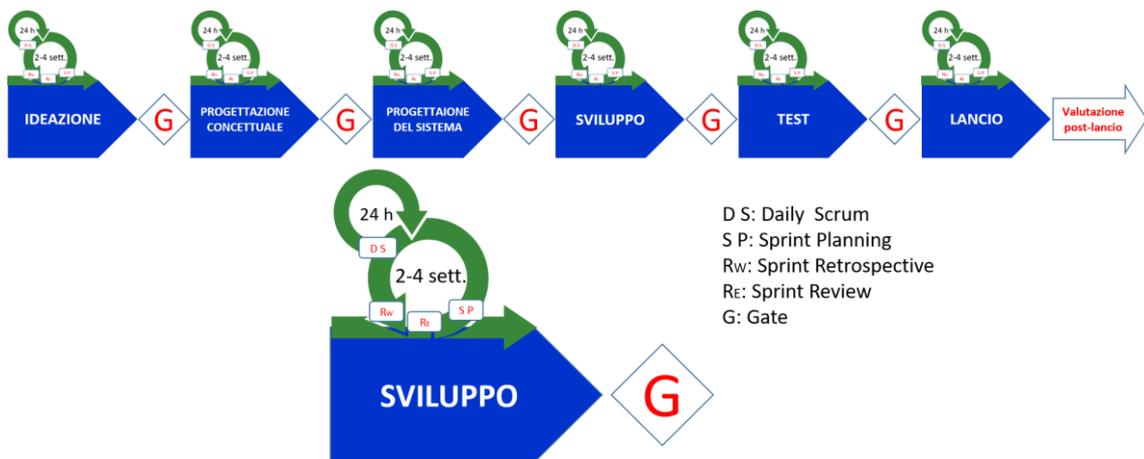


Figura 4.1 Schema del modello di sviluppo di prodotto Agile-Stage-Gate

4.2.2 Lo Scrum nell'Agile-Stage-Gate

L'utilizzo dello Scrum per lo sviluppo di prodotti fisici è molto simile all'applicazione per prodotto software [9]. Nel dettaglio, la metodologia usa principalmente nove elementi: tre eventi fondamentali, tre ruoli distinti e tre artefatti.

4.2.2.1 Tre Eventi fondamentali:

Come descritto del Capitolo 2.1.3 gli eventi scandiscono il tempo per versioni veloci e incrementali dei prodotti. Il primo evento è lo Sprint che è un lavoro a tempo determinato in cui viene eseguita una piccola parte del progetto. L'obiettivo è quello di avere un aumento di percentuale di prodotto completato alla fine di ogni Sprint, pronto per essere presentato al cliente e/o ad altri stakeholder interessati [26].

Nella revisione di uno Sprint, si presentano i risultati ai clienti e al Management: in pratica, non tutti gli Sprint producono un miglioramento di prodotto o un prototipo; alcuni possono produrre un progetto, un concetto, un prototipo semplice o anche i risultati di uno studio dei requisiti dei clienti (Voice of Customer) l'importante è che sia "qualcosa di Fatto". Alcuni produttori impiegano Sprint più lunghi rispetto alle normali 2-4 settimane, soprattutto quando i loro prodotti sono complessi. Questo Sprint più lungo consente al Team di avere più tempo per creare qualcosa di più concreto e più aderente alla realtà, da mostrare agli stakeholder. In altre aziende, le risposte dei clienti sono ritardate rispetto allo Sprint Review e certe volte l'intero Sprint successivo è dedicato a ottenere le risposte dei clienti [26].

Il secondo evento utilizzato è il Daily Scrum o stand-up quotidiano. In questo meeting, in cui il Team di sviluppo si riunisce all'inizio di ogni giorno per 10-15 minuti, vengono illustrati i risultati avuti il giorno precedente, i compiti da svolgere il giorno stesso e i problemi che si possono avere, come descritto nel Capitolo 2.1.

L'ultimo evento fondamentale per applicare correttamente la metodologia è lo Sprint Retrospective alla fine di ogni Sprint. Lo scopo di questa riunione non è quello di rivedere i risultati dello Sprint (questo è già fatto nello Sprint Review), ma è rivolto ai membri del team per migliorare il funzionamento della squadra.

4.2.2.2 I tre ruoli distinti

Il ruolo più importante in Agile è il singolo membro del team di sviluppo o progetto. In Agile, la squadra ha il potere di assumersi la responsabilità per l'esecuzione del progetto: è la squadra, e non il Management, che definisce, seleziona e assegna attività o compiti tra i membri del team.

Sulla base dell'obiettivo concordato dello Sprint, il lavoro del team è quello di definire le attività necessarie per raggiungere gli obiettivi dello Sprint, decidere chi di loro svolgerà le attività ed eseguire i compiti durante lo Sprint [26].

La metodologia Agile ha anche un responsabile di processo, chiamato Scrum Master, il cui ruolo è quello di sostenere le squadre che eseguono lo Sprint facilitando lo svolgere delle attività ed eliminando i possibili problemi di gestione.

Lo Scrum Master modera e aiuta lo svolgimento dei Daily Scrum, dei Sprint Planning e gli incontri post-Sprint. Assicura che il Team segua la metodologia Agile e che si utilizzino correttamente artefatti e strumenti. Lo Scrum Master è anche responsabile della rimozione degli impedimenti alla squadra, in modo da poter eseguire lo sviluppo rapidamente e senza blocchi [26].

Il terzo ruolo è il Product Owner, responsabile del Product Backlog. Il Product Owner lavora con il Team di sviluppo per assicurare che i requisiti di prodotto siano rispettati nello Sprint.

Il Product Owner ha le stesse responsabilità che ha un Project Leader nello sviluppo tradizionale del prodotto e si concentra sulla gestione degli stakeholder al fine di garantire il supporto, l'input e le risorse al progetto, nonché gestire il coinvolgimento del cliente [26].

4.2.2.2.1 Composizione del Team di sviluppo

In alcune aziende che utilizzano metodi Stage-Gate tradizionali, il progetto del prodotto sviluppato viene consegnato a un Sales Team. Nel fare questo, si perde lo slancio, la conoscenza, la responsabilità e la proprietà del progetto.

Al contrario, in Agile-Stage-Gate, il team di base rimane intatto dall'inizio alla fine del progetto. Mentre questo progredisce, nuove persone possono entrare a far parte del progetto. L'intensità di lavoro per area funzionale sarà ovviamente diversa durante il progetto, per esempio i progettisti e gli sviluppatori hanno un ruolo più importante durante

le fasi intermedie del progetto, ma per garantire continuità i membri del team rimangono nella squadra durante tutto il progetto fino alla revisione post-lancio [26].

I membri del Team di Agile-Stage-Gate devono aver un profilo professionale particolare: sono esperti nella loro area funzionale, ma detengono un ampio insieme di capacità generali.

Lavorando insieme, i membri del team funzionano come una vera squadra e sono in grado di aiutare gli esperti di altre aree funzionali. Nella pratica tuttavia, le persone con tali profili sono molto ricercate, e una squadra composta da tutti membri con questo profilo è molto difficile da trovare [26].

4.2.2.2.2 Nuovi ruoli e vecchi ruoli

Agile introduce alcuni nuovi ruoli e nuove terminologie, ma alcuni utilizzatori di Agile-Stage-Gate tendono ad usare ancora la dicitura ed i ruoli tradizionali. Ad esempio, alcune aziende non utilizzano il termine usato in Agile di Product Owner ma hanno un Project Leader, responsabile della guida del progetto. Progetti più grandi e più complessi hanno un Process Manager in Team, il cui ruolo è simile a quello del tradizionale Project Manager [26].

Il Project Leader è molto diverso da Project o Process Manager: infatti un Project Manager è tipicamente responsabile del coordinamento e rispetto dei tempi, assicurando che il team rispetti il piano di progetto e segua metodi di Project Management. Il Project Leader, al contrario, è un soggetto che vuole guidare il progetto verso il suo obiettivo finale, assicurare l'impegno degli stakeholder, generare una chiara proposta di valore e di solito sostenere il progetto.

Il Project Leader del progetto è spesso paragonato all'imprenditore che avvia il proprio business. Le caratteristiche dei responsabili efficaci del Team di progetto includono: credibilità, entusiasmo, imprenditorialità, competenze personali e capacità di gestione del progetto [26].

In Agile-Stage-Gate, sia il Project Leader, sia il Project o Process Manager sono ruoli preziosi e necessari e sono preferibilmente assegnati a due persone diverse: un Product Owner o un Project Leader e uno Scrum Master. Il Project Manager o lo Scrum Master è generalmente in grado di supportare più di un progetto alla volta, a seconda delle dimensioni e della complessità. Per i progetti più piccoli, talvolta il leader del Team riempie entrambi i ruoli come Leader e Project Manager [26].

4.2.2.3 I tre artefatti

Agile-Stage-Gate come la metodologia Scrum descritta nel Capitolo 2.1, prevede tre importanti strumenti visivi per aiutare a gestire e monitorare i progetti. Il primo è il Product Backlog (Figura 4.2), che è l'equivalente delle specifiche dei requisiti nelle metodologie tradizionali. A differenza di una definizione di prodotto, tuttavia, il portfolio di prodotti non contiene specifiche dettagliate (quali i materiali desiderati o i requisiti di prestazione), ma piuttosto le esigenze e le preferenze dei clienti, fornendo libertà al team di progetto di sperimentare il design del prodotto, limitandosi a seguire però le linee guida del Backlog [26].

In alcune aziende si parla di Backlog di progetto più che di prodotto, in quanto questo deve colmare anche lacune di conoscenza necessarie per le fasi successive e non solo di progettazione. Le voci sono classificate secondo la priorità per avere una sequenza ottimale, la voce più importante va affrontata per prima. Questo è un documento molto dinamico e viene continuamente aggiornato.

All'inizio di ogni Sprint, nella riunione di pianificazione, si prendono gli elementi prioritari dal Product Backlog e li si traduce in attività per creare lo Sprint Backlog. Alcune aziende utilizzano User Story nel loro Product Backlog. Piuttosto che i requisiti per il prodotto, le User Story descrivono come gli utilizzatori finali si aspettano il prodotto e come soddisfare le esigenze dell'utente. Alle User Story vengono assegnati degli Story Point per indicare il livello di lavoro richiesto [26].

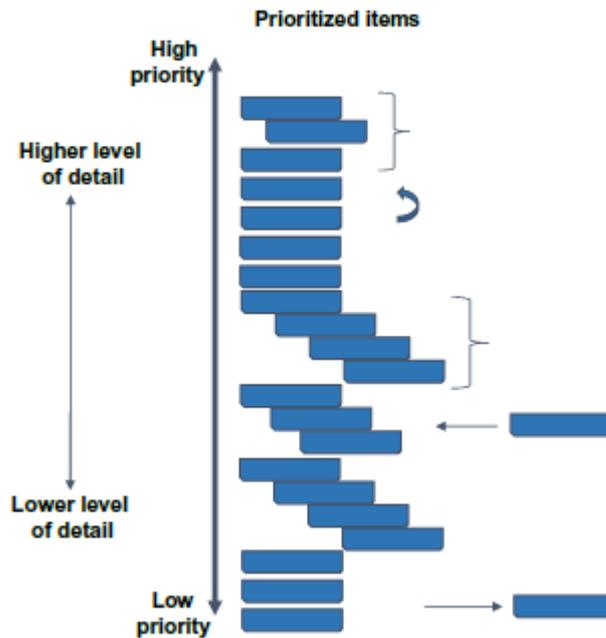


Figura 4.2 Rappresentazione del Product Backlog. Alle User Story vengono date delle priorità e nel momento dello Sprint Planning, queste vengono scelte per essere affrontate nello Sprint [26]

Il secondo artefatto è la lavagna Sprint o la lavagna Kanban.

La lavagna contiene lo Sprint Backlog, l'elenco delle cose da fare durante lo Sprint. In pratica, queste vengono poste come Post-It sulla lavagna o come attività su una lavagna digitale. Ci sono molti modi per impostare la lavagna Sprint, ma il formato più comune è quello di utilizzare tre colonne: "Da fare", "In corso" e "Fatto". Durante lo Sprint, le attività vengono avviate spostando un compito da "Da fare" nella colonna "In corso", con sopra il nome della persona che la sta svolgendo e, quando è finita, si sposta in "Fatto". Ciò crea un flusso continuo di attività che vengono spostate attraverso le colonne per popolare la colonna più a destra come illustrato in Figura 4.3. Se lo Sprint è stato ben pianificato, ci saranno solo le attività sufficienti per l'esecuzione di quello Sprint [26].

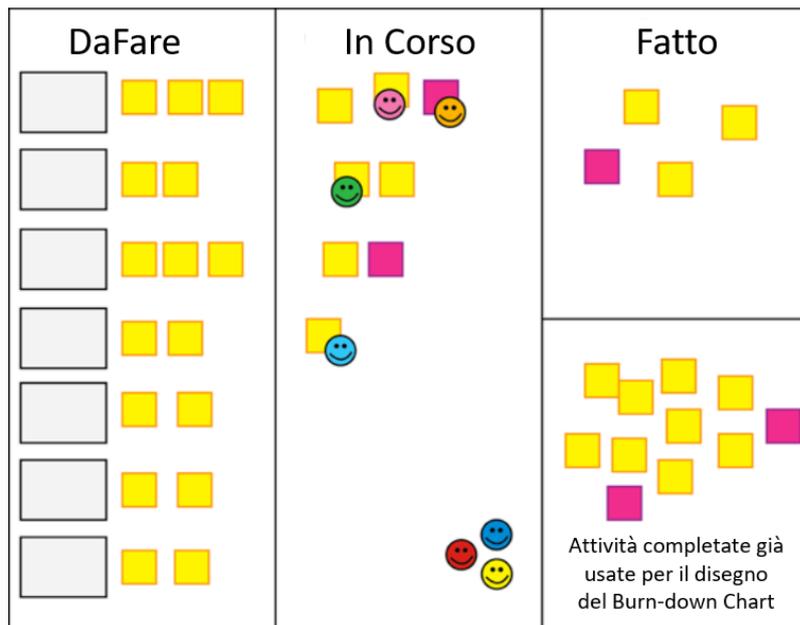


Figura 4.3 Lavagna Kanban [28]

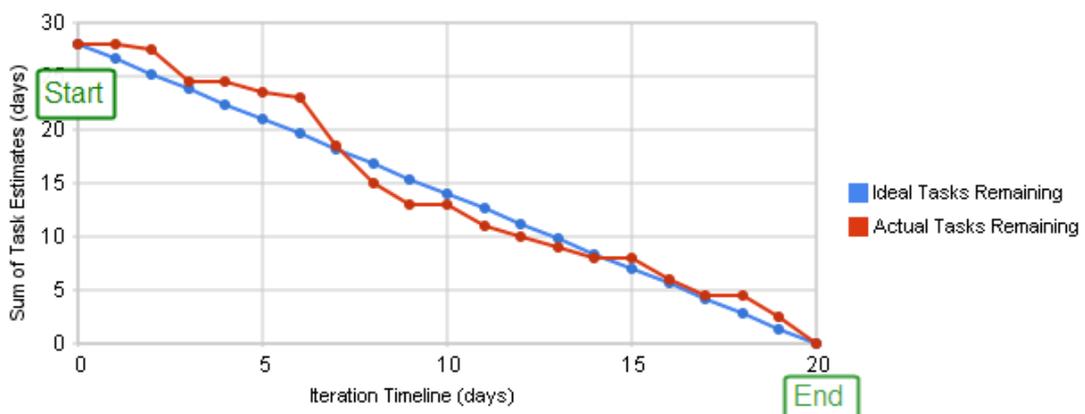


Figura 4.4 Burn-Down Chart [29]

Per tenere traccia del flusso di attività durante uno Sprint, viene utilizzato un Burn-Down Chart che mostra il numero di giorni dello Sprint contro le attività finite, come mostrato in Figura 4.4. Idealmente, questo grafico dovrebbe essere una linea retta (la linea blu in Figura 4.4), che termina con zero attività rimanenti nella giornata finale dello Sprint. In pratica si verificano deviazioni, che vengono catturate nel Burn-Down Chart su base giornaliera rappresentate dalla linea rossa in Figura 4.4.

Il team di sviluppo può quindi facilmente vedere se sono in ritardo con il lavoro pianificato. L'obiettivo della pianificazione è quello di avere un flusso costante di lavoro e la capacità di stimarlo a priori [26].

4.2.2.3.1 Definizione di “Fatto”

La differenza principale nell'applicazione del modello Agile-Stage-Gate ai prodotti fisici piuttosto che ai prodotti software è la definizione di “Fatto”.

Il software, come illustrato nel Capitolo 4.2 per sua natura è quasi infinitamente divisibile, quindi essendo costituito da molteplici funzionalità, lo sviluppo può essere suddiviso in innumerevoli sotto progetti (ad esempio, scrivere migliaia di righe di codice e produrre alcuni schemi) che possono essere completati in un singolo Sprint. Ogni incremento o Sprint produce un prodotto funzionante, anche se limitato. Al contrario, lo sviluppo di un nuovo dispositivo medico, macchina o polimero non può essere facilmente spezzato in modo tale che durante lo sviluppo ci siano degli incrementi di funzionalità. Se il prodotto è un motore o un dispositivo di scansione, non è possibile costruire parte del motore o parte dello scanner e dimostrare che funziona entro poche settimane, né potrà essere rilasciato sul mercato [6], [30].

Le imprese manifatturiere, per superare questo ostacolo hanno apportato delle modifiche alla definizione di “Fatto” descritto nella guida Scrum.

Secondo lo studio condotto da Cooper [30] e Cooper e Sommer [31], un incremento “Fatto” è qualcosa che, anche se non potenzialmente lanciabile sul mercato, mostra incremento eseguito in quello Sprint.

Per esempio possono essere definiti incrementi finiti [30], [31]:

- Un prototipo di prodotto visivo costruito velocemente.
- Un report redatto in un foglio A4, facile da leggere e da mostrare con un proiettore o essere inviato per posta, per essere esaminato da un collega esperto.
- Una check list con diversi elementi, completata da tutti i membri del team che viene approvata dal Product Owner e dal management.

Quando Agile-Scrum viene applicato alle fasi precedenti lo sviluppo del progetto, ad esempio le fasi di concetto e di business, allora la definizione di “Fatto” può includere qualsiasi cosa “Fatta” che possa essere riesaminata da un esperto. Ad esempio, i risultati di uno studio di mercato o un'analisi tecnica di fattibilità sarebbero considerati come uno "Sprint fatto".

Per i prodotti fisici un'altra definizione di “Fatto” possono essere disegni 3D generati dal computer, prototipi virtuali, modelli grezzi, prototipi costruiti velocemente grazie alle stampanti 3D, modelli di lavoro o prototipi precoci. Il risultato di uno “Sprint fatto”,

potrebbe non essere un prodotto funzionante, ma è qualcosa a cui il cliente può fornire un feedback confrontabile con la sua visione di prodotto finale [6].

4.2.2.3.2 Gestione della pipeline di sviluppo dei progetti

Un aspetto positivo dell'Agile-Stage-Gate è la possibilità di avere un eccellente controllo visivo dell'avanzamento delle attività all'interno e nei vari progetti. Le lavagne di lavoro contengono i dettagli di quali attività devono essere fatte durante lo Sprint o l'iterazione, le attività in corso e da chi vengono eseguite. In qualsiasi momento, il Product Owner può entrare in un ambiente di progetto e vedere immediatamente i progressi all'interno dello Sprint, lo stato del Product Backlog e delle priorità nei requisiti [31].

Di maggior interesse è il software capace di creare le lavagne di lavoro. Tale software infatti, non solo consente alle aziende di gestire singoli progetti, ma consente di visualizzare il portfolio completo in tempo reale.

Con tali funzioni, i Manager possono eseguire analisi in tutto il portfolio progetti, ad esempio [31]:

- Revisionare il flusso delle attività, evidenziando ritardi, indicando così la necessità di un'azione immediata.
- Mostrare i tipi di attività svolte da persone e dipartimenti, informazioni utili per l'organizzazione dei team.
- Analizzare la capacità e l'utilizzo delle risorse in tempo reale al fine di migliorare la pianificazione della capacità nel portfolio.

Di conseguenza, in Agile-Stage-Gate, il Management diventa proattivo e agisce con il Project Leader quando si verifica qualche problema e dove potrebbe essere necessario un intervento. Tradizionalmente, il Management risponde in modo reattivo basandosi su dati forniti dai vari Project Manager che non sono in tempo reale e questo può ritardare l'intervento necessario di settimane. Al contrario, con la maggiore visibilità data dall'Agile-Stage-Gate, i problemi vengono immediatamente identificati e si interviene nel minor tempo possibile [31].

A volte si verificano interventi prima che il problema appaia: per esempio, intervenendo quando la velocità del progetto diminuisce o le attività di base sono in ritardo.

4.2.3 Lo Stage-Gate nell'Agile-Stage-Gate

4.2.3.1 Il ruolo dei Gate

Si potrebbe aspettarsi che la definizione di un Gate sia abbastanza rigida, il che contraddice la mentalità Agile di flessibilità. In verità l'Agile è molto coerente con le decisioni Go/Kill sui progetti. La flessibilità del sistema Agile si basa sul prodotto fornito più aderente possibile alle richieste del cliente, non nella struttura decisionale se avanzare o fermarsi con un progetto.

La differenza principale nell'Agile-Stage-Gate rispetto ai tradizionali sistemi Stage-Gate è la capacità di affrontare i progetti più innovativi che presentano alti livelli di ambiguità e incertezza nei requisiti iniziali.

Le incertezze relative ai requisiti dei prodotti vengono gestite accettando e sostenendo i cambiamenti, a condizione che tali modifiche non incidano sul piano generale del progetto, sul budget e sull'attrattività finanziaria del progetto. Se succede, il progetto può essere segnalato, rivalutato ed eventualmente chiuso. Il portfolio prodotti nel modello Agile-Stage-Gate è molto dinamico, infatti il prodotto può essere definito solo al 20% al momento dell'approvazione del progetto (l'intervallo spesso citato è del 20-90% di requisiti definiti all'approvazione del progetto, ma è più tipico 40-60%). Il team di progetto, avendo requisiti poco definiti, può imparare e adattarsi durante il progetto [26]. I Gates continuano a svolgere un ruolo vitale in Agile-Stage-Gate e permettono di avere, oltre ad un punto di controllo qualità, una revisione da parte dei dirigenti e un riesame periodico del progetto per chiudere i progetti deboli e riassegnare le risorse alle migliori iniziative.

Le risorse insufficienti sono infatti la causa dell'allungamento dei tempi nello sviluppo del prodotto [32].

Mentre ogni Sprint ha una propria pianificazione e un tempo flessibile, la linea temporale a lungo termine rimane relativamente stabile. I progetti nell'Agile-Stage-Gate vengono gestiti con timeline o grafici Gantt completi di milestone che definiscono chiaramente il progresso del progetto su un orizzonte temporale prefissato a lungo termine. Certamente queste scadenze sono abbastanza lunghe e i dettagli possono cambiare nel tempo, ma il piano e la timeline a lungo termine rimangono intatti [26].

4.2.3.2 Pianificazione nella metodologia Agile-Stage-Gate

Il problema che hanno riscontrato i manager nell'integrare la metodologia Agile con la metodologia Stage-Gate è come approvare un progetto per lo sviluppo se non si conosce la definizione finale del prodotto e di conseguenza non si dispone di un piano di sviluppo concreto (grafico Gantt) e una stima del costo dello sviluppo. Il Product Backlog in parte copre il ruolo della tradizionale "definizione dei requisiti", afferma Peter Fürst, esperto di sviluppo prodotto presso "Five I's Innovation" in Germania [26]. Il Product Backlog non è congelato ed evolve nel tempo.

All'inizio non è necessario un livello elevato di definizione del prodotto, ma questo deve essere sufficiente per la pianificazione a lungo termine e l'approvazione del progetto.

Come il Product Backlog, anche il piano di progetto evolve nel tempo e pur essendo molto simile ad un piano di sviluppo tradizionale del prodotto (un grafico Gantt), deve necessariamente essere aperto ai cambiamenti. Da questa pianificazione a lungo termine viene determinato un costo stimato e un budget massimo del progetto.

Essendo i requisiti del prodotto e di conseguenza il progetto in continua evoluzione, il costo dello sviluppo diventerà sempre meno incerto con l'avvicinarsi della data del lancio.

4.2.3.3 Analisi dei prerequisiti nella metodologia Agile-Stage-Gate

Cooper [30], descrive come l'utilizzo dell'Agile-Stage-Gate può portare il Team a fare l'errore di non valutare attentamente, nelle fasi precedenti allo sviluppo, i pre-requisiti chiesti dai clienti per lo sviluppo del prodotto. Infatti, gli utilizzatori della metodologia Agile-Stage-Gate, partendo dall'idea che come dice Steve Jobs "La gente non sa cosa vuole finché non l'ha vista" [33], pensando che i requisiti potrebbero cambiare nel tempo, potrebbe reputare questi compiti una perdita di tempo, affermando che possono essere sostituiti con iterazioni e mostrando un incremento finito alla fine dello Sprint.

Al contrario, il sistema Agile-Stage-Gate richiede un'analisi corretta del mercato e dei requisiti del cliente (Voice of Customer) ancora nelle fasi iniziali, per fornire le basi necessarie per il progetto e orientarlo nella giusta direzione. Le richieste del cliente costituiscono la base per lo sviluppo del Product Backlog, che a sua volta porta alla stima dei tempi e del budget per lo sviluppo del prodotto, elementi essenziali per garantire l'approvazione del progetto.

L'Agile-Stage-Gate richiede anche una valutazione tecnica preventiva, anche se la definizione del prodotto può cambiare, in modo che gli sviluppatori siano informati su problematiche tecniche e rischi prima di cominciare la fase di sviluppo.

Un vantaggio di Agile-Stage-Gate è che l'analisi dei prerequisiti tecnici e di marketing non devono essere troppo approfonditi, evitando così un frequente problema riscontrato nei sistemi di sviluppo tradizionali di "paralisi per analisi". I requisiti infatti, verranno rivisti più volte durante lo sviluppo del prodotto, attraverso gli Sprint nelle fasi seguenti, costruendo versioni del prodotto, facendole testare a utenti o clienti, assicurando un feedback regolare da parte dei principali clienti sul mercato.

5 GRANDEZZE UTILIZZATE PER LA CLASSIFICAZIONE

In questo elaborato si propone una classificazione delle metodologie esposte nei precedenti capitoli e dei casi di studio che verranno analizzati, utilizzando due dimensioni qualitative: la rigidità dei Gate e la presenza di feedback da parte dei clienti durante lo sviluppo.

Queste grandezze rappresentano la contrapposizione delle metodologie Agile con le metodologie di sviluppo di prodotto Stage-Gate. Infatti la prima grandezza è caratteristica della metodologia Stage-Gate e la seconda delle metodologie Agile.

5.1 Rigidità dei Gate

I Gate [2] sono importanti per individuare i prodotti che possono aver successo nel mercato o meno. Questo processo, se ben gestito, è finalizzato a concentrare le risorse, specialmente in fase di sviluppo, in progetti che avranno un ritorno una volta lanciati sul mercato.

La grandezza qualitativa è stata divisa in 4 livelli, a seconda della presenza e della rigidità dei Gate all'interno del processo di sviluppo di un prodotto.

Le grandezze sono scaglionate secondo i seguenti criteri:

- **Nulla.** In questo caso, i Gate sono molto flessibili e pressoché inesistenti. Si ha uno sviluppo che, non avendo dei requisiti fissati all'inizio del progetto, non ha delle linee guida definite. Nella maggior parte dei casi lo sviluppo del prodotto termina quando si raggiunge un limite di tempo o un budget fissato. Il bilancio del progetto viene eseguito al lancio del prodotto che conferma il suo successo nella grande distribuzione.
- **Bassa.** In questo livello i Gate sono presenti ma flessibili, non tutti gli Stage hanno un Gate che controlla la corretta esecuzione del lavoro e che revisiona la pianificazione a lungo termine del progetto. Si possono trovare Gate dopo lo Stage di definizione dei requisiti del cliente, dopo i test e dopo il lancio. In questo caso la gestione ed il controllo di alcuni Stage vengono lasciati alla competenza del Team che deve valutare il proprio lavoro e decidere se continuare il progetto.

- **Media.** Questo livello viene assegnato ai processi di sviluppo di prodotto che hanno dei Gate definiti al termine di ogni Stage, con un controllo rigoroso dell'avanzamento, mantenendo una certa flessibilità nel decidere la chiusura di un progetto. In questo caso infatti, i Gate eseguono un controllo della pianificazione a lungo termine del progetto e dell'esecuzione del lavoro. I requisiti per lo Stage successivo sono chiari ma non limitanti per uno sviluppo flessibile.
- **Alta.** In questo caso i Gate sono il cuore del processo di sviluppo di prodotto. A ogni Stage si produce la relativa documentazione che viene analizzata da esperti o manager, i quali decidono l'avanzamento o meno del progetto secondo le potenzialità e il portfolio di progetti a cui la squadra potrebbe lavorare. I Gate, generano poi i requisiti definiti e congelati per lo Stage successivo.

5.2 La presenza di feedback

La presenza di feedback da parte del cliente porta la metodologia di sviluppo di prodotto a creare prototipi/progetti che possano essere mostrati al cliente per avere una risposta per il proseguimento del lavoro [7]. Questi sono importanti per il miglioramento del prodotto e per avere successo nel mercato.

La grandezza qualitativa è stata divisa in 4 livelli, a seconda della presenza di feedback da parte dei clienti all'interno del processo di sviluppo di un prodotto.

Le grandezze sono scaglionate secondo i seguenti criteri:

- **Nulla.** In questo caso il feedback dei clienti durante il processo di sviluppo di prodotto non è presente. In questi progetti, vengono interpellati i clienti solo all'inizio del processo di sviluppo, definendo i prerequisiti del processo secondo le richieste e valutando il lavoro solo quando viene terminato il progetto, con l'uso del prodotto da parte dei clienti.
- **Bassa.** in questo livello si hanno pochi feedback da parte dei clienti. I clienti sono chiamati a dare la loro opinione solo in certi eventi, dopo un certo numero di Stage, solitamente non direttamente ma attraverso altre figure all'interno dell'azienda che fanno da portavoce. Gli sviluppatori devono affidarsi alla propria idea su cosa al cliente possa piacere, seguendo le indicazioni fornite in precedenza.

- **Media.** Le metodologie di sviluppo il prodotto con questo livello di feedback da parte dei clienti, hanno la capacità di coinvolgere gli utilizzatori finali durante il progetto. I clienti vengono coinvolti spesso e viene chiesto loro di fornire feedback per decidere la strada da percorrere nell'avanzamento dello sviluppo. Come nel caso precedente, può esserci una figura che fa da tramite con i clienti e che segue la squadra per tutto lo sviluppo di prodotto. Questa però non si sostituisce ai clienti nel test dei prototipi.
- **Alta** in questo caso, i clienti sono al centro dello sviluppo di prodotto. Viene chiesto loro un feedback continuo, molto importante in quanto senza di esso il lavoro non progredirebbe. Il cliente inoltre ha la possibilità di chiedere di cambiare parti del prodotto già sviluppate perché siano più aderenti alle proprie richieste. In questo modo il prodotto risultante sarà completamente aderente alle richieste del cliente e di successo sul mercato.

5.3 Classificazione dei modelli presenti in letteratura

Usando queste dimensioni, si possono classificare le tre metodologie esposte nei precedenti capitoli:

5.3.1 Modello Stage-Gate

Questo modello, come descritto nel Capitolo 1, se applicato e ben gestito porta ad avere dei Gate molto rigidi con un riesame del lavoro eseguito durante lo Stage e una decisione se continuare o chiudere il progetto. Questo, come descritto sopra, riesce a focalizzare le risorse, sempre scarse, nei progetti più importanti che saranno più proficui per l'azienda. Questo modello ha dei limiti, come descritto nel Capitolo 3.2, quali la definizione a priori dei requisiti dei clienti e questo comporta l'averne un feedback sul lavoro svolto solo al termine dello sviluppo del prodotto, poco prima del lancio del prodotto con il feedback dei Lead User [3] o subito dopo con il feedback del mercato.

Questo modello può essere classificato con le grandezze proposte riportate nella tabella sottostante:

Tabella 5.1 Classificazione della metodologia Stage-Gate

Rigidità dei Gate	Alta
Presenza di feedback	Nulla

5.3.2 Modello Agile (Scrum)

Il modello Agile, come descritto nel Capitolo 2.1, è un modello di sviluppo di prodotto adattivo che riceve frequenti feedback da parte dei clienti fino al lancio sul mercato. Questo porta, alla fine dello sviluppo, ad avere un prodotto molto aderente alle richieste del cliente. Il limite di questo modello è che fissando il tempo di sviluppo del prodotto e il budget, non si hanno punti decisionali se il progetto deve essere portato a compimento o può essere chiuso. Questo può essere deleterio per l'azienda, se è impegnata in vari progetti, in quanto le risorse sono impiegate, a volte monopolizzate, in certi progetti che non avranno poi un risvolto positivo sul mercato.

Questo modello può essere classificato secondo le grandezze proposte:

Tabella 5.2 Classificazione della metodologia Agile-Scrum

Rigidità dei Gate	Nulla
Presenza di feedback	Alta

5.3.3 Modello ibrido Agile-Stage-Gate

Questo modello, come descritto nel Capitolo 4, integra le migliori caratteristiche dei modelli per lo sviluppo di prodotto: Agile e Stage-Gate.

Tra le caratteristiche di questa metodologia si possono individuare:

- La presenza di Gate ben definiti, in cui i progetti vengono esaminati e valutati per evitare il proseguimento di quelli che non portano valore all'azienda.
- La presenza di feedback da parte dei clienti (o del Product Owner) che guidano lo sviluppo del prodotto in modo che esso sia aderente alle esigenze dei clienti nel momento prossimo al lancio.

Come descritto nel Capitolo 4, i progetti per cui si usa questo modello di sviluppo di prodotto hanno una certa incertezza, per questo, per progredire nello sviluppo, i Gate devono avere una certa flessibilità.

A differenza dei software, i prodotti tangibili non possono raccogliere feedback da parte dei clienti e modificare il prodotto fino a qualche momento prima del lancio. Nell'hardware le modifiche dello stesso vengono congelate progressivamente durante lo sviluppo per riuscire a completare la fase di produzione prima del lancio.

Questo modello può essere classificato con le grandezze proposte riportate nella tabella sottostante:

Tabella 5.3 Classificazione della metodologia Agile-Stage-Gate

Rigidità dei Gate	Media
Presenza di feedback	Media

5.4 Matrice Rigidità dei Gate/Presenza di feedback

Avendo definito le grandezze, è possibile quindi costruire una matrice Rigidità dei Gate vs Presenza di feedback da parte dei clienti. Secondo la precedente classificazione, si possono posizionare le metodologie:

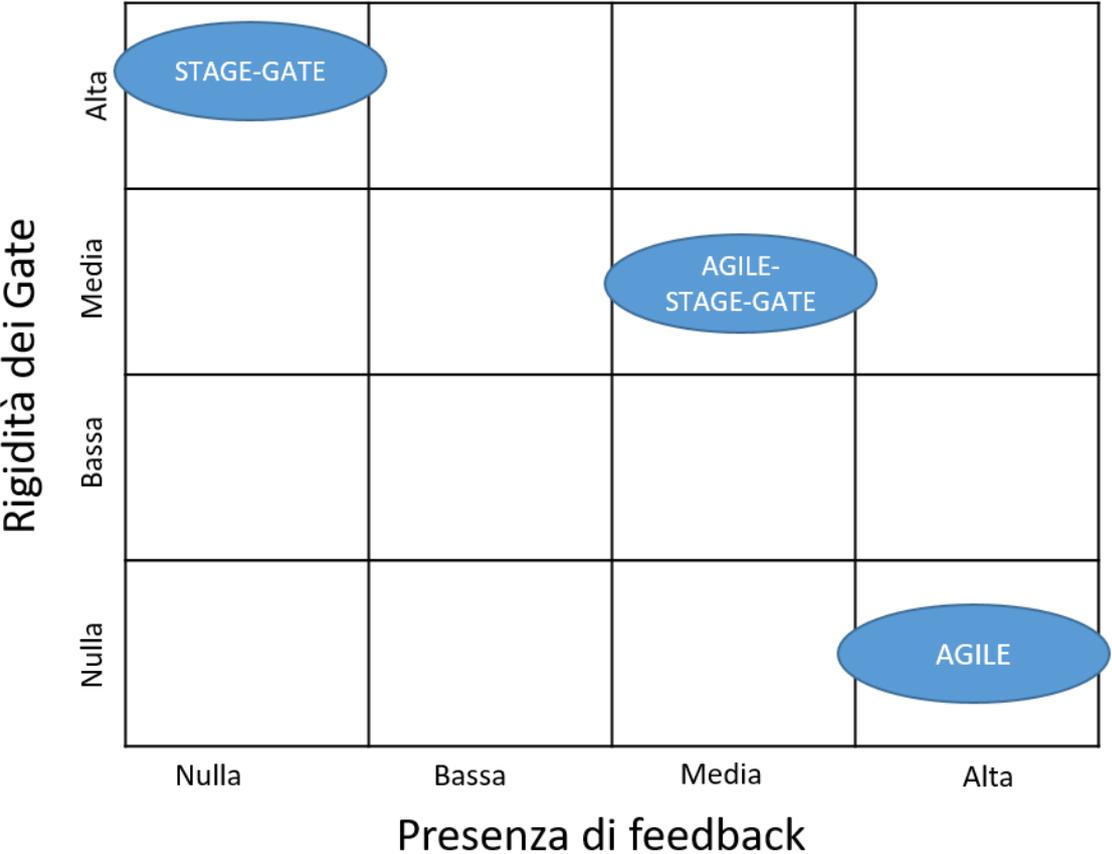


Figura 5.1 Metodologie di letteratura nella matrice Rigidità dei Gate v.s. Presenza dei feedback da parte dei clienti

6 CASI DI STUDIO

6.1 Caso Wikispeed

Wikispeed è stata fondata da Joe Justice [34], un consulente software del settore Seattle appassionato di automobili e di sostenibilità ambientale.

Nel 2008, Progressive Insurance X ha annunciato il premio da 10 milioni di dollari per chi sarebbe riuscito a costruire un'automobile da 100 miglia per gallone che soddisfacesse le specifiche legate alla sicurezza stradale. Joe Justice decise di concorrere alla competizione.

Iniziò in solitaria ma, attraverso i social network, alcuni volontari vennero a conoscenza del suo progetto e si unirono a lui. Tre mesi dopo, il suo team era composto da 44 membri in 4 paesi diversi e svilupparono un prototipo funzionante che venne iscritto alla competizione X-Prize.

Nonostante i pochi fondi e l'esperienza limitata nella produzione di autovetture, Wikispeed è arrivata al decimo posto nella classe mainstream, davanti a più di cento vetture di aziende e università ben finanziate in tutto il mondo. Wikispeed non è riuscita a vincere il concorso, ma il suo caso può anche essere visto come un'innovazione della produzione Agile [34].

Il prototipo di Wikispeed non è attualmente un rivale serio per le grandi compagnie automobilistiche. Il suo prototipo aveva dei limiti: non aveva ancora porte, non era resistente alle intemperie e lo spazio per il passeggero era stretto.

La versatilità dell'autovettura proviene dall'architettura modulare con un'ottima definizione delle interfacce in modo da poter sviluppare separatamente i componenti. Wikispeed è una società automobilistica legalmente registrata con prototipi funzionanti che sta sviluppando per il mercato [34].

Wikispeed non ha rilasciato solo un prototipo di automobile funzionante che può essere acquistato, ma ha voluto aderire ad un progetto hardware open source in cui è possibile trovare le istruzioni per costruire l'automobile nel garage di casa [35].

È possibile infatti con 14500\$ e 100 ore di lavoro, senza particolari conoscenze ed abilità, costruire un'automobile funzionante. Questo è possibile grazie alla metodologia di sviluppo di prodotto e all'Extreme Manuctory.



Figura 6.1 Progetto del prototipo di Wikispeed [36]

6.1.1 L'organizzazione

Wikispeed è diventato un Team di 140 volontari che lavorano con risorse finanziarie molto limitate, spesso anche la notte e il fine settimana.

Il punto di forza del gruppo di sviluppo è l'auto-organizzazione, infatti per coordinare tutti questi sviluppatori sono state create delle squadre che collaborano e si organizzano per portare a termine un lavoro [34].

Per coordinare questi gruppi, si sono dovuti usare mezzi di comunicazione chiari come ad esempio la chiamata collettiva in videoconferenza o un video caricato su YouTube, per mostrare lo stato di avanzamento del lavoro [34].

La metodologia Agile ha aiutato il processo di sviluppo di Wikispeed anche nella formazione degli sviluppatori. Utilizzando l'Extreme Program e in particolare il Pair Program, il lavoro viene eseguito contemporaneamente da due persone. Per le metodologie tradizionali questo è uno spreco in quanto vengono utilizzate due persone per svolgere il lavoro di una sola. In verità, questa pratica velocizza l'innovazione e quindi è in ultima analisi meno costosa. Inoltre riduce i tempi di formazione che non producono valore e diminuisce drasticamente anche la necessità di documentazione, perché le persone che eseguono il lavoro condividono regolarmente la conoscenza [34].

6.1.2 Lo sviluppo di prodotto

Per avere uno sviluppo veloce e un maggior focus sulle richieste del cliente, Wikispeed ha adottato la metodologia Agile. L'idea è di sviluppare una vettura utilizzando brevi cicli, per fornire funzionalità o prodotti che soddisfano i requisiti. Questo lo si può ottenere cercando di individuare cosa i clienti vogliono, definendo quei requisiti in termini di test precisi e articolati, dando delle priorità ai test che devono essere elaborati, lavorando in brevi cicli per fornire funzionalità o prodotti che soddisfano i requisiti [34]. In questo caso di studio la metodologia più utilizzata è lo Scrum.

Per adattarsi meglio a questa metodologia di sviluppo di prodotto, Wikispeed ha scelto per la propria autovettura un design modulare del prototipo che consente di adattarsi e svilupparsi rapidamente. Ad esempio, il motore può essere commutato da un motore benzina a un motore elettrico secondo le esigenze dei clienti in circa il tempo necessario per cambiare un pneumatico. La semplicità e la modularità riducono anche i costi in utensili e macchine [34].

6.1.2.1 Framework di sviluppo prodotto

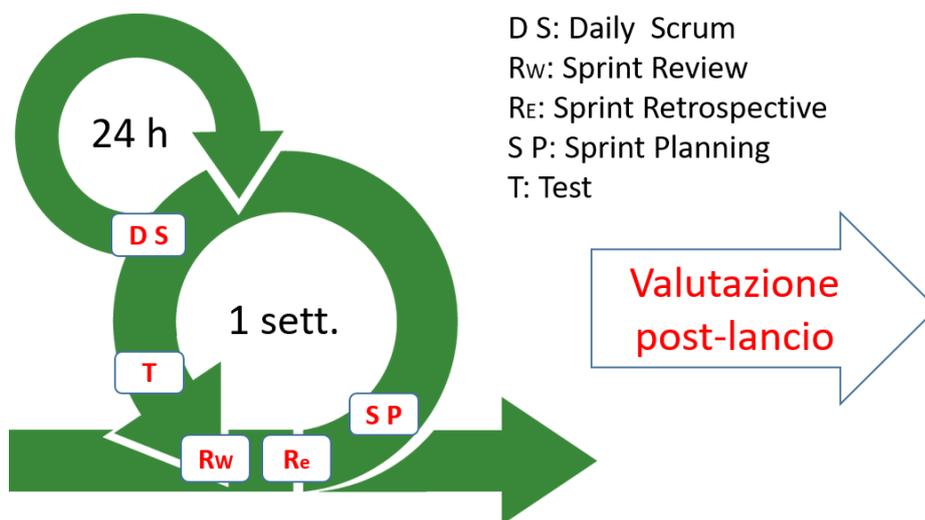


Figura 6.2 Framework di sviluppo di prodotto in Wikispeed

Lo Scrum in questo caso di studi è composto da Sprint della durata di una settimana ciascuno, nella quale i Team di sviluppo implementano i requisiti del Backlog nel prodotto e alla fine viene eseguito un test, definito al momento della pianificazione, riguardante il successo dello Sprint appena eseguito [34].

Una parte molto importante nel processo di sviluppo di prodotto è la pianificazione degli Sprint. L'idea è di eseguire un lavoro che tra quelli disponibili possa portare maggior valore al prodotto. Si utilizza uno strumento online di gestione del Backlog chiamato Scrumy, [37] nel quale i membri del Team si assegnano i compiti presenti nel Backlog [34].

All'interno di uno Sprint, si utilizzano lavagne Kanban per ottimizzare il flusso di lavoro. Ogni gruppo di sviluppo ha la propria lavagna Kanban che viene sincronizzata con lo strumento online almeno settimanalmente [34].

La riunione settimanale di Sprint Review viene effettuata in videoconferenza in modo che tutti possano partecipare. In questo meeting la maggior parte delle persone parla per circa un minuto, alcune prendono parola solo quindici secondi. Questa riunione dura complessivamente trenta minuti [34].

La pianificazione dello Sprint successivo viene effettuata per la prima parte in videoconferenza, mentre la pianificazione in dettaglio viene effettuata off-line. In questa seconda parte, gli sviluppatori di ogni gruppo di lavoro iniziano a inserire le attività dalla colonna "Da Fare" della lavagna Kanban. Di solito questa fase dura circa un'ora, inclusa la video-conference e la risoluzione dei problemi rimanenti.

Importanti nello Sprint per lo sviluppo di Wikispeed sono la fase di test per verificare l'aderenza del prototipo alle norme vigenti sulla sicurezza stradale ed i risultati ottenuti nell'aumento dell'efficienza, diminuzione delle emissioni ed aumento del confort.

Nella pianificazione settimanale si ha una definizione dei test da superare perché il prototipo si avvicini maggiormente alla visione del Product Owner [34].

Il test inoltre deve fornire chiari criteri di successo, in modo che ogni squadra o coppia nel team sappia l'esito del proprio lavoro e quando l'obiettivo d'implementazione della funzionalità sarà raggiunto. Questo è indispensabile se si hanno dei Team grandi, per consentire al Team di concentrarsi su lavori più piccoli, più facili e con problemi più semplici da risolvere. Le norme di circolazione sono state di grande aiuto nel test-driven, in quanto i regolamenti sono descritti come test. Questo produce un valore numerico e univoco su cui lavorare [34].

Una particolarità del framework utilizzato in Wikispeed non riportato nella guida Scrum [9], è il ruolo del Product Owner. Questo ha la responsabilità di definire le priorità tra i possibili requisiti su cui i team potrebbero lavorare in quello Sprint, quindi la persona che dovrebbe ricoprire quel ruolo è chi, a quel punto del processo di sviluppo, ha la visione più chiara del prodotto. In Wikispeed, il ruolo del Product Owner è spesso assunto dal

fondatore, ma non sempre. Dipende da ciò che è importante in un determinato Sprint. Ad esempio, se il team sta lavorando sull'aerodinamica in quello Sprint, il Product Owner è chiunque nella squadra in quel momento è più esperto sull'aerodinamica, così ha la più chiara visione del lavoro che deve essere fatto [34].

6.1.3 Vantaggi e sfide

6.1.3.1 Vantaggi

Un vantaggio dell'utilizzo della metodologia Agile in Wikispeed è l'auto-organizzazione del lavoro, che aiuta ad aumentare il morale e la velocità del team. Il risultato è stato una comunicazione più chiara e una maggiore velocità, infatti nella riunione di gruppo settimanale, i membri dicono cosa hanno fatto la settimana precedente, cosa dovranno fare la settimana successiva e discutono su eventuali problemi che potrebbero bloccare il progredire dello sviluppo [34].

Un altro punto di forza dell'uso di questo framework è la collaborazione con i clienti per dare una priorità alle attività che dovranno essere eseguite nei prossimi Sprint per dare maggior valore al prodotto. Per far questo si continua a chiedere ai Lead User e ai micro investitori cosa piace loro dell'auto, cosa non piace e quale caratteristica vogliono che venga implementata [34].

L'approccio di Wikispeed nello sviluppo del prodotto aumenta la velocità di risoluzione dei problemi. Ad esempio, dopo poche ore dalla ripresa di un video di un test d'impatto laterale, la squadra in tale occasione ha capito che anche se gli standard imposti dalle normative erano rispettati, non risultava il top della categoria; ha provveduto così in poche ore ad aggiungere una protezione laterale anti impatto sull'autovettura. La prima volta che Wikispeed ha fatto un'iterazione per la sicurezza come quella sopracitata, ha impiegato molte settimane per risolvere il problema. Ora è in grado di farla entro un ciclo di Sprint di sette giorni. Questo si ha perché tutti i membri condividono le loro informazioni [34].

A differenza di una tradizionale burocrazia gerarchica, Wikispeed abbraccia i valori di trasparenza radicale e miglioramento continuo. Ognuno può vedere in qualsiasi momento quali sono gli obiettivi generali e come il lavoro s'inserisce in un quadro più grande. Chiunque può fare suggerimenti in qualsiasi momento su qualsiasi lavoro. Qualsiasi

problema può essere segnalato durante la riunione settimanale e viene risolto in maniera trasparente. Di conseguenza, anche se la squadra è sparsa in tutto il mondo, ognuno può contribuire a identificare e risolvere i problemi. In Wikispeed c'è un'idea fondamentale: il prodotto può sempre essere migliorato in modo da aggiungere più valore ai clienti. Infatti la mentalità principale di che lavora è pensare quali possano essere le attività che daranno il massimo valore per i clienti e affrontarle [34].

Le comunicazioni verticali, proprie della gerarchia tradizionale, consistono nello svolgere i compiti a loro assegnati. In Wikispeed, invece, sono peer-to-peer. Infatti in qualsiasi momento c'è un Product Owner, che è responsabile nel definire le priorità sulle attività su cui bisognerebbe lavorare, ma quella persona non è un capo convenzionale in quanto il team può chiedere in qualsiasi momento spiegazioni relative alle scelte che egli ha deciso, o criticarle. Inoltre al Product Owner viene spesso chiesto consiglio su come procedere nelle attività, in quanto lui ha una visione chiara dello sviluppo del prodotto [34].

6.1.3.2 Sfide

La sfida principale in Wikispeed è il coordinamento di un gruppo così grande di sviluppatori. Il Team supera i 100 sviluppatori, ciò è una violazione alla regola imposta nella guida Scrum. Questo è stato risolto, come scritto in precedenza, utilizzando Sprint brevi e una comunicazione chiara e trasparente [34].

Un'altra sfida alla costruzione del prototipo di Wikispeed è l'integrazione di tutti i moduli dell'autovettura sviluppati in posti differenti del mondo, che devono essere testati sia separatamente che nell'automobile completa [34].

In Wikispeed, utilizzando un modello non Plan-Driven, non sono stati fissati né il budget, né le scadenze per lo sviluppo del prodotto. Infatti l'autovettura alla presentazione del prototipo all'X-Prize, era incompleta, non resisteva alle intemperie e aveva poco spazio per il passeggero.

6.1.4 Classificazione del caso di studio

In questo caso di studi i Gate sono inesistenti non si nota quindi una valutazione del Team di sviluppo da parte di altri esperti in quanto l'organizzazione è completamente

autogestita. Questo porta a definire un budget e una tempistica che deve essere rispettata.

Il controllo può essere fatto solo alla fine del progetto.

Si notano numerosi feedback da parte dei clienti che sono al centro del processo di sviluppo. Questo, oltre ad accelerare il processo, fa in modo di avere un prodotto, alla fine del progetto, aderente alle richieste dei clienti.

Secondo le grandezze scelte, il caso di studio può essere classificato:

Tabella 6.1 Classificazione del caso di studio Wikispeed

Rigidità dei Gate	Nulla
Presenza di feedback	Alta

6.2 Caso ThermoValves

Il caso di studio analizzato da Cooper e Sommer [26] illustra l'esperienza della società ThermoValves (nome di invenzione) che è un'unità commerciale all'interno di una grande società che fornisce soluzioni per il riscaldamento, mirate al settore delle costruzioni residenziali. L'azienda è un produttore mondiale con sede in Europa, con vendite annuali di 7 miliardi di dollari ed una spesa per ricerca e sviluppo pari al 4% delle vendite. Lo sviluppo di soluzioni per il riscaldamento è altamente innovativo, e comporta sia lo sviluppo di hardware che di software utilizzando nuove tecnologie.

6.2.1 Lo sviluppo di prodotto

La sperimentazione dell'implementazione dell'Agile nel processo Stage-Gate si ha nel 2015 [26].

Prima si adottava un processo di sviluppo di prodotto Stage-Gate molto simile al processo descritto da Cooper nel 1990 [1]. Questo era composto da Stage chiari, milestone e Gate ben definiti dove il progetto veniva revisionato e si decideva se proseguire, reiterarlo o chiuderlo a seconda che i prerequisiti fossero stati implementati o meno nel prodotto sviluppato.

Si è deciso di implementare l'Agile nel dipartimento di sviluppo di prodotti fisici, in particolare lo Scrum, da un'esperienza positiva all'interno del dipartimento software. L'azienda ha visto nell'Agile una soluzione ai rischi derivati dal mercato. Infatti c'era la consapevolezza che le esigenze del mercato stavano cambiando velocemente a causa di una maggiore concorrenza globale, richiedendo per lo sviluppo di prodotti una frequente conferma delle esigenze del mercato per garantirne il successo. Per l'implementazione sono stati destinati dei fondi e un Team completamente dedicato al progetto [26].

6.2.1.1 Framework di sviluppo di prodotto

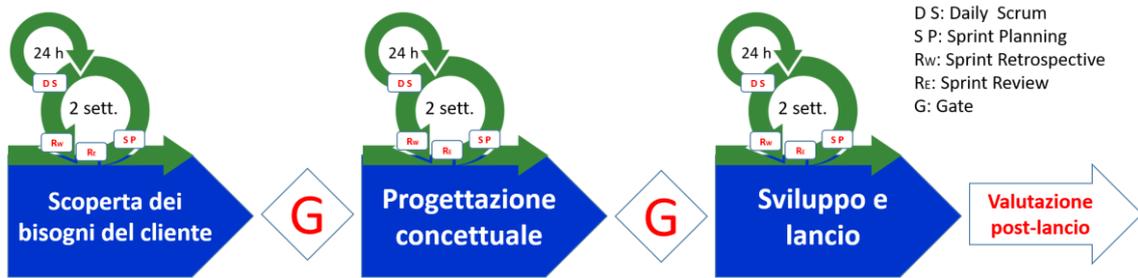


Figura 6.3 Metodologia di Sviluppo in ThermoValves

ThermoValves ha deciso di integrare la metodologia Scrum nella preesistente metodologia Stage-Gate ottenendo una struttura di macro planning con Stage definiti e Gate alla fine di ogni fase di lavoro e una struttura di micro planning con iterazioni e continue ripianificazioni a livello operativo ogni 2 settimane. Durante ogni Stage si ha una validazione rapida delle implementazioni nei prodotti fisici e virtuali da parte dei clienti. La metodologia Agile-Stage-Gate viene applicata fin dalla prima fase del progetto per tutto il processo di sviluppo di nuovi prodotti per ogni Stage [26].

Per evitare incomprensioni sui ruoli manageriali, il Product Owner, per questo progetto, ha deciso di mantenere il titolo di Project Manager. Questa decisione è stata presa per non creare confusione e mantenere la stessa dicitura che aveva in altri progetti a cui lavorava [26].

6.2.1.1.1 Framework a livello di macro planning

La modifica principale alla metodologia Stage-Gate sta nella diversa fissazione dei prerequisiti stabiliti prima della fase di progettazione. Infatti per avere un processo di sviluppo di prodotto più flessibile per permettere di rispondere alle necessità del cliente, i prerequisiti non possono più essere congelati all'inizio dello sviluppo. Le specifiche di progettazione non sono più fissate in anticipo, ma sono continuamente adattate attraverso le iterazioni di progettazione. Questo porta ad avere Gate che, mentre nel processo tradizionale controllavano la presenza dell'implementazione dei prerequisiti alla fine dello Stage, ora sono molto flessibili per permettere al progetto di progredire anche con una certa incertezza sui requisiti che il prodotto deve avere. I Gate comunque sono pensati per essere basati sul risultato di frequenti test utente, i quali forniscono una migliore

comprensione dell'accettazione del prodotto da parte del cliente e del successo finale del prodotto [26].

In questo modo, il sistema Stage-Gate mantiene invariate le sue caratteristiche a livello di pianificazione del progetto.

I requisiti e la possibilità di modifica dei prodotti per renderli più aderenti alle richieste del cliente nell'avanzare attraverso lo sviluppo del prodotto, vengono progressivamente congelati e i Gate diventano sempre più simili ai Gate tradizionali. Questo è dovuto alla necessità di avere un progetto definito nel momento della produzione a causa dei tempi lunghi di fabbricazione [26].

6.2.1.1.2 Framework a livello di micro planning

La metodologia Scrum viene applicata nella metodologia Agile-Stage-Gate a livello operativo. In questo caso di studi, gli Sprint dello Scrum durano due settimane e sono impiegati per eseguire sia gli sviluppi di progettazione che le convalide dei clienti.

Tutti i componenti del Team, seguendo le metodologie Agile, lavorano nello stesso locale utilizzando tecniche visive di avanzamento, come lavagne Kanban sia fisiche che digitali. I Daily Scrum sono molto usati e la presenza dello Scrum Master come moderatore della riunione ne facilita lo svolgimento [26].

Il Backlog fornito dal Product Owner, non è usato come elenco con delle priorità rigide, ma come elenco flessibile contenente i punti senza ordine e senza priorità.

Le Customer Review sono state condotte regolarmente, ma non come parte dello Sprint Review alla fine di ogni sprint: esso è considerato un compito talmente importante da essere stato assegnato come tema principale di alcuni Sprint, e i risultati sono stati presentati allo Sprint Review [26].

6.2.2 Vantaggi e sfide

6.2.2.1 Vantaggi

Il maggior vantaggio che si è ottenuto nell'implementazione dell'Agile nel processo di Stage-Gate è la maggior soddisfazione delle esigenze del mercato attraverso le validazioni dei clienti [26].

Un vantaggio è stato rappresentato dalle Customer Review. Queste infatti venivano viste non come un'attesa, come accadeva nelle metodologie tradizionali, ma come un vero e proprio Sprint, permettendo test più approfonditi e iterazioni tra le varie versioni di progetto.

L'interazione del Team di sviluppo con gli stakeholder interni (Management) che normalmente forniscono input per le specifiche del prodotto, ha portato notevoli vantaggi. Con la metodologia tradizionale il management doveva definire a priori tutte le specifiche di prodotto, mentre ora i requisiti seguono l'evolversi degli input dai clienti e vengono definiti in collaborazione tra il management e il Team di sviluppo durante le iterazioni di progettazione [26].

6.2.2.2 Sfide

La sfida più grande dell'integrazione della metodologia Agile è stata superare la frustrazione degli sviluppatori che erano abituati ad avere le specifiche di prodotto congelate all'avvio della progettazione. Alcuni preferivano restare ad aspettare di risolvere da soli un problema, piuttosto che coinvolgere persone al di fuori della squadra e iterare il processo [26].

6.2.3 Classificazione del caso di studi

In questo caso di studi si hanno dei Gate ben definiti alla fine di ogni stage, che decidono se il progetto deve continuare o meno. Questi Gate anche se seguono le logiche Go/Kill, sono abbastanza flessibili e ammettono una revisione della pianificazione a lungo termine.

Il feedback da parte dei clienti è costante e abbondante in ogni Stage. Questo non permette però di modificare la pianificazione seguendo le esigenze attuali dei clienti. Infatti i requisiti vengono congelati nel progredire dello sviluppo e se le esigenze dei clienti cambiano poco prima del lancio non è possibile modificare il prodotto.

Questo può essere classificato nella matrice rigidità dei Gate vs presenza dei feedback dei clienti con le grandezze riportate in tabella:

Tabella 6.2 Classificazione del caso di studio ThermoValves

Rigidità dei Gate	Media
Presenza di feedback	Media

6.3 Caso Siemens Westinghouse Power Generation

Siemens Westinghouse Power Generation [4] progetta e produce turbine che generano elettricità nelle centrali elettriche. La società ha 26000 dipendenti e un fatturato netto di 8 miliardi di dollari all'anno. Questo caso è focalizzato sullo sviluppo di turbine a gas o a combustione usati nei cicli semplici o combinati. Tale settore serve principalmente società di servizio pubblico e il mercato è oligopolista.

Siemens e Westinghouse sono ex rivali. Le strategie a lungo termine per l'innovazione e la tecnologia dell'intero settore della turbina, sono state influenzate molto dall'introduzione di standard imposti dal governo americano, che lo reputa un settore strategico per il paese. Cinquant'anni di concorrenza e innovazione hanno portato ad un mercato maturo e ad un accento sullo sviluppo industriale per aumentare l'efficienza. Le due società si sono fuse quando Siemens ha acquisito Westinghouse Power Generation dalla società madre (CBS) nel 1998 [4].

Le turbine a combustione sono grandi macchine che devono essere progettate con tolleranze di pochi micron a causa della piccola distanza delle loro lame e palette. Le turbine a gas possono essere fornite come impianti a diverso grado di completezza, da "Econopacs", a "isole di potenza" a "impianti finiti". Gli Econopacs comprendono solo le unità turbina con i loro sistemi di combustibile e di controllo. Le isole intermedie comprendono generatori di vapore per il recupero di calore e altre apparecchiature per la produzione di energia a partire dal calore di scarto della turbina a gas. Infine, le centrali termoelettriche complete comprendono tubazioni di bilanciamento, pompe, torri d'acqua, trasformatori e distributori [4].

A differenza delle isole di potenza, le singole turbine non sono generalmente modulari, nonostante alcuni tentativi dell'azienda di modulare alcuni componenti per uso comune tra le diverse famiglie di turbine. La maggior parte delle turbine sono vendute come pacchetti e la maggior parte dei componenti della turbina sono unici per ogni singola tipologia di turbina prodotta [4].

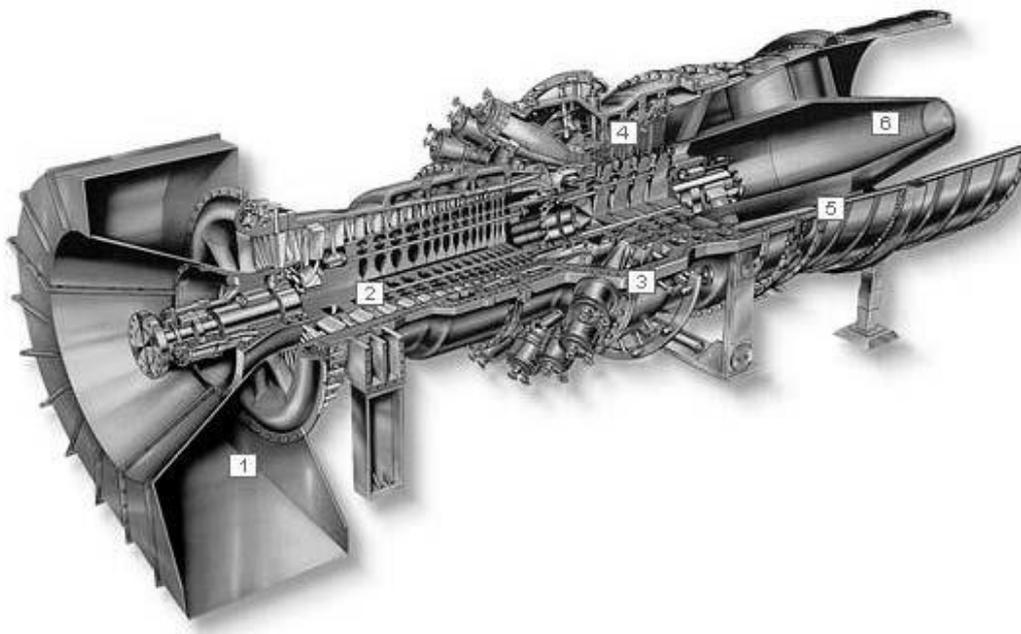


Figura 6.4 Turbina a gas Siemens-Westinghouse 501G [4]

Un esempio di una singola turbina di combustione, il 501G, può essere visto in Figura 6.4. La 501G è la più grande della famiglia 501 di turbine e può generare 253 MW ad una efficienza massima del 39% come turbina a ciclo unico [4].

In combinazione con una turbina a vapore a valle come parte di una centrale a ciclo combinato, l'efficienza termica complessiva può raggiungere quasi il 60%.

Le turbine a gas sono nate negli anni '40 come spin-off dei motori a reazione dei jet militari e hanno avuto una grande diffusione dopo il blackout avvenuto in America negli anni '60 [4].

La principale caratteristica tecnica che decreta il valore di una turbina è l'efficienza termica. I principali fattori su cui gli ingegneri possono agire sono i materiali delle palette e il raffreddamento delle stesse.

Per il primo fattore il problema principale che i materiali devono affrontare è la dilatazione termica, l'ossidazione e il creep. Ora le palette vengono costruite in leghe di cobalto, facili da lavorare, saldare e su cui fare manutenzione. Altri invece usano leghe di titanio con caratteristiche superiori ma molto costose. Le ricerche tecnologiche ora sono rivolte ai materiali ceramici [4].

Come i materiali, anche le tecniche di raffreddamento delle palette hanno subito evoluzioni nella storia. L'efficienza è determinata anche dalla temperatura di combustione dei gas. I materiali usati hanno però una certa resistenza termica, quindi per essere usati a temperatura superiore devono essere raffreddati. Dagli anni '70 si è

sviluppato un sistema di raffreddamento che si è evoluto fino a portare alle efficienze odierne [4].

Un ruolo fondamentale nello sviluppo tecnico delle turbine l'ha avuto il governo americano. Questo ha speso 13 miliardi di dollari negli anni '40 e 400 milioni di dollari all'anno per la ricerca e sviluppo attraverso aziende come General Electics e Pratt & Whitney. Ciò ha portato ad un "supermercato di tecnologia" che le aziende possono utilizzare dalle leghe per le palette ai sistemi di raffreddamento [38]. Gli anni '70 hanno visto la sponsorizzazione del Dipartimento dell'Energia americana nel programma di tecnologia della turbina ad alta temperatura e la sponsorizzazione del governo giapponese del suo progetto nazionale Moonlight con Mitsubishi per migliorare le turbine. Un'iniziativa più recente è il programma ATS (Advanced Turbine Systems), guidato dal Federal Energy Technology Center che include sei produttori di turbine statunitensi, 83 università e centri di ricerca del Dipartimento dell'Energia Americana [4].

L'obiettivo è di sovvenzionare e coordinare la ricerca e sviluppo per portare alla creazione di una nuova generazione di efficienti turbine a ciclo combinato.

L'intervento del governo ha introdotto degli standard nella costruzione delle turbine a gas che non ha però influenzato troppo i design delle diverse case. Il design dominante è una serie di bruciatori disposti ad anello intorno alla turbina stessa. Le imprese americane hanno potuto impostare lo standard di progettazione dominante perché avevano beneficiato di finanziamenti di ricerca militare sponsorizzati dal governo. Tra le varie compagnie beneficianti di questi finanziamenti c'erano General Electric e Westinghouse [4], [38].

Questo ha portato a una elevata concorrenza focalizzata principalmente sull'efficienza ed il prezzo.

6.3.1 Lo sviluppo di prodotto

Il rischio più importante nello sviluppo della turbina è il rischio tecnico, in parte perché il rischio di mercato è ridotto da contratti preliminari e da un sistema di risarcimento dei danni [4].

Una garanzia di risarcimento danni è diversa da una garanzia standard. Le normali garanzie sono obblighi di riparazione o sostituzione di parti danneggiate o inadeguate per un determinato periodo di tempo. Tuttavia, le turbine intere sono troppo grandi, troppo

poco modulari e troppo costose per essere sostituite interamente in caso di sotto performance, come ad esempio un'efficienza minore [4].

Le garanzie del risarcimento danni consistono in pagamenti al cliente per compensare qualsiasi perdita subita da basse efficienze.

Il risultato di questo sistema contrattuale è un congelamento preliminare delle specifiche dei clienti. I dipartimenti di vendite e marketing promettono elevate efficienze ai clienti e queste sono le specifiche assegnate nella fase di sviluppo del prodotto [4].

Le specifiche dei requisiti congelati portano a un rischio limitato di mercato, ma a un elevato rischio tecnico.

Il sistema di sviluppo di prodotto di Siemens-Westinghouse Power Generation è stato attuato poco dopo che Westinghouse è stato acquistato da Siemens. Fino ad allora, la società ha utilizzato un processo di Stage-Gate meno formale e spesso impegnato in prodotti unici progettati su richiesta specifica del cliente. La fusione ha portato al lancio di una nuova metodologia di sviluppo di prodotto più formale ed ad una maggiore focalizzazione sulle famiglie di prodotti e sulle piattaforme [4].

Il processo di sviluppo prodotto in Siemens Westinghouse Power Generation è una metodologia Stage-Gate, con Stage definiti e Gate molto rigidi.

6.3.1.1 Framework di sviluppo di prodotto

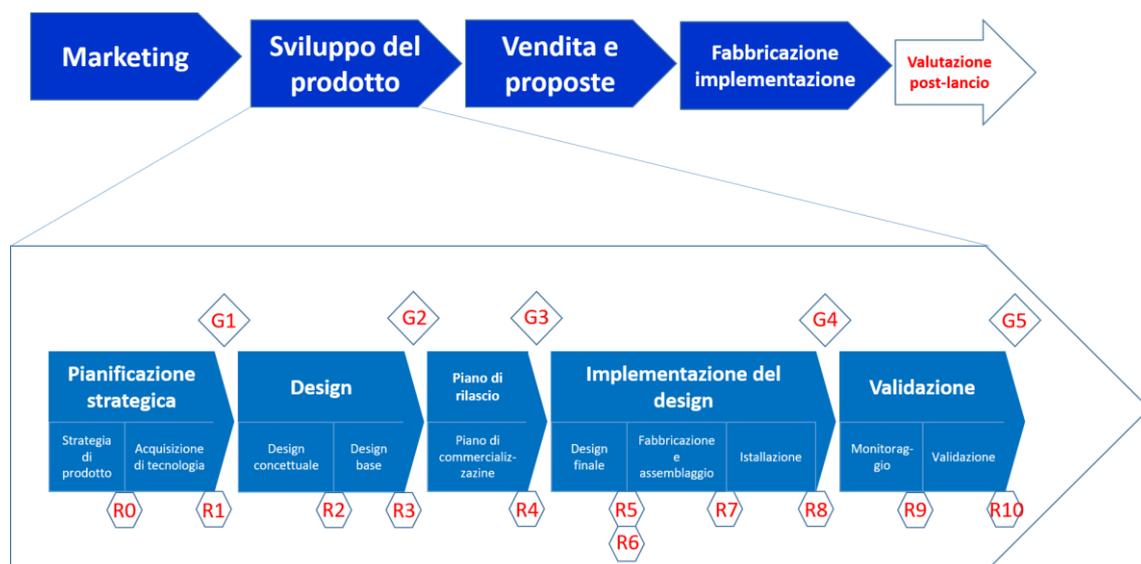


Figura 6.5 Metodologia di sviluppo di prodotto in Siemens-Westinghouse Power Generation [4]

Tabella 6.3 Elenco dei Gate e delle revisioni nello stadio di sviluppo prodotto in Siemens Westinghouse [4]

G1	Iniziazione del programma	R0	Revisione della strategia di prodotto
		R1	Revisione dei prerequisiti di prodotto
G2	Progettazione	R2	Revisione delle specifiche di progettazione
		R3	Revisione della progettazione
G3	Rilascio del prodotto	R4	Revisione del piano di commercializzazione
		R5	Revisione del progetto finale
		R6	Revisione dei fornitori
		R7	Revisione del prodotto
G4	Rilascio della serie	R8	Revisione della commissione
		R9	Revisione del monitoraggio del prodotto
G5	Chiusura del programma	R10	Revisione delle performance

Il processo di sviluppo di prodotto è illustrato in Figura 6.5. Essendo una metodologia molto formale e rigida, deve essere seguita nei minimi dettagli.

In Figura 6.5 e in Tabella 6.3 si può notare in dettaglio la sequenza degli stadi all'interno dello Stage di sviluppo del prodotto. La metodologia permette di avere un controllo massimo sul lavoro e sulla pianificazione dello sviluppo del prodotto grazie a Gate e revisioni rigidamente fissate dopo ogni stadio [4].

I Gate sono controllati esclusivamente da manager e sono punti di decisione in cui i responsabili valutano se il progetto proseguirà o verrà chiuso. Le revisioni, invece, sono riunioni più grandi, presentazioni degli ingegneri, domande e valutazioni tecniche. Il comitato di riesame delle revisioni comprende dirigenti e tecnici assegnati per posizione e formazione. Questi infatti hanno il titolo di Project Leader, avendo partecipato da sviluppatori ad un certo numero di revisioni e avendo completato la formazione aziendale interna [4].

I criteri di revisione sono espressi come una lista di controllo e vengono assegnati prima della revisione. Con eccezione rara, ogni criterio deve essere soddisfatto per continuare il lavoro. In alcune occasioni i progettisti possono ricevere una "release condizionale" in

una revisione, il che significa che il lavoro può continuare mentre un progetto viene modificato o corretto. In questi casi la metodologia di sviluppo di prodotto definisce in modo specifico cosa può essere un'eccezione alla regola e assegna anche un tempo massimo per le correzioni [4].

Essendo i requisiti congelati all'inizio della fase di sviluppo del prodotto, sono molto rare e non previste le iterazioni dovute da feedback del mercato.

Durante la reale applicazione della metodologia ci sono revisioni e modifiche al piano che non sono previste come ad esempio le revisioni fantasma. Queste sono effettuate dal Top Manager che ha la capacità di superare la rigidità della metodologia di sviluppo di prodotto adottato dalla società e in casi estremi può decidere di chiudere un progetto. La cancellazione improvvisa di un progetto di sviluppo (a favore di altri) si verifica di solito se un cambiamento di mercato crea una potenziale situazione finanziaria grave, se il lavoro su un determinato progetto continua. Dal momento che tali cancellazioni non sono affrontate, o tecnicamente consentite dal modello di sviluppo di prodotto ufficiale, non esistono criteri fissati per formalizzare queste decisioni. Infatti sono lasciate alla discrezione dei Top Manager [4].

6.3.2 Vantaggi e sfide

6.3.2.1 Vantaggi

Il vantaggio principale di questa metodologa è che le continue revisioni da parte di manager e tecnici formati riescono a scandire e far seguire la pianificazione per riuscire a consegnare un prodotto senza ritardi.

Una metodologa di prodotto così formalizzata non lascia spazio a deviazione dalla pianificazione che è stata fatta dopo il congelamento dei requisiti del prodotto.

6.3.2.2 Sfide

La principale sfida in Siemens Westinghouse Power Generation è l'estrema rigidità della metodologia nel caso dello sviluppo di un prodotto che richieda flessibilità nella definizione dei requisiti. Il modello di sviluppo di prodotto non prevede le iterazioni allo Stage precedente, anche se qualche volta avvengono [4].

La rigidità del sistema porta alla formazione di strumenti che non sono previsti dalla metodologia, ad esempio le revisioni fantasma che bloccano lo sviluppo di un prodotto senza un Gate formale. Questo fa sì che il progetto venga chiuso immediatamente senza altro spreco di tempo e risorse; seguendo la metodologia invece, si sarebbe proseguito fino al Gate successivo.

Un altro problema riguardante l'eccessiva rigidità della metodologia di sviluppo di prodotto è la disuguaglianza delle revisioni: alcune sono più importanti di altre.

Ad esempio la revisione del progetto finale è più importante delle altre e, oltre ad essere più impegnativa, impiega anche più tempo per essere fatta. La metodologia invece mette tutte le revisioni sullo stesso piano, prevedendo la stessa importanza [4].

6.3.3 Classificazione del caso di studio

Siemens Westinghouse impiega un rigoroso processo di Stage-Gate per gestire quasi esclusivamente il rischio tecnico nello sviluppo di turbine a gas. Esso è elevato a causa dei requisiti di qualità, delle piccole tolleranze nella costruzione delle turbine e delle esigenze produttive delle piattaforme complesse di prodotti su vasta scala. La metodologia di sviluppo di prodotto consente pochissimi cambiamenti nei requisiti durante lo sviluppo poiché le specifiche vengono congelate prima dell'inizio dei lavori. Inoltre raramente si ha un cambio di pianificazione o si consente ai Gate di essere flessibili. Le eccezioni al processo ordinato si verificano maggiormente in negativo, ad esempio quando un progetto viene chiuso dai Top Manager.

La maturità del mercato delle turbine a gas porta a progettare la macchina in un'ottica esclusivamente tecnica. Il sistema dei contratti basati sul risarcimento dei danni, assicura che la variabilità del mercato si traduca in un costo specifico che può essere evitato esclusivamente attraverso il successo tecnico.

Come descritto sopra, il caso di studio utilizza una metodologia di sviluppo di prodotto propria del sistema Stage-Gate. Infatti, a parte nelle fasi di definizione dei requisiti, né i clienti, né chi è a stretto contatto con i clienti interviene nel processo di sviluppo, il quale viene interamente eseguito dalla squadra di ingegneri e controllato da Manager tecnici con una pianificazione minuziosa e controlli rigidi.

I controlli ai Gate sono rigorosi e ciò porta ad avere un pieno controllo del processo di sviluppo di prodotto.

Secondo i criteri fissati nel Capitolo 5 il caso di studi può essere classificato:

Tabella 6.4 Classificazione del caso di studi Siemens Westinghouse Power Generation

Rigidità dei Gate	Alta
Presenza di feedback	Nulla

6.4 Caso Marel GRB

Marel GRB una compagnia islandese che produce macchine, sistemi e servizi per i processi industriali nel settore alimentare.

I prodotti di Marel GRB sono le linee di lavorazione del pesce, della carne e del pollame di cui un esempio è riportato in Figura 6.6. Questi macchinari vanno dal lavaggio, al taglio, alla lavorazione manuale, alla pesata della carne e del pesce. I prodotti integrano una parte hardware ed una software per il controllo delle componenti elettroniche e meccaniche chiamate “INNOVA” che offrono una panoramica totale sul processo e un coordinamento di tutti i macchinari [39]



Figura 6.6 Smart Line Grader di Marel GRB [39]

Marel GRB è stata fondata nel 1983 nel quartier generale di Gardabaer in Islanda. Nel 2007 cambia la sua identità in Marel Food System. Nel 2010 l'integrazione con la Stork Food Systems è stata completata e il suo nome ritorna a essere Marel GRB. L'azienda, a seguito della fusione ha cambiato la sua struttura organizzativa globale. La nuova organizzazione si concentra sui mercati come ad esempio la carne, il pollame, industrie di trasformazione alimentare di pesce. Nel 2013 è stata nominata fornitore dell'anno

dall'American Meat Institute. La compagnia vince il premio dell'innovazione del VIV Europe per i suoi processi innovativi [39].

L'azienda è composta da circa 4.700 dipendenti in tutto il mondo, uffici e filiali in 30 paesi ed una rete di più di 100 agenti e distributori. I suoi valori sono: unità, innovazione ed eccellenza.

Il Consiglio di Amministrazione di Marel GRB è impegnato nella buona corporate governance e nelle pratiche commerciali etiche che promuovono l'interesse a lungo termine degli azionisti e contribuiscono a creare fiducia nella società. I membri del Consiglio sono eletti in ciascuna Assemblea Generale per un periodo di un anno.

La corporate governance aziendale è costituita da un insieme di principi e regole, tra cui il suo statuto e le linee guida per la corporate governance emanate nel maggio 2015 dalla Camera di commercio dell'Islanda, Nasdaq OMX Iceland hf. e della Confederazione dei datori di lavoro islandesi.

L'organizzazione è stata suddivisa in Centri Industriali e Centri di Prodotto. I Centri Industriali si concentrano sugli interi processi del settore, mentre i Centri di Prodotto si concentrano su alcune fasi del processo. Questo è illustrato in Figura 6.7.

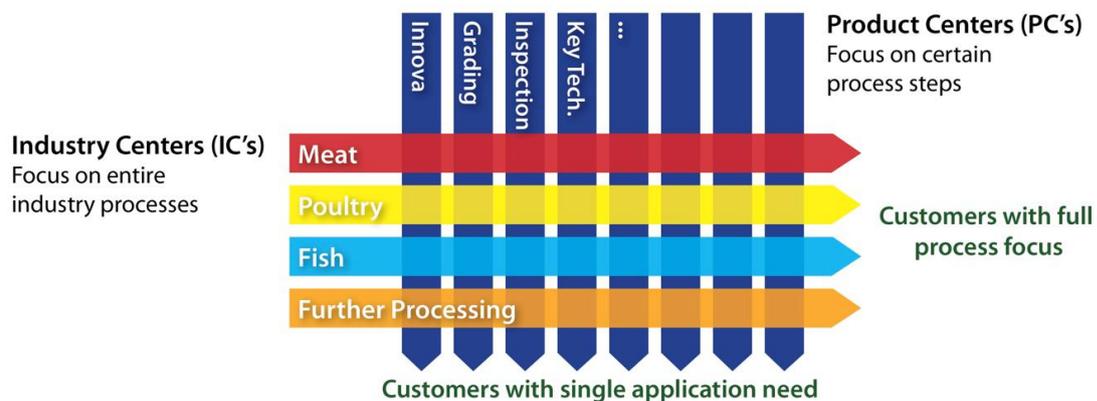


Figura 6.7 Struttura organizzativa di Marel GRB [28]

6.4.1 Lo sviluppo di prodotto

Marel GRB, per far fronte alle richieste del mercato di velocità nello sviluppo di nuovi prodotti, ha adottato la metodologia di sviluppo di prodotti Agile. Il dipartimento di sviluppo del prodotto si occupa sia di sviluppo software che di quello di prodotti tangibili. Prima dell'introduzione delle metodologie Kanban e Agile, il dipartimento di sviluppo

non aveva un sistema di lavoro definito. C'erano diversi progetti in corso e c'erano solo uno o due sviluppatori che lavoravano su ciascuno di essi.

Alcuni sviluppatori lavoravano su diversi progetti allo stesso tempo, e questi, erano spesso prodotti dove era già stata completata la fase di progettazione, ma si richiedeva un follow-up. C'era spesso discrepanza nella comunicazione, mancanza di coordinamento e condivisione delle informazioni tra il Product Manager e il Team meccanico [28].

Per migliorare la comunicazione all'interno dei Team di sviluppo e con il management, Marel GRB ha provato a introdurre il metodo Kanban. Questo però è stato un fallimento in quanto è stato poco seguito dagli sviluppatori [28].

È stata quindi introdotta la metodologia Agile, in particolare lo Scrum per lo sviluppo di software. È stata utilizzata per la prima volta in tre squadre che si occupavano di sviluppo software con un enorme successo.

Un anno dopo è stata introdotta e adattata per lo sviluppo di prodotti tangibili. Come per lo sviluppo di software, lo Scrum è stato applicato su un Team pilota di ingegneri meccanici che per lo sviluppo di un prodotto hanno collaborato con un altro Team di sviluppatori software. Il progetto pilota ha la durata di sette mesi al termine dei quali si deciderà la prosecuzione o meno dell'utilizzo dell'Agile nello sviluppo di prodotti fisici [28].

6.4.2 Framework di sviluppo di prodotto

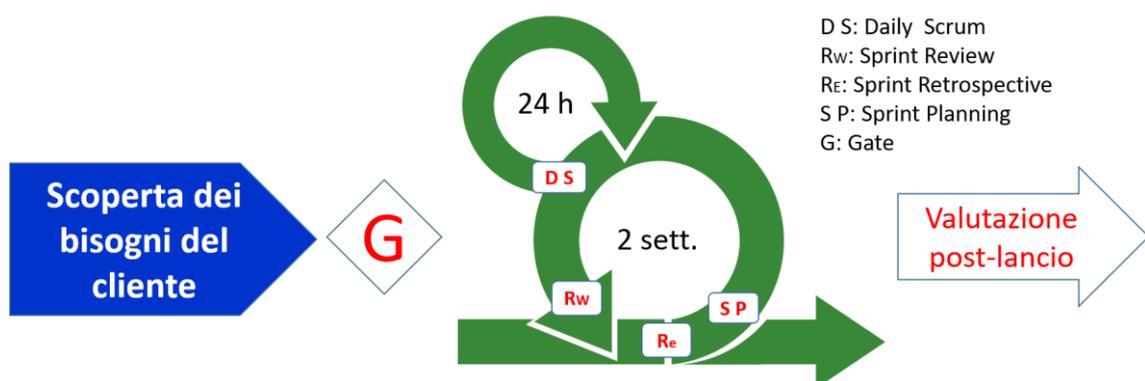


Figura 6.8 Metodologia di sviluppo di prodotto in Marel GRB

L'uso dello Scrum in Marel GRB è definito da un framework molto aderente alla guida Scrum [9]. In questo caso gli Sprint hanno una durata di 2 settimane e usano tutti i ruoli, artefatti ed eventi che la guida prevede.

Per la corretta introduzione delle squadre allo Scrum, la prima parte del progetto è dedicata alla formazione del Team. Questo periodo viene chiamato Agile Start-up.

Molto importante in Marel GRB la definizione dei ruoli [28].

il Product Owner è tassativo che sia estraneo al Team di sviluppo in quanto questi due ruoli possono essere in conflitto. Il Product Owner dovrebbe spingere la squadra dall'esterno, avere una visione del progetto e del mercato e assegnare le priorità trasferendo le informazioni dal mercato al Team.

Se fosse all'interno del Team potrebbe avere pregiudizi sulle priorità di progetto e dare priorità al proprio lavoro o difenderlo. Inoltre un buon Product Owner dovrebbe avere conoscenza del mercato e dei clienti, ma anche dei prodotti e della tecnologia usata. Una figura come questa è a volte difficile da trovare.

Il Team necessita inoltre di un leader tecnico, anche se il Product Owner dovrebbe prendere le decisioni più critiche e dare le priorità [28].

Lo Scrum Master non è un subalterno del Product Owner, e quindi il suo pensiero è indipendente dalle decisioni di quello. Lo Scrum Master dovrebbe essere anche una parte del Team in quanto dovrebbe conoscere bene tutti i suoi componenti.

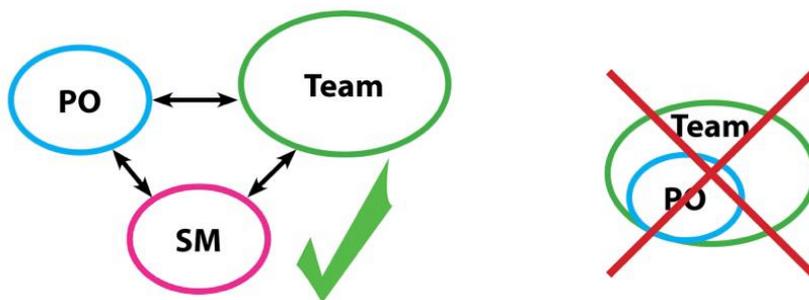


Figura 6.9 Miglior set up di ruoli Scrum [28]

6.4.2.1 Agile Start Up

In questa fase il Team viene formato all'utilizzo della metodologia di sviluppo di prodotto Scrum. Durante questa sessione, che dura pochi giorni, il Team è seguito da un consulente con esperienza nello Scrum. In Marel GRB è presente un Agile Center ovvero un centro di competenza che aiuta nella formazione dei gruppi, nelle riunioni e i Scrum Master nel

loro lavoro. L'Agile Center assiste le squadre nelle loro sessioni di pianificazione, e li sostiene nel lavorare sui loro progetti di miglioramento continuo. L'Agile Center definisce anche le linee guida che i Team dovrebbero seguire nell'utilizzo della metodologia Scrum [28].

La sessione di formazione del Team di solito inizia il lunedì mattina e finisce in pausa pranzo del venerdì della stessa settimana. Durante il primo giorno è spiegata la teoria, passando attraverso presentazioni ed esercizi. Successivamente si cerca di spiegare la visione Agile e si costruisce una tabella di marcia sulla pianificazione del primo Sprint e dei successivi.

Nella formazione, vengono chiariti gli obiettivi che si possono raggiungere con il metodo Agile [28]:

- Rendere più visibili i problemi
- Possibilità di suddividere il progetto in unità più piccole
- Più cooperazione tra i membri del Team
- Migliore distribuzione della conoscenza
- Visibilità maggiore dell'avanzamento
- Migliore visibilità degli elementi esterni al progetto
- Migliore organizzazione
- Migliore definizione dei ruoli
- Maggiore chiarezza delle priorità

Il secondo punto che viene affrontato è la definizione del Release Planning, nel quale il Team spezza il progetto in diverse parti in modo da poter ricavare delle attività che andranno a popolare il Product Backlog. Alle attività chiamate "Storie di Uscita" vengono assegnati dei punteggi a seconda del tempo e delle risorse che richiedono.

Sono stati definiti il numero di Sprint per il primo rilascio e la quantità di lavoro che il Team avrebbe dovuto fare per ogni Sprint [28].

In seguito si sono date le priorità alle Storie di Uscita (post-it blu nella Figura 6.10) e sono state distribuite sui sette Sprint. I sette gruppi di attività vengono chiamati Sprint Backlog. I lavori extra progetto sono stati aggiunti alla time-line (post-it verdi nella Figura 6.10). Le storie che necessitavano di aiuti esterni o di consegne esterne sono stati contrassegnati in rosa.

Sono state identificate le dipendenze esterne che in questo progetto di sviluppo di un prodotto fisico sono [28]:

- Il Team che lavora alla parte software
- Il lavoro in officina che costruisce i prototipi per tutte le squadre di sviluppo in Marel GRB. Questo è spesso un collo di bottiglia se l'officina è sovraccarica.
- PC Key-Tech (Operation System Center Product)
- L'arrivo di componenti critici che dovevano essere acquistati e di cui si deve attendere i tempi di spedizione. Qui il rischio principale è la lunghezza della spedizione.

Si è proceduto poi alla pianificazione dello Sprint. Le Storie di Uscita sono state spezzate in attività più piccole preferibilmente indipendenti, scrivendolo su post-it.



Figura 6.10 Lavagna di Scrum nel caso Marel GRB [28]

Infine si è pianificato l'orario del Daily Scrum o Stand-up quotidiano.

Un artefatto non previsto dalla guida Scrum, adottato in Marel sono le Next-in Story. Questo è stato introdotto dai Team software ed è una lista di attività e che vengono inserite nell'eventualità che le attività dello Sprint siano state completate in anticipo. Queste dovrebbero essere inserite nello Sprint successivo ma possono essere già affrontate nello Sprint attuale [28].

6.4.2.2 Lo Scrum nello sviluppo

Gli Sprint, iniziano con lo Sprint Planning, in cui i membri del Team di sviluppo, insieme con il Product Owner e lo Scrum Master, pianificano lo Sprint. In questa riunione, gli elementi dal Product Backlog vengono spostati allo Sprint Backlog secondo l'ordine di priorità e di una quantità ottimale per portarlo a termine nel periodo di due settimane.

Tutti i compiti presenti nello Sprint Backlog dovrebbero essere in ordine decrescente di priorità e i membri del Team dovrebbero prendere le attività dalla parte alta della lista. Se così non fosse, significa che nella fase di Sprint Planning le priorità non sono state assegnate correttamente o le attività non sono indipendenti [28].

Dovendo lavorare a stretto contatto con il Team di sviluppo software, il Team di sviluppo meccanico ha sincronizzato il tempo ciclo Sprint con il tempo di ciclo della squadra software per creare pianificazioni integrate. Così facendo sono riusciti a coordinarsi per organizzare in maniera più efficiente lo sviluppo ed i test.

Ogni giorno di sviluppo comincia con il Daily Scrum. In questo incontro quotidiano tutti i membri sono in piedi disposti a semicerchio. Prendono la parola uno alla volta passandosi il "Talking-Token", che in questo caso è una penna, che dà la parola a chi la possiede. Tutti i membri del Team rispondono a tre domande: "Che cosa ho fatto ieri?", "Cosa devo fare oggi?" e "Ci sono impedimenti?". Poi, ogni sviluppatore trasferisce un post-it dalla colonna "Da fare" alla colonna "In corso" e ci mette sopra la targhetta con il proprio nome.

Alla fine dello Sprint, il Team presenta qualcosa di "Fatto" nella riunione di Sprint Review. Qui, insieme con il Team software, presentano il loro prodotto di fronte al Product Owner e chiunque sia interessato alla presentazione.

Dopo lo Sprint Review si ha la riunione dello Sprint Retrospective nella quale il Team discute con lo Scrum Master sui problemi presenti nella metodologia adottata e su soluzioni o miglioramenti che andrebbero implementati.

6.4.3 Vantaggi e sfide

6.4.3.1 Vantaggi

Il vantaggio principale dell'uso di questo framework di sviluppo di prodotto è l'incremento di comunicazione all'interno del Team. Per incrementare ulteriormente questo fattore, è stato introdotto il kick-off che consiste nel dedicare tre ore al giorno nell'orario di lavoro al Team per interagire in modo che i componenti si conoscano meglio a livello personale e professionale. Le persone sono portate a parlare dei loro interessi nel progetto attuale e del contributo che possono portare nel Team. Condividono anche alcuni dettagli relativi alla loro vita personale e parlano del loro Team ideale [28].

La comunicazione all'interno del gruppo ha portato il Team a non lavorare più come singoli ma come una squadra. Da quando operano come un gruppo, ognuno ha la possibilità di dare un feedback sulle attività in cui altri sviluppatori stanno lavorando. Il modello Agile sostiene il trasferimento di conoscenze e crea un ambiente di discussione. Il miglioramento della comunicazione ha un effetto positivo anche nelle riunioni di pianificazione con il Product Owner e lo Scrum Master; infatti hanno dichiarato che la pianificazione e il Review erano più veloci ed efficaci [28].

Un altro vantaggio che hanno riscontrato i membri del Team di sviluppo è stato l'utilizzo del Daily Scrum, in quanto queste riunioni creavano discussioni sugli eventuali problemi che venivano riscontrati durante il giorno precedente. Questi venivano analizzati e si riusciva a trovare una soluzione velocemente. Se la soluzione non veniva trovata in un tempo ragionevole, si potevano organizzare meeting ad-hoc per risolverli [28].

Con l'utilizzo della metodologia Scrum, c'è stato un miglioramento dell'organizzazione dei progetti, una migliore suddivisione del lavoro secondo le varie priorità e un sistema visivo di avanzamento del progetto molto chiaro. Tale sistema è stato un vantaggio anche a livello manageriale poiché è stato possibile individuare i problemi di sviluppo del prodotto prima che con altre metodologie. Così possono essere affrontate il prima possibile problematiche come i colli di bottiglia e le attività che sono spesso causa di attese e potrebbero portare ritardi nello sviluppo del prodotto [28].

Un maggior focus degli sviluppatori in un singolo progetto, come impone la metodologia Agile, ha portato ad un miglioramento e ad un incremento della velocità di sviluppo, in quanto gli sviluppatori non erano più distratti da lavori diversi ed interferenze esterne.

Un altro vantaggio importante è che il framework fornisce alla squadra un feedback veloce per il proseguimento dello sviluppo nella giusta direzione e la visione immediata dello stato dei lavori completati, in fase di sviluppo e non ancora affrontati. Gli Sprint Retrospective invece, portano a feedback da parte del Team per migliorare la metodologia adattata all'azienda in cui viene applicata [28].

La sincronizzazione tra il Team meccanico e software, ha reso più facile vedere discrepanze: per esempio, se la squadra meccanica assegna troppi pochi giorni per i test del software su un nuovo prototipo, il Team di software può farlo notare durante la sessione di pianificazione.

6.4.3.2 Sfide

La principale sfida che questo progetto pilota ha dovuto sostenere è la difficoltà nell'applicazione delle metodologie Scrum, che solitamente sono usate per lo sviluppo di prodotti software e fisici. I prodotti fisici, a differenza dei software, hanno tempi d'attesa se si vuole ottenere un prototipo da utilizzare o testare [28].

Per la costruzione del prototipo sono due i fattori che possono rallentare il lavoro: la coda di lavoro e l'attesa dal fornitore per un pezzo non presente nel magazzino di Marel GRB. Per il primo fattore in Marel GRB tutti i prototipi vengono costruiti da un altro reparto, il Product Development Workshop. Questo reparto è un'officina che costruisce i prototipi per tutti i Team di sviluppo e i tempi di attesa dipendono dalla complessità, dal lead time e specialmente da quanto è occupato il reparto con altri prototipi [28].

L'attesa per l'ordine di un componente da un fornitore è un altro fattore che porta ritardo poiché richiede tempo, a maggior ragione se il pezzo richiesto è sbagliato.

La sfida in questo caso è modificare la metodologia applicata in Marel GRB in quanto quella utilizzata seguendo la guida Scrum non prevede la presenza di ritardi nella consegna del prodotto finito. Per ovviare a questi problemi, la metodologia applicata dovrebbe essere modificata allargando la definizione di "Fatto" che viene richiesto alla fine dello Sprint per esempio ad un modello 3D del prodotto o ammettendo la possibilità di posticipare la consegna del prototipo fisico [28].

Un'altra sfida che è stata riscontrata è la difficoltà di spezzare un lavoro in attività più piccole indipendenti tra di loro. Questo è un limite dell'applicazione della metodologia Agile ai prodotti fisici, in quanto questi non sono solo righe di codice, ma componenti fisici che non possono funzionare senza la presenza di altri.

Nell'esperimento di applicazione della metodologa Agile a un Team che ha lavorato sempre con la metodologia tradizionale, la squadra ha avuto la tendenza a lavorare più come individui che come una squadra. Il lavoro di squadra è fondamentale per la buona riuscita dell'applicazione della metodologia Scrum [28].

È anche importante una cultura organizzativa che permetta la fiducia tra colleghi all'interno delle squadre. È necessario che: le persone non abbiano paura degli individui più anziani; mantengano una mentalità aperta così che le persone possano dire la loro su ciò che stanno facendo; non abbiano paura di ammettere gli errori. Un'atmosfera rilassata nel Team per quanto riguarda questi aspetti è importante [28].

Inoltre i Team devono cambiare modo di lavorare: da singoli a lavoro di squadra. Le metodologie di sviluppo di prodotto Agile non sono aderenti alle capacità di persone che preferiscono lavorare da sole.

Un altro problema che è stato riscontrato durante lo svolgimento degli Sprint è stata la poca disponibilità del Product Owner. Il ruolo del Product Owner è quello di sostenere il Team durante il progetto, specialmente negli Sprint Review [28].

Il Product Owner, infatti, in questo caso di studi non aveva la piena comprensione del proprio ruolo, che era principalmente quello di dare priorità al lavoro di Team, in base al valore che le attività avevano per il cliente.

6.4.4 Classificazione del caso di studio

Questo caso di studi è un buon esempio di Scrum applicato allo sviluppo di prodotti fisici. Pur non avendo avuto un seguito, dato il rifiuto del Team nel continuare a lavorare usando questa metodologia, questo caso offre una testimonianza dei vantaggi e delle difficoltà affrontate durante l'applicazione di questa metodologia nello sviluppo di prodotti fisici. In particolare, dopo aver definito i requisiti ed essere stato approvato, il progetto è stato sviluppato utilizzando la metodologia Agile, applicandola in modo più simile possibile a quella usata per lo sviluppo software.

Il caso di studi può essere classificato secondo le grandezze scelte in questo elaborato molto vicino alla metodologia Agile in quanto ci sono frequenti feedback da parte del Product Owner per conto dei clienti, necessari per il proseguimento del progetto.

Per quanto riguarda la presenza e la rigidità dei Gate, questi non sono ben definiti, come la presenza degli Stage e delle milestone; inoltre sono molto flessibili per permettere uno sviluppo dinamico del prodotto.

Esiste comunque un controllo da parte del Product Owner che funge anche da Project Manager e valutava alla fine dello Sprint Review il lavoro del Team. I Gate sono poco presenti e molto flessibili. È presente un Gate chiaro all'inizio del processo di sviluppo di prodotto che dà il via al progetto.

La classificazione del caso di studio secondo le grandezze scelte, è riportato nella tabella sottostante:

Tabella 6.5 Classificazione del caso di studi Marel GRB

Rigidità dei Gate	Bassa
Presenza di feedback	Alta

6.5 Caso Andritz Hydro AB Svezia

Andritz è un fornitore leader a livello mondiale di impianti, attrezzature e servizi per le stazioni idroelettriche, l'industria della cellulosa e della carta, delle industrie metallurgiche e per la separazione solido/liquido [40].

Il gruppo Andritz ha sede a Graz, in Austria, con un personale di circa 25.200 dipendenti e opera in oltre 250 siti in tutto il mondo.

La sezione di Andritz Hydro è un fornitore globale di sistemi e servizi elettromeccanici per le centrali idroelettriche e uno dei leader del mercato mondiale per la produzione di energia idraulica [40].

L'energia idroelettrica è la forma più economica dell'energia rinnovabile. In stretta collaborazione con i suoi clienti, Andritz Hydro elabora soluzioni di lunga durata, rispettose dell'ambiente e economicamente efficienti per la produzione di energia idraulica. Mantenere l'ambiente naturale, l'umanità e la tecnologia in perfetta armonia è una parte importante della cultura aziendale di Andritz Hydro [40].

Andritz Hydro, oltre a vantare più di 175 anni di esperienza nel settore, è responsabile di aver installato più di 30.000 turbine idroelettriche nel mondo e ha le capacità e l'esperienza per sviluppare e installare turbine con potenza maggiore di 800 MW.

La divisione è capace di gestire completamente il progetto, progettare l'impianto e eseguire i test del modello [40].

Inoltre tutte le apparecchiature meccaniche ed elettriche destinate alla centrale idroelettrica vengono progettate, fabbricate, assemblate e installate da Andritz Hydro.

Un impianto idroelettrico si compone di turbine, regolatori di velocità, valvole di arresto, generatori (sincroni e asincroni), sistemi di alimentazione elettrica, compresi i dispositivi a bassa, media e alta tensione [40].

Andritz Hydro garantisce inoltre: automazione, controllo, protezione, eccitazione e sincronizzazione di tutte le parti, gestione del sito, installazione, messa in servizio, test, monitoraggio, diagnosi dell'impianto, formazione per gli addetti ai lavori e grandi progetti di riabilitazione di impianti vecchi. La compagnia, oltre a progettare e costruire impianti nuovi, ha molta esperienza nella manutenzione e nella ristrutturazione di impianti vecchi, anche dismessi. [40]



Figura 6.11 Esempio di un impianto progettato da Andritz Hydro [40]

Le grandi turbine sono il fiore all'occhiello di Andritz Hydro, le turbine possono arrivare fino ad 800 MW. L'azienda offre: le turbine Pelton per applicazioni ad alte prestazioni; le turbine di Francis per una portata di rete media; le turbine Kaplan per elevate potenze generate. Andritz Hydro è nota per quelle sviluppate per scopi specifici con tecnologie particolari come le turbine a pompaggio o turbine a corrente continua, generando potenza dalla velocità dell'acqua [40].

6.5.1 L'organizzazione

Andritz Hydro per gestire i vari progetti, ha cinque divisioni, una delle quali è la divisione Service & Rehab. Questa divisione è responsabile dello sviluppo di soluzioni innovative per incrementare le caratteristiche e fornire servizi per tutto il ciclo di vita delle centrali idroelettriche [28].

Qui ogni progetto è adattato al cliente e pertanto tutti i prodotti possono essere considerati unici. Il suo lavoro potrebbe essere classificato più come vendita di design, che come sviluppo puro del prodotto di Ricerca & Sviluppo. Prima della fusione con Andritz Hydro non c'erano squadre di progetto, ma squadre specializzate per alcuni settori tecnici,

costruiti a seconda della tipologia dei prodotti. L'integrazione dei prodotti veniva eseguita da un piccolo team di progetto, che includeva un Project Manager, un System Engineer, oltre ad un Site Manager durante la costruzione in situ. In questo modo tutte le squadre tecniche erano coinvolte in quasi tutti i progetti che venivano svolti nell'azienda [28].

Dopo la fusione della società Waplans nel gruppo Andritz Hydro, sia il numero dei progetti che i requisiti di mercato sono aumentati, costringendo l'azienda a un cambio di organizzazione del dipartimento della divisione Service & Rehab. È stata creata una nuova struttura organizzativa, come rappresentato in Figura 6.12 [28].

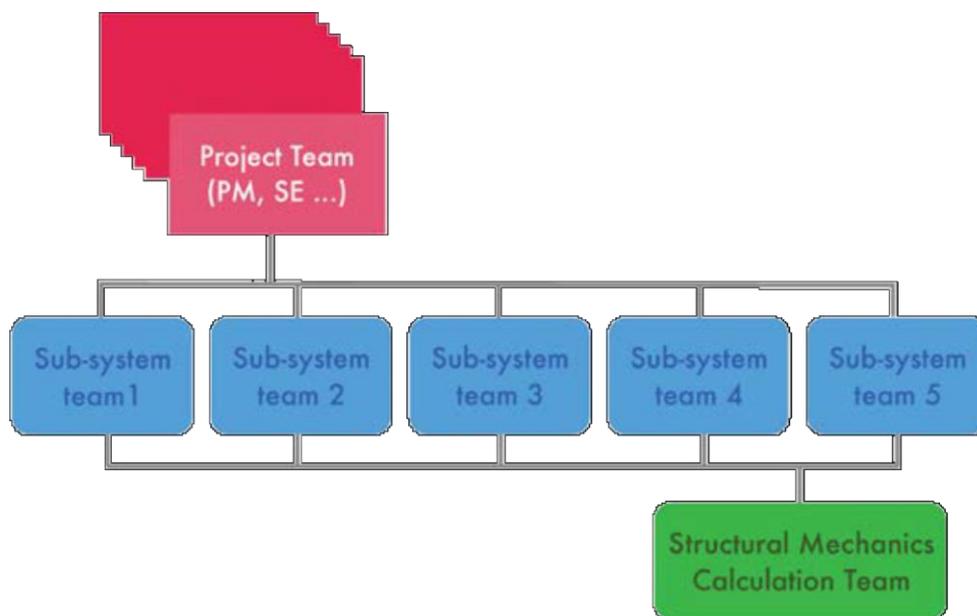


Figura 6.12 La nuova struttura organizzativa della divisione Service & Rehab [28]

Nella nuova struttura organizzativa della divisione Service & Rehab, gli sviluppatori non sono più assegnati a progetti specifici, ma sono stati collocati in sette squadre di sviluppo. Per ogni lavoro c'è un Team di progetto con almeno un Project Manager e un System Engineer.

I Team di progetto sono responsabili nel fornire alle squadre di sviluppo le informazioni necessarie, quali i requisiti dei clienti, le specifiche tecniche e le specifiche del progetto. Inoltre sono anche responsabili nel dare la priorità a determinati pacchetti di lavoro o compiti all'interno dei loro progetti.

I gruppi di sviluppo sono responsabili della propria pianificazione del lavoro e dello sviluppo del prodotto. Infine è presente un Team di verifica strutturale che serve tutti gli altri Team di sviluppo [28].

6.5.2 Lo sviluppo di prodotto

I progetti di sviluppo in Andritz Hydro hanno scadenze molto rigide, con enormi penalità per eventuali ritardi. Se un progetto viene consegnato con due settimane di ritardo, l'intero profitto del progetto è utilizzato per coprire le sanzioni. Questo mette un'alta pressione sulla pianificazione e l'organizzazione, non ci si può permettere alcun ritardo nel rilascio [28].

Prima della fusione, i progetti potevano richiedere 2-3 anni a causa di lunghi tempi di consegna. Per questo motivo il personale doveva lavorare su almeno due progetti contemporaneamente.

Dopo la fusione della società Waplans nel gruppo Andritz Hydro, l'aumento del lavoro e dei requisiti del mercato ha portato, oltre alla riorganizzazione del dipartimento, anche il cambio di mentalità e di framework di sviluppo di prodotto [28].

Per rispondere alla velocità del mercato, hanno adottato una mentalità Agile pur rimanendo molto legati alla metodologia di sviluppo di prodotto Plan-Driven. Come spiegato sopra, con la nuova organizzazione avente sette squadre di sviluppo i Team vengono suddivisi a seconda della tipologia fisica di prodotto, cioè ogni squadra è responsabile di uno dei sottosistemi o dei componenti diversi di cui esso è costituito [28]. Ogni progetto utilizza il supporto tecnico necessario per il suo svolgimento, in alcuni casi solo una delle squadre potrebbe essere coinvolta, per altri progetti invece possono essere coinvolte tutte.

L'implementazione della metodologia Scrum è iniziata con una squadra pilota propensa al cambiamento e con maggiore esperienza. Durante la sperimentazione, tutte le squadre hanno partecipato allo Sprint Retrospective, così da poterne vedere il funzionamento, ma non tutti erano costretti a iniziare a lavorare con la metodologia Agile. Dopo qualche tempo, visti i risultati della squadra pilota, anche le altre squadre hanno chiesto di iniziare a utilizzare il nuovo metodo.

L'adozione della metodologia Agile ad un certo punto, quindi, non è stata solo un'imposizione Top-Down ma un volere degli stessi membri dei Team di sviluppo [28]. Un fattore importante che ha portato questa metodologia ad essere integrata con successo all'interno del processo di sviluppo di prodotto è stato l'aiuto di un consulente affidabile ed esterno all'organizzazione [28].

6.5.2.1 Framework di sviluppo di prodotto.

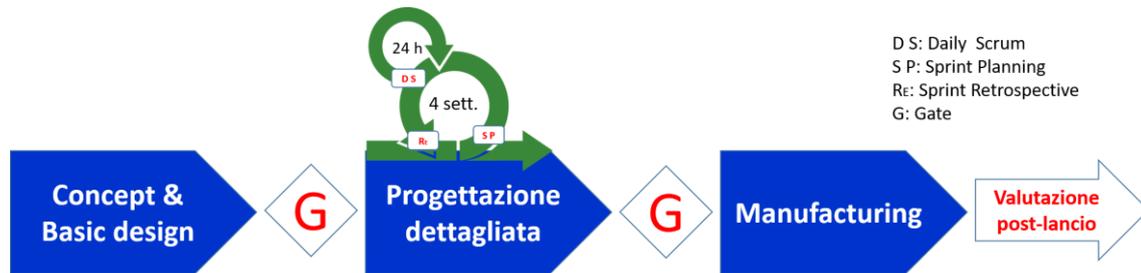


Figura 6.13 Metodologia di sviluppo di prodotto in Andritz Hydro

La metodologia maggiormente utilizzata è quella Plan-Driven: il lavoro infatti, ha delle tempistiche e obiettivi definiti dai prerequisiti forniti dal Team di progetto, congelati al momento della partenza del progetto stesso.

Ci sono tre Stage principali [28]:

1. Concept & Basic design
2. Progettazione dettagliata
3. Manufacturing

Ogni Stage ha una serie di Gate intermedi che monitorano strettamente l'avanzamento e la scadenza del progetto.

La particolarità del Framework in Andritz Hydro consiste nell'integrazione della metodologia Agile a livello di pianificazione.

Vista l'importanza che le scadenze hanno per i progetti, nella divisione Service & Rehab ci sono stati tre livelli di pianificazione all'interno dello sviluppo: strategico, tattico e operativo. Per riuscire a rendere efficaci questi livelli di pianificazione, il framework richiede molti meeting con diverse cadenze. In Figura 6.14 sono riportate le cadenze di pianificazione operativa e tattica [28].

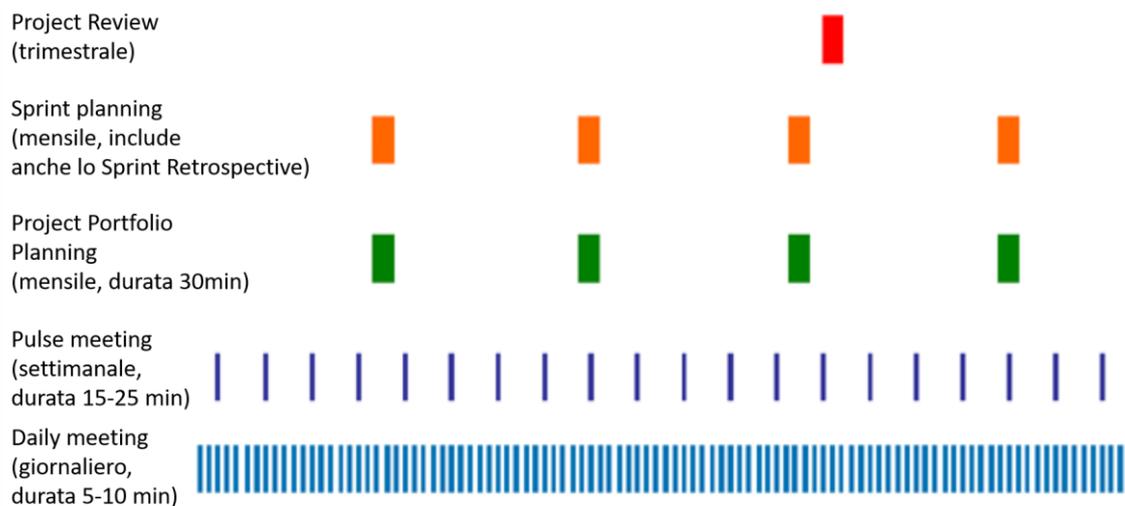


Figura 6.14 La cadenza delle riunioni della divisione Service & Rehab [28]

La pianificazione strategica è un piano a lungo termine che non viene revisionato e modificato spesso. A questo livello, viene stimato il numero di risorse necessarie allo sviluppo del prodotto e si scelgono i Team adatti a portare il progetto al successo [28].

La pianificazione tattica è un livello intermedio: in essa viene organizzato il progetto, per una durata di 1-3 anni oppure una parte indipendente del progetto e riguarda principalmente la gestione e l'assegnazione delle risorse, le scadenze e le milestone. Durante le riunioni per questo tipo di pianificazione, con cadenza trimestrale, ci si concentra sul budget e vi partecipa un rappresentante della sede di Andritz [28].

A livello mensile si svolgono le riunioni del comitato direttivo con i vari Team. Nella prima parte del meeting i Team di progetto si riuniscono con i System Engineer per un meeting Sprint Retrospective comunitaria e viene riferito lo stato finanziario, mentre nella seconda parte viene affidato ad ogni squadra di sviluppo il lavoro da compiere [28].

Il meeting Sprint Retrospective all'inizio della giornata di pianificazione serve ad attuare un processo di auto-miglioramento all'interno della metodologia di sviluppo di prodotto e il dipartimento adotta una speciale lavagna di miglioramento continuo per tenere traccia di progetti di implementazione della metodologia [28].

La seconda parte della riunione ha la durata di 30 minuti per squadra e qui i System Engineer assegnano i pacchetti di lavoro preparati per lo Sprint successivo. Questi vengono preparati insieme ai Team di progetto prima della riunione, stimandone la dimensione e la durata. Durante questa parte del meeting, viene data la priorità alle attività e si decide ciò che sarà incluso nello Sprint successivo [28].

La pianificazione operativa è il livello più basso di pianificazione e riguarda una pianificazione molto dettagliata. Si divide anch'essa in due livelli temporali: la pianificazione mensile e la pianificazione quotidiana, segue le regole di pianificazione scritte nella guida Scrum ed è fatta dagli sviluppatori.

Questi infatti sono responsabili della gestione del proprio lavoro. Le riunioni di pianificazione dello Sprint si tengono mensilmente e sono collegate alla riunione del portfolio progetti. In questo meeting vengono assegnate le priorità ai requisiti che sono stati decisi nella riunione portfolio progetti con il comitato direttivo [28].

Durante questa riunione, infatti, ogni squadra prende il proprio pacchetto di lavoro allocandolo nella propria workstation ed esegue la propria pianificazione, distribuendo internamente i pacchetti. È importante che tutta la squadra partecipi alla riunione di pianificazione dello Sprint per creare unità all'interno del gruppo. Se partecipasse solo il Team Leader, potrebbe capitare che nel momento della spiegazione del piano al gruppo, gli sviluppatori non capiscano le attività da svolgere o non siano d'accordo con le priorità assegnate a ciascun task [28].

A livello inferiore ci sono i Pulse meeting. Questi si svolgono settimanalmente, di solito il venerdì. L'incontro è tra i Team di progetto, i Team di sviluppo e il Line Manager. Il focus è sulle deviazioni dal piano, dal calendario o dalla qualità del prodotto sviluppato.

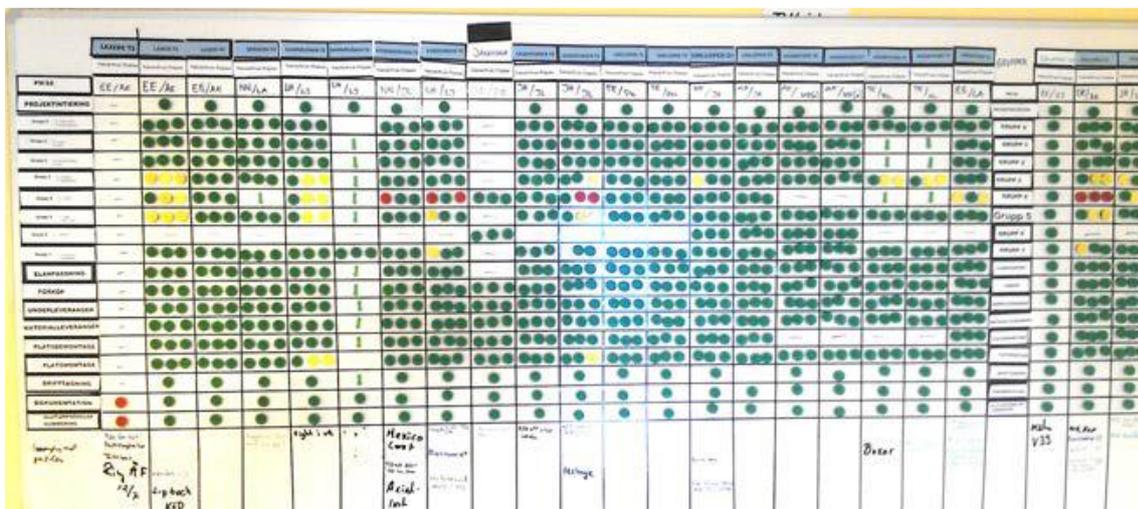


Figura 6.15 Tavola di Pulse Meeting della divisione Service & Rehab [28]

Si utilizza una lavagna (rappresentata in Figura 6.15) nella quale su ogni colonna è rappresentato un progetto, mentre sulle righe vengono rappresentati i diversi Team di sviluppo ed altri dipartimenti che sono coinvolti nello sviluppo del prodotto. In questo

modo si ha un controllo visivo dei possibili problemi che ciascun Team può avere in ogni progetto [28].

Per ogni progetto ci sono tre indicatori importanti per identificare il corretto svolgimento, senza intoppi di lavoro: la progettazione, la produzione e la pianificazione. Per ogni indicatore ci sono tre etichette che possono essere inserite nella lavagna: il verde significa che va tutto bene; giallo che c'è un problema ma c'è un piano d'azione; con il rosso invece c'è un problema ma non esiste una soluzione e le persone in questione non possono gestire la situazione per conto loro. Nel caso in cui non ci sia un'etichetta, significa che quell'attività non è presente [28].

Infine a livello inferiore si ha la pianificazione utilizzata dal Team di sviluppo seguendo le regole del Daily Scrum. Anche se, secondo la metodologia originale, la pianificazione dovrebbe essere fatta giornalmente, la cadenza può variare da una volta al giorno a due volte alla settimana.

Ogni squadra ha un proprio piano di lavoro giornaliero, diverso da quello di altre squadre. Il Team di sviluppo riceve una certa libertà quando si tratta di organizzare il lavoro operativo [28].

Come la metodologia Scrum suggerisce nella lavagna, le attività che devono essere completate in quello Sprint sono raccolte inizialmente nella colonna "Backlog".

Queste, quando vengono prese in carico da uno sviluppatore, sono spostate nella colonna "In Progress". Una volta completate passano nella colonna "Finished". Infine vengono spostate nella colonna "Implemented" una volta che il lavoro è stato approvato [28].

Inoltre nella parte superiore della lavagna c'è una sequenza temporale con tre frecce. Questo strumento serve come Burn-Down Chart, la scala è 0-100% e la freccia rosa rappresenta il tempo trascorso dall'attuale Sprint, quella verde rappresenta quanto è il lavoro previsto, mentre la gialla il lavoro extra non pianificato che è stato aggiunto al Sprint [28].

All'inizio di un Sprint, spesso c'è una grande distanza tra la freccia rosa e quella verde, ma la distanza diminuisce a mano a mano che ci si avvicina al termine. La timeline viene aggiornata durante la riunione quotidiana.

Nella lavagna anche la sezione di Follow up è molto importante, in quanto assicura che i nuovi miglioramenti funzionino come previsto. Una nota di miglioramento non viene presa dalla lavagna finché la modifica non funziona, viene verificata e documentata ed inserita nel sistema di gestione ISO 9000 [28].

6.5.3 Vantaggi e sfide

6.5.3.1 Vantaggi

Il primo e più importante vantaggio di questa metodologia è che con questa fitta rete di pianificazione, lo sviluppo del prodotto è completamente monitorato dalla definizione dei requisiti da parte del cliente fino alla consegna.

Qui ogni livello di pianificazione è eseguito dalle persone più competenti in merito. Per la pianificazione operativa, gestita dal Team di sviluppo, l'uso delle regole Scrum portano ad avere un piano costruito ad hoc ogni giorno, in modo che ogni membro della squadra sappia in ogni momento come avanzare con lo sviluppo [28].

Per verificare i vantaggi della metodologia, il management ha voluto analizzare le ore di sviluppo usate per ogni progetto in cui ha lavorato il Team pilota. Come standard è stato scelto un progetto su cui la stessa squadra ha lavorato prima dell'attuazione della nuova metodologia di sviluppo di prodotto [28].

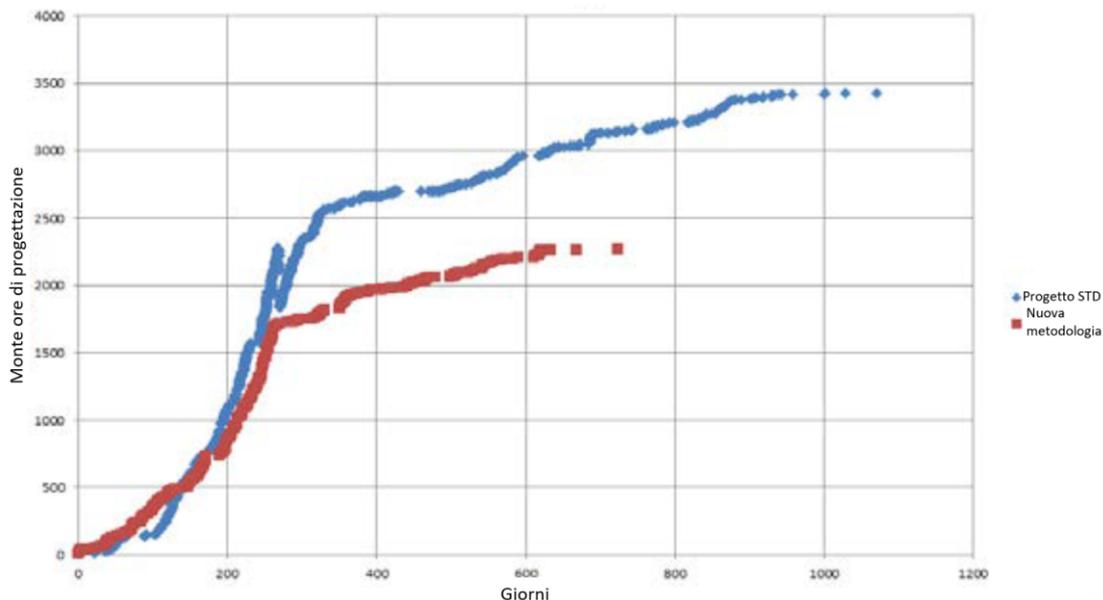


Figura 6.16 Confronto tra le ore impiegate per lo sviluppo prima (blu) e dopo (rossa) l'implementazione della metodologia Agile [28]

Nel grafico (Figura 6.16), la curva Blu rappresenta il progetto di riferimento e la curva rossa rappresenta il primo progetto della squadra pilota dopo la nuova implementazione del metodo. Come si può notare, la squadra è più veloce in quanto impiega meno ore di progettazione per arrivare allo scopo e riesce a sviluppare il prodotto in minor tempo [28].

Altri vantaggi che porta la metodologia sono gli strumenti visivi di avanzamento. Questi forniscono una buona panoramica per la gestione dello stato delle attività quotidiane ed anche per il management è semplice identificare i possibili problemi nello sviluppo e gli impedimenti che possono portare a ritardi nella consegna [28].

6.5.3.2 Sfide

Ci sono stati degli handicap che hanno limitato il corretto uso della metodologia nell'esperimento della squadra pilota, come l'implementazione di un nuovo sistema CAD, la bassa ripetibilità delle attività nel progetto, un rigoroso controllo sulla qualità sia sulle prime fasi del design che prima della produzione con conseguente dilatazione dei tempi di sviluppo [28].

Anche se il Team, composto da membri che lavoravano in azienda da 1-2 anni, tranne uno di essi che vi lavorava da 8 anni, aveva poca esperienza, la squadra si è comportata bene e ha portato a termine il progetto in anticipo [28].

Nei tre mesi successivi, i miglioramenti visti in precedenza sono stati in una certa misura persi, in quanto gli sviluppatori impiegavano un monte ore maggiore per lo sviluppo dei prodotti. Questo non è imputato alla metodologia, ma all'aumento dei requisiti dei clienti e dal sovraccarico del dipartimento di sviluppo [28].

Come per qualsiasi metodologia applicata, se le risorse sono insufficienti, il carico di lavoro per ogni singolo sviluppatore aumenta con un'inevitabile dilatazione dei tempi di consegna.

Una limitazione della metodologia attuata è la mancanza degli Sprint Review al termine di ogni Sprint. I team di sviluppo infatti non consegnano niente di "Fatto" al termine del ciclo di lavoro mensile, ma la rete di pianificazione e controllo è talmente fissa che il loro lavoro viene monitorato costantemente dal management [28].

I fattori che appesantiscono molto il lavoro dei Team di sviluppo sono le attività non pianificate. Il sistema, infatti, non fornisce alcuno strumento per gestirle.

Quando il monte ore non pianificato aumenta, è molto difficile ridefinire il piano in corso d'opera per includerle questo lavoro extra. [28].

6.5.4 Classificazione del caso di studio

Andritz Hydro nel suo metodo di sviluppo di prodotto ha integrato l'Agile nel modello Stage-Gate.

In questo caso il metodo Scrum è integrato come micro planning in cicli di tempo limitato in quanto in questa azienda le scadenze sono molto strette. Anche se la pianificazione è minuziosa, l'Agile, in particolare lo Scrum, si adatta perfettamente alla metodologia Stage-Gate dell'azienda in quanto fornisce strumenti efficaci per la pianificazione a breve termine. La pianificazione operativa è lasciata ai membri del Team di sviluppo che sono le persone più competenti per riuscire a pianificare il proprio lavoro.

Per quanto riguarda le grandezze scelte in questo elaborato per la classificazione dei casi di studio, i feedback da parte del cliente sono inesistenti. Infatti nella maggior parte dei progetti tutti i requisiti vengono congelati all'inizio dello sviluppo. Gli unici feedback diretti con i clienti si hanno al momento dell'ordine e della consegna dei prodotti.

Il Team di progetto svolge il ruolo del Product Owner e si occupa, oltre che della pianificazione, anche di controllare che il progetto rispetti i requisiti. Non essendo presenti nella metodologia gli Sprint Review, gli sviluppatori mostrano l'avanzamento del progetto durante le revisioni effettuate nei Gate.

I Gate in questo caso di studio sono molto usati poiché il lavoro viene pianificato su tre livelli, due dei quali usano la metodologia tradizionale. Trimestralmente si indice una riunione in cui viene verificato, oltre il rispetto dei requisiti, anche se le risorse impiegate nel progetto rispettano il budget iniziale. L'uso delle metodologie Agile nel micro planning non influisce sull'efficacia dei Gate in quest'azienda.

Tabella 6.6 Classificazione del caso di studi Andritz Hydro AB

Rigidità dei Gate	Alta
Presenza di feedback	Nulla

6.6 Caso Compagnia A

Il caso di studi in questo paragrafo è stato analizzato nella tesi di Ovesen [41]. Questa società danese è un'azienda produttrice di chitarre, bassi e amplificatori. L'azienda è presente in Danimarca dove ha la sua sede centrale, ma una parte della produzione è in Thailandia e sta avviando uno stabilimento in Cina. Inoltre la società possiede dei distributori indipendenti sia in Europa che in America e Asia. La sede centrale in Danimarca, dove avviene lo sviluppo di nuovi prodotti, comprende 185 dipendenti di cui 40 sviluppatori che lavorano nella sezione di ricerca e sviluppo [41].

6.6.1 Lo sviluppo di prodotto

Il dipartimento di sviluppo della Società è organizzato in tre settori di attività: la divisione Chitarra e Basso, la divisione Amplificatori e la divisione Pro. Ogni divisione dispone di un proprio team cross-funzionale composto da ingegneri informatici, ingegneri elettronici, ingegneri meccanici e un certo numero di persone con altre competenze. Alcuni membri del team sono "risorse condivise", il che significa che questi dipendenti specifici sono collegati a più team [41].

Nel 2008 il dipartimento di sviluppo è stato sottoposto a un processo di ristrutturazione da un'organizzazione tradizionale a matrice con ciascun dipendente affiliato principalmente ad un gruppo di competenze professionali simili e allo stesso tempo impegnato in progetti multipli di carattere cross-funzionale.

Lo Scrum è stato prima implementato in tre squadre del dipartimento di sviluppo software su richiesta del Develop Manager. L'esperimento è stato seguito con interesse da tutto il management, in quanto è stato assegnato a questo Team un progetto critico già affrontato da un'altra divisione ma con scarsi risultati e a rischio fallimento [41].

Il progetto sviluppato con l'uso di questa metodologia è stato un successo e il Management ha deciso di puntare sull'uso dello Scrum anche per altri progetti. Data la crisi globale nel 2008, l'Amministratore Delegato decide di implementare attentamente lo Scrum con squadre cross-funzionali in tutta l'organizzazione di sviluppo [41].

Prima dell'attuazione dell'attuale framework, i reparti di sviluppo lavoravano molto con il modello di sviluppo Stage-Gate, raffinando la metodologia per portarla il più possibile ad essere aderente ai prodotti sviluppati. Nell'uso di quella metodologia, i dipendenti

erano impegnati in più progetti contemporaneamente e si incontravano solo una volta alla settimana per fare il punto della situazione su ogni progetto.

Alcuni Line Manager, con insufficiente contatto con i progetti, discutevano sulle priorità dei diversi compiti. Data l'organizzazione a matrice, gli sviluppatori erano in un conflitto costante su chi seguire, se il Line Manager o il Project Manager [41].

Il Top Management, viste queste difficoltà all'interno dell'organizzazione, ha deciso di apportare un cambio di paradigma e di integrare lo Scrum all'interno della metodologia Stage-Gate già presente all'interno dell'azienda. Lo Scrum è stato imposto dal Top Management, il che significa che l'organizzazione era ben preparata all'innovazione introdotta [41].

Per i prodotti commercializzati dall'azienda studiata, il design e lo styling sono una parte importante nello sviluppo del prodotto. Tuttavia, queste caratteristiche del prodotto sono congelate presto nel processo e con ciò si ha che il design non è più una parte dinamica che viene sviluppata durante dell'intero processo di sviluppo seguendo i feedback dei clienti.

Nella progettazione concettuale gli Sprint, di solito 2, si concentrano spesso sul design concettuale del prodotto, sia per quanto riguarda la specificazione hardware, software, sia per la costruzione meccanica. Completata la fase, il risultato di questo processo potrebbe essere un disegno del prodotto, della selezione dei colori o dell'interfaccia [41], congelato durante lo sviluppo.

Si utilizza lo Scrum anche nella fase di sviluppo, mentre negli altri Stage si usa la metodologia tradizionale con pianificazioni precise e Gate efficienti.

6.6.1.1 Framework di sviluppo di prodotto



Figura 6.17 Metodologia di sviluppo di prodotto in Compagnia A

La società utilizza un modello di processo Stage-Gate come strumento di gestione generale per le attività di sviluppo. Il modello Stage-Gate è stato implementato prima dell'introduzione dello Scrum e ha continuato ad essere il principale schema guida. Le attività di Scrum nell'organizzazione dello sviluppo sono state integrate nel modello Stage-Gate, soprattutto nella terza fase, quella di sviluppo. L'azienda cerca di spingere l'utilizzo della metodologia Scrum anche nella fase di sviluppo del concetto, ma con scarsi risultati [41].

In questa fase, l'Azienda utilizza un Product Backlog, un elenco per descrivere il prodotto.

Con la descrizione presente del Product Backlog, vengono create delle User Story che sono requisiti di prodotto che in questo caso possono essere descritti in modo generale con poche frasi [41].

Si sviluppano poi dei diagrammi a blocchi, con descrizioni particolareggiate per ogni step che andranno poi a formare lo Sprint Backlog.

Una parte importante di questo processo è la stima delle risorse necessarie per ognuna delle voci del Product Backlog. Questa stima viene effettuata dal Team di sviluppo, essendo esso il più competente in materia.

La stima viene quantificata in punti storia, che è una misura che unisce dimensioni e complessità di un determinato compito. Con una stima complessiva sotto forma di punti di storia, basandosi anche sulle esperienze esistenti riguardanti la velocità del team, è possibile calcolare approssimativamente i tempi e il budget [41].

Progredendo nel progetto, la stima dei tempi e il budget sono monitorati con l'aiuto del Burn-Down Chart e, se necessario, corretti.

Gli Sprint sono, in larga misura, aderenti alle linee guida della Guida Scrum da Schwaber e Sutherland [9]. Gli Scrum Team hanno suddiviso la riunione di Sprint Planning in due parti, proprio come descrive la Guida Scrum. Tuttavia, nella parte di pianificazione del lavoro durante la riunione Sprint Planning, la squadra si divide in due gruppi distinti, rispettivamente hardware e software, per sviluppare i piani dettagliati delle attività di sviluppo nel prossimo Sprint.

Il resto del ciclo di tre settimane procede con le riunioni Daily Scrum congiunte per le squadre di sviluppo hardware e software [41].

La particolarità introdotta dall'azienda è rendere pubblici uno o due degli incontri Daily Scrum durante uno Sprint in corso, in maniera che l'invito al meeting venga esteso anche a persone esterne al Team di sviluppo come il Product Owner. Questo ha il vantaggio di

consentire di monitorare ulteriormente lo stato di sviluppo rispetto allo Sprint Backlog [41].

Essendo i prodotti sviluppati per la maggior parte tangibili, questi hanno difficoltà ad essere sviluppati con la metodologia Scrum, troppo rigida sulle scadenze e sull'indipendenza degli incrementi di prodotto.

I Team di sviluppo per la gestione delle scadenze del progetto e per presentare i risultati a partner esterni utilizzano varie versioni dei grafici Gantt. Questo strumento, deriva dalle metodologie Plan-Driven e è molto importante a livello di macro planning. L'uso di questi grafici fornisce un coordinamento esterno delle squadre di sviluppo [41].

Alla fine dello Sprint, il Team di sviluppo effettua lo Sprint Review e lo Sprint Retrospective.

Allo Sprint Review, il Team di sviluppo presenta un report dove vengono descritte le funzionalità che sono state implementate nel prodotto in fase di sviluppo. Questo è dovuto al livello d'integrazione che hanno i vari componenti sviluppati in diversi Sprint.

Lo Sprint Retrospective non è molto preso in considerazione dall'azienda. Non viene infatti effettuato al termine di ogni Sprint, ma la riunione viene fissata ad-hoc solo se sono presenti problemi particolari da risolvere che coinvolgono tutti i membri del Team di sviluppo o c'è bisogno di identificare punti di miglioramento del processo [41].

6.6.2 Vantaggi e sfide

6.6.2.1 Vantaggi

Il successo dell'integrazione della metodologia Scrum all'interno di un processo di sviluppo di prodotto Stage-Gate è dovuta alla decisione del Top Management di adottarla nella fase di sviluppo. Questa decisione Top-Down ha portato l'organizzazione ad essere ben preparata all'innovazione introdotta e non ci sono stati freni burocratici [41].

Un vantaggio nell'uso dello Scrum è che il Team di sviluppo può gestire il proprio lavoro, anche se è legato ad una pianificazione di più alto livello, rappresentata da grafici Gantt che devono essere rispettati.

Un altro vantaggio dell'adottare questa metodologia è che si ha una buona pianificazione e nonostante un Product Backlog flessibile, si cercano di fare stime precise per avere una revisione affidabile del progetto alla fine dello Stage.

L'uso di diagrammi di Gantt riesce a dare una pianificazione abbastanza precisa per permettere agli sviluppatori di avere una linea guida nello sviluppo del prodotto.

6.6.2.2 Sfide

La metodologia di sviluppo di prodotto Agile, in particolare lo Scrum, è applicata solo alla fase di sviluppo vero e proprio del prodotto. Per le altre fasi si utilizza la metodologia tradizionale Stage-Gate. Questo limita la flessibilità del processo di sviluppo di prodotto. Nello Stage di sviluppo del concept, la natura non strutturata della generazione di idee limita molto il possibile uso dello Scrum [41].

Il design di prodotto, la selezione dei colori e le interfacce sono definite e congelate nei primi Sprint dello sviluppo del prodotto. Questo, semplifica e velocizza lo sviluppo del prodotto, ma è uno svantaggio per un'azienda che punta molto su style e design dei propri prodotti. Infatti, congelando queste caratteristiche all'inizio del processo di sviluppo, essa limita i feedback da parte dei clienti con il rischio di consegnare un prodotto poco aderente alle richieste del mercato [41].

La sfida maggiore nell'azienda è quella di stimare le risorse in termini di tempo, di sviluppatori e di budget anche per progetti in cui il Product Backlog e le User Story non sono molto definite. Questo è molto importante per il management in quanto deve stilare dei grafici Gantt e coordinare i vari Team di sviluppo.

Il Product Backlog è pensato come uno strumento flessibile anche per progetti che sono poco innovativi. In questi tipi di progetti, presenti nell'azienda studiata, il Product Backlog viene poco preso in considerazione e seguito poco dagli sviluppatori che preferiscono affidarsi all'esperienza e dalla conoscenza derivante da altri progetti. Inoltre per questi progetti i requisiti sono praticamente già fissati prima dello Stage di sviluppo. Questo va a scapito del mantenimento della flessibilità verso i cambiamenti e verso la maggior attenzione ai desideri dei clienti [41].

Una sfida dell'uso corretto della metodologia Scrum per lo sviluppo, specialmente per prodotti fisici, è la presentazione di un prodotto potenzialmente completo e vendibile ad ogni Sprint. Questo è dovuto alle dimensioni fisiche ed alla forte integrazione dei prodotti, spesso complessi, che rende quasi impossibile raggiungere il livello desiderato di maturazione attraverso uno Sprint. Allo stesso tempo, è difficile considerare una singola funzionalità senza considerare il prodotto globalmente. Un prodotto che include hardware e meccanica è chiaramente in conflitto con l'idea di avere qualcosa che può essere

sviluppato in una sola fase e al termine può essere lanciato sul mercato. Il prodotto spesso necessita di parecchi Sprint prima che diventi qualcosa di tangibile [41].

Un'altra sfida derivante dalla natura tangibile dei prodotti fisici è la divisione delle User Story in task in modo da creare Sprint indipendenti. Infatti l'integrazione delle varie parti del prodotto non permettono l'indipendenza degli Sprint [41].

Nello sviluppo di prodotti fisici, il Team è cross-funzionale. Questo Team di sviluppo possiede competenze software, hardware e meccaniche. Gli sviluppatori hanno inoltre interessi e campi di lavoro differenti e non necessariamente parlano una lingua comune, a differenza dei Team di sviluppo solo software che, pur avendo profili professionali diversi, ha comunque il software come campo di lavoro comune. Nei Team cross-funzionale la difficoltà è mantenere unito il gruppo e mantenere alta la motivazione. Queste difficoltà si notano specialmente nel Daily Scrum, nel quale alcuni sviluppatori si sono chiesti più volte se fosse utile ascoltare gli interventi di tutti, dal momento che essi non sono a conoscenza degli argomenti trattati [41].

Il bilanciamento tra gli obiettivi di sviluppo a breve e lungo termine è un'altra sfida che l'azienda ha dovuto affrontare. I cicli a breve termine come gli Sprint, consentono agli sviluppatori di concentrarsi sulle attività immediate e più importanti in un determinato momento, ma una comprensione comune sui piani di sviluppo a lungo termine e una visione del prodotto è fondamentale per lo sviluppo di prodotti di successo [41].

6.6.3 Classificazione del caso di studio

Nella classificazione del caso di studio sono stati presi in considerazione i progetti poco innovativi, essendo questi in numero molto maggiore rispetto agli altri. Inoltre sono quelli che si adattano meglio ad una pianificazione minuziosa propria della compagnia.

I requisiti vengono definiti come User Story che la squadra di sviluppo porta a compimento. C'è un controllo dell'avanzamento da parte del Product Owner che alla fine di ogni Sprint, insieme al Team, ridefinisce le priorità. La metodologia Agile rimane comunque confinata alla fase di Sviluppo, anche se è in programma un'implementazione a livello anche di sviluppo del concetto.

Essendo progetti di routine, i requisiti sono molto molto simili tra di loro e l'esperienza porta a colmare i divari di informazioni.

Un altro fattore che limita il seguire delle idee del cliente anche in questo tipo di progetti, è il fatto che la metodologia Scrum sia implementata solo a livello di sviluppo e non a livello di test, produzione e lancio. Questo comporta un ritardo tra l'ultimo feedback del cliente e la consegna del prodotto e ciò potrebbe decretarne il non completo successo sul mercato.

Inoltre il design del prodotto è congelato nei primi due Sprint del processo di sviluppo, il che limita molto il feedback da parte dei clienti per quanto riguarda il design e lo style del prodotto.

Il controllo e la rigidità dei Gate è molto sentito in questo caso di studi. Infatti rimane molto importante l'attuazione della metodologia Stage-Gate. L'efficacia dei controlli durante il progredire dello sviluppo non avviene solo nello Sprint Review, ma anche durante uno o due Daily Scrum aperti in ogni Sprint. Per il controllo del budget e delle tempistiche per lo sviluppo e il coordinamento dei Team si utilizzano diagrammi a blocchi e Gantt tipici delle metodologie tradizionali.

In questo caso, sono stati assegnati dei valori alle grandezze prese in considerazione in questo studio, riportate nella tabella sottostante:

Tabella 6.7 Classificazione del caso di studi Compagnia A

Rigidità dei Gate	Alta
Presenza di feedback	Bassa

6.7 Caso LEGO Education

LEGO Education è un'unità commerciale del gruppo LEGO, responsabile dello sviluppo del software e prodotti fisici a scopo educativo per i bambini in età prescolastica e superiore.

LEGO Education dal 1980 lavora con insegnanti e specialisti educativi per offrire esperienze di apprendimento tramite il gioco che porta i bambini ad un'esperienza di apprendimento divertente. LEGO Education ha un'ampia gamma di risorse didattiche fisiche e digitali che incoraggiano gli studenti a pensare in modo creativo, ad essere sempre motivati ed a liberare il loro potenziale per modellare il proprio futuro [42].

Le soluzioni per l'insegnamento e l'apprendimento pratico destinate a scuola elementare e secondaria ispirano l'interesse per la scienza, la tecnologia, l'ingegneria e la matematica. Queste sono basate sul sistema LEGO® per l'apprendimento giocoso, combinato con materiale didattico adatto alla scuola e alle risorse digitali.

Nel 2011, LEGO Education ha avviato un progetto altamente innovativo, Story Starter che è una soluzione educativa con lo scopo di stimolare i ragazzi delle scuole elementari alla scrittura e alla lettura. A causa dell'introduzione di componenti digitali nel prodotto, la direzione ha provato un metodo di sviluppo ibrido totalmente nuovo in LEGO, che si è rivelato avere significativi vantaggi nel progetto ed infine, ha portato al lancio di questo nuovo prodotto di grande successo 12 mesi dopo [31].



Figura 6.18 LEGO Educational [42]

6.7.1 Lo sviluppo di prodotto

L'iniziativa che è stata sviluppata da LEGO è chiamata Story Starter.

Inizialmente il progetto è stato gestito utilizzando il modello di sviluppo di prodotto tradizionale, ormai assodato in LEGO. Questo modello, però, non si adattava bene alle iterazioni di progettazione richieste dai continui feedback dei clienti. Per aiutare gli sviluppatori, la società ha messo a disposizione un software come strumento per la documentazione digitale, per tenere traccia delle esperienze di apprendimento dei bambini. Questo non ha però portato a notevoli incrementi di iterazioni con i clienti [31]. Il prodotto in sviluppo, Story Starter, è un prodotto fisico che incorpora anche una parte software. Per lo sviluppo di questa parte, il team ha dovuto lavorare in collaborazione con i membri del Digital Solutions di LEGO. Questi hanno portato con loro la metodologia di sviluppo Agile, di cui ne avevano già provato i pregi nell'incremento della velocità di sviluppo e nella miglior aderenza del prodotto finale ai desideri del cliente [31].

Dopo aver visto i vantaggi della metodologia Agile per lo sviluppo del software, il Team di sviluppo hardware ha deciso di provare questo nuovo approccio per la gestione dell'intero progetto.

Il Management però, non ha voluto usare esclusivamente l'Agile come metodologia di sviluppo di prodotto perché non voleva perdere i vantaggi che dava la pianificazione e il controllo del progetto nello Stage Gate [31].

Si è deciso quindi di applicare un metodo ibrido nel quale si utilizzava l'Agile come pianificazione micro planning e per la gestione dei feedback dei futuri clienti e lo Stage-Gate come pianificazione macro planning e monitoraggio del lavoro [31].

Dopo il successo della squadra pilota di Agile-Stage-Gate con Story Starter, questa metodologia è stata applicata anche al progetto "More to Math" per lo sviluppo del prodotto, per aiutare i bambini delle scuole elementari ad imparare a risolvere i problemi di matematica. Anche in questo caso la metodologia ha avuto successo, riuscendo a sviluppare un prodotto velocemente ed aderente alle richieste dei clienti. La metodologia Agile-Stage-Gate è stata formalizzata per essere condivisa con altri dipartimenti nello sviluppo di prodotti sia fisici che software. Tutte le release di LEGO Education nel 2016 sono state sviluppate utilizzando questo nuovo approccio ibrido [31].

6.7.1.1 Framework di sviluppo di prodotto

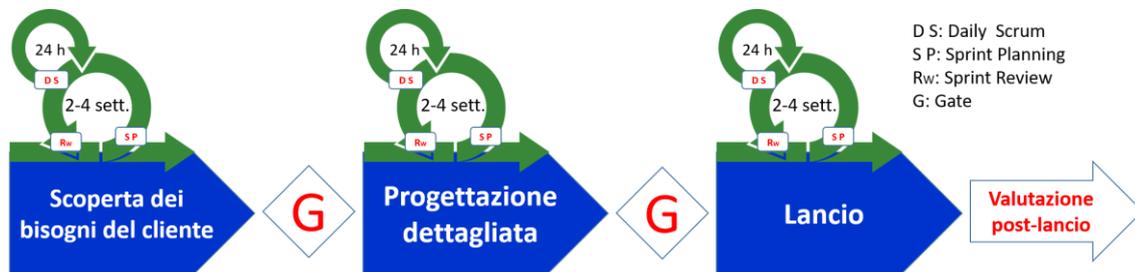


Figura 6.19 Metodologia di sviluppo di prodotto in LEGO

La metodologia Agile-Stage-Gate adottata in LEGO, è dotata, come detto sopra, di una parte di micro planning Agile e una di macro planning Stage-Gate.

Nella parte di operativa di micro planning, si utilizza la metodologia Agile Scrum. Si utilizzano tutti gli strumenti riportati nella guida Scrum: gli Sprint, il Daily Scrum di 15 minuti ogni giorno, le lavagne visive per la gestione dell'avanzamento, il report di attività quotidiane (importanti per i Burn-Down Chart), il Backlog con le priorità per le attività, incontri di pianificazione dei Sprint e gli Sprint Review [31].

Una regola molto importante per il successo della metodologia è che il Team sia dedicato totalmente al progetto e lavori nella stessa struttura per poter comunicare direttamente [31].

Durante la fase di sviluppo del progetto, gli insegnanti di più di 50 scuole hanno partecipato ai test sui prodotti. Gli Sprint Review sono stati condotti con rappresentanti interni dei clienti e sono stati portati a termine studi coinvolgendo i clienti primari (scuole) e facendo test con gli utilizzatori finali, i bambini [31].

LEGO, ha dato una definizione di “Fatto” per il risultato di uno Sprint diverso da quello scritto nella guida Scrum. Infatti uno Sprint “Fatto” nella metodologia di sviluppo in LEGO è quando una check-list di dieci elementi viene completata da tutti i membri del Team, dall'addetto al Controllo Qualità e dal Product Owner, dichiarando che tutti i criteri di accettazione dei prodotti sono stati soddisfatti e che tutte le azioni aperte sono state completate, consegnate o chiuse [31].

6.7.2 Vantaggi e sfide

6.7.2.1 Vantaggi

Dopo l'implementazione di questo nuovo modello ibrido Agile-Stage Gate, il progetto Story Starter ha registrato una notevole accelerazione e, in breve tempo, ha avuto maggiori progressi rispetto ai progetti precedentemente intrapresi [31].

Il cambiamento più importante per la squadra è stato l'immediato aumento della produttività. I membri del team hanno imputato tale incremento principalmente ad una migliore comunicazione della squadra e ad una diminuzione delle incomprensioni. Inoltre i membri del team hanno sperimentato un miglioramento del flusso di lavoro giornaliero, in parte grazie alla consapevolezza che i loro problemi attuali sarebbero stati facilmente risolti al Daily Scrum la mattina dopo [31].

Agile-Stage-Gate ha dato un importante strumento di feedback da parte degli utilizzatori finali, specialmente negli Sprint Review nei quali sono stati coinvolti proprio gli utilizzatori finali [31].

6.7.2.2 Sfide

Nello sviluppo di prodotti fisici, a differenza dei prodotti software, ci sono dei tempi morti di attesa, come ad esempio la consegna di pezzi ordinati che non fanno parte dello sviluppo del prodotto. In questo lasso di tempo, il Team può lavorare su altri progetti che nelle aziende manifatturiere sono sempre presenti [31].

Un'altra sfida proposta dallo sviluppo di prodotti fisici con questa metodologia è il bisogno di un Team cross-funzionale. A differenza dello sviluppo di software, per il quale bastano poche persone competenti in programmazione, nell'hardware è necessario che lavorino al prodotto persone con competenze molto diverse, dalla meccanica, all'automazione, al design [31].

Per la metodologia Agile è importante che le persone si dedichino totalmente ad un solo progetto, ma per lo sviluppo hardware, sviluppatori che si occupano per esempio del marketing, della gestione delle operation e delle vendite, non riescono a dedicarsi al 100% ad un progetto soltanto. In LEGO si è giunti ad alcuni compromessi, ad esempio un addetto del marketing potrà dedicarsi ad un singolo Sprint solo nel momento di

individuazione dei requisiti o del feedback da parte dei clienti. Questo assicura al Team di sviluppo una dedizione totale al progetto per quello Sprint, senza però gravare come peso all'azienda nel momento in cui non sussiste il bisogno. Il compromesso ha però anche un effetto negativo le persone che lavorano “una tantum” al progetto, non avranno lo stesso entusiasmo di uno sviluppatore che l’ha visto nascere e lo segue fino al lancio del prodotto [31].

6.7.3 Classificazione del caso di studio

In questo caso la presenza di feedback da parte dei clienti è molto importante. Infatti il passaggio dal modello Stage-Gate al modello ibrido Agile-Stage-Gate è dovuto appunto a questa particolarità del prodotto che doveva essere sviluppato. Sono stati coinvolti direttamente nello sviluppo maestre di cinquanta istituti e molti bambini delle scuole elementari.

Il management invece ha voluto mantenere i vantaggi di una metodologia Plan-Driven per avere una pianificazione a lungo termine ed un controllo sull’evoluzione del prodotto. Dato il grado di innovazione del prodotto e il bisogno che sia aderente ai bisogni del cliente, è molto importante che i Gate all’interno di questo progetto siano flessibili per permettere la validazione di un progetto con dei prerequisiti incompleti e non congelati. Infatti in questo caso di studi, i Gate sono presenti e svolgono il loro lavoro di valutazione sul proseguimento dello sviluppo.

La classificazione del caso di studio secondo le grandezze scelte, è riportato nella tabella sottostante:

Tabella 6.8 Classificazione del caso di studi LEGO Education

Rigidità dei Gate	Media
Presenza di feedback	Media

7 ANALISI DELLA CLASSIFICAZIONE DEI CASI DI STUDIO

In questo elaborato sono stati analizzati diversi casi di studio scelti come best practice e come esempi significativi dei diversi framework di sviluppo di prodotto adattati al contesto aziendale. Tali casi sono stati classificati ed è stato assegnato un valore alle grandezze scelte nel Capitolo 5.

Come spiegato nel Capitolo 5, le due grandezze proposte possono formare una matrice Rigidità dei Gate vs Presenza di feedback da parte dei clienti. In questa matrice si possono individuare tre punti, classificando le metodologie Stage-Gate, Agile e Agile-Stage-Gate secondo la tabella riportata sotto.

Tabella 7.1 Valori di rigidità dei Gate e presenza dei feedback dei clienti assegnati in seguito all'analisi delle metodologie riportate in letteratura

Metodologia	Rigidità dei Gate	Presenza di feedback
Stage-Gate	Alta	Nulla
Agile	Nulla	Alta
Agile-Stage-Gate	Media	Media

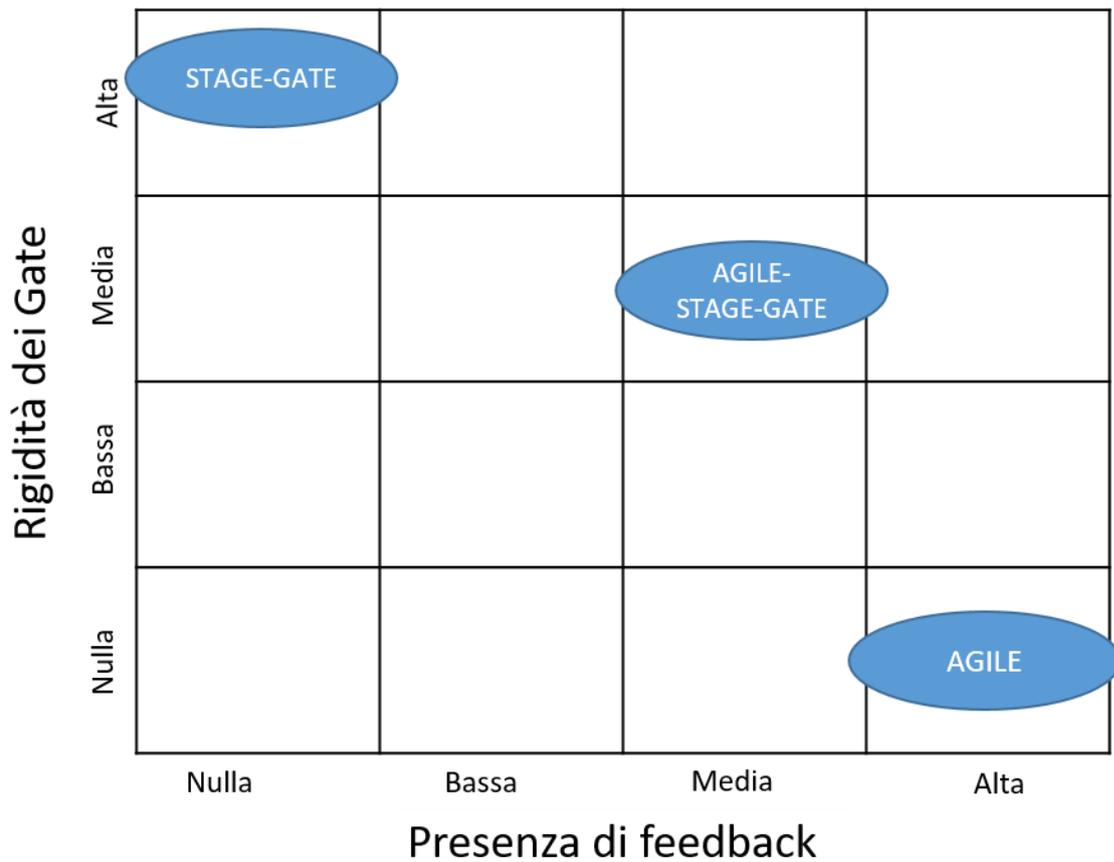


Figura 7.1 Posizionamento dei punti teorici nella matrice rigidità dei Gate vs presenza dei feedback da parte dei clienti

In questa matrice si può notare come il modello Agile-Stage-Gate non sia perfettamente a metà tra l'Agile e lo Stage-Gate, ma sia una metodologia ibrida che riesce a combinare la maggior parte delle caratteristiche positive dei due.

7.1 Posizionamento dei casi di studi nella matrice

Dall'analisi delle best practice si possono posizionare i vari casi di studio, a cui è stata assegnata una classificazione di rigidità dei Gate e presenza di feedback da parte dei clienti (Tabella 7.2), nella matrice riportata sopra. Questi, si dispongono in modo simile ai punti derivanti dalle metodologie formalizzate in letteratura, classificate in precedenza.

Tabella 7.2 Valori di rigidità dei Gate e presenza dei feedback dei clienti assegnati in seguito all'analisi del caso di studio

Caso di studi	Rigidità dei Gate	Presenza di feedback
Caso Wikispeed	Nulla	Alta
Caso ThermoValves	Media	Media
Caso Siemens Westinghouse	Alta	Nulla
Caso Marel GRB	Bassa	Alta
Caso Andritz Hydro	Alta	Nulla
Caso Compagnia A	Alta	Bassa
Caso LEGO Educational	Media	Media

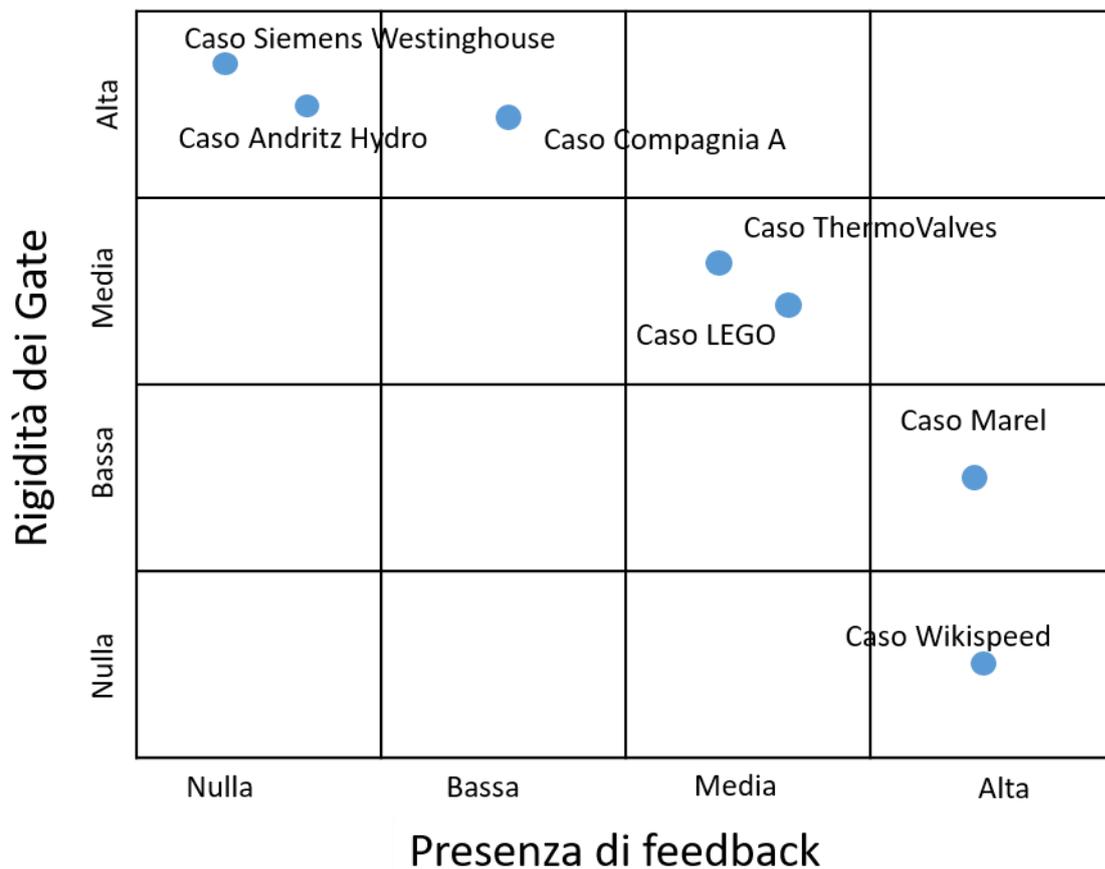


Figura 7.2 Posizionamento dei casi di studio nella matrice rigidità dei Gate vs presenza dei feedback da parte dei clienti

Si può notare che questi si posizionano in uno spazio che può essere rappresentato da una curva, in quanto c'è una contrapposizione tra l'aver feedback continui da parte del cliente che modificano le richieste iniziali, e l'aver dei requisiti congelati all'inizio dello sviluppo che possano essere controllati durante un Gate rigido, evitando che i progetti che non portano valore all'azienda vengano continuati.

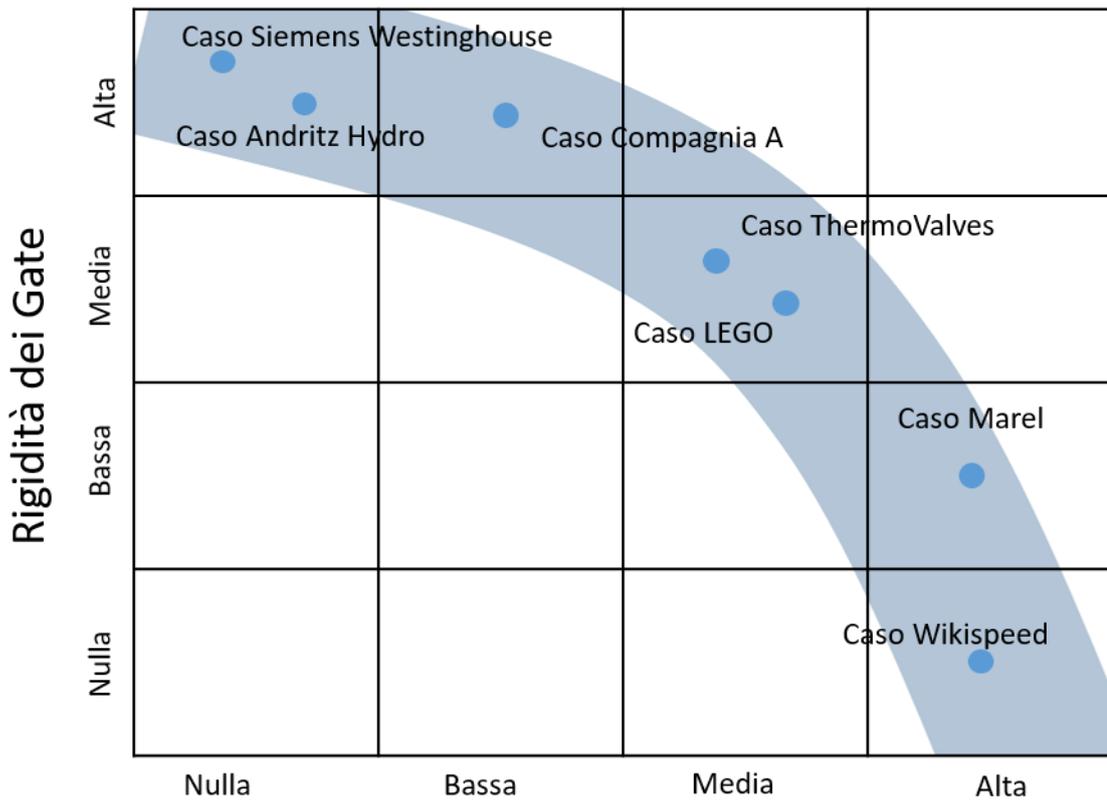


Figura 7.3 Curva comprendente le best practice classificate

Questa curva inoltre, rappresenta il livello di completezza e immutabilità dei requisiti iniziali. Infatti a mano a mano che ci si avvicina al punto di Stage-Gate, i requisiti sono sempre più definiti. Partendo dal caso Siemens Westinghouse nel quale i requisiti sono fissati da una normativa dettata dal Dipartimento dell'Energia Americana e nei diversi progetti cambia solo l'efficienza richiesta, si passa attraverso i casi LEGO Educational e ThermoValves in cui i requisiti sono abbastanza chiari per definire una pianificazione, ma non così completi da acconsentire allo sviluppo innovativo di seguire le richieste dei clienti e si giunge infine al caso Wikispeed nel quale gli unici requisiti fissati sono che l'automobile sia aderente alle norme di sicurezza vigenti e riesca a fare 100 miglia al gallone.

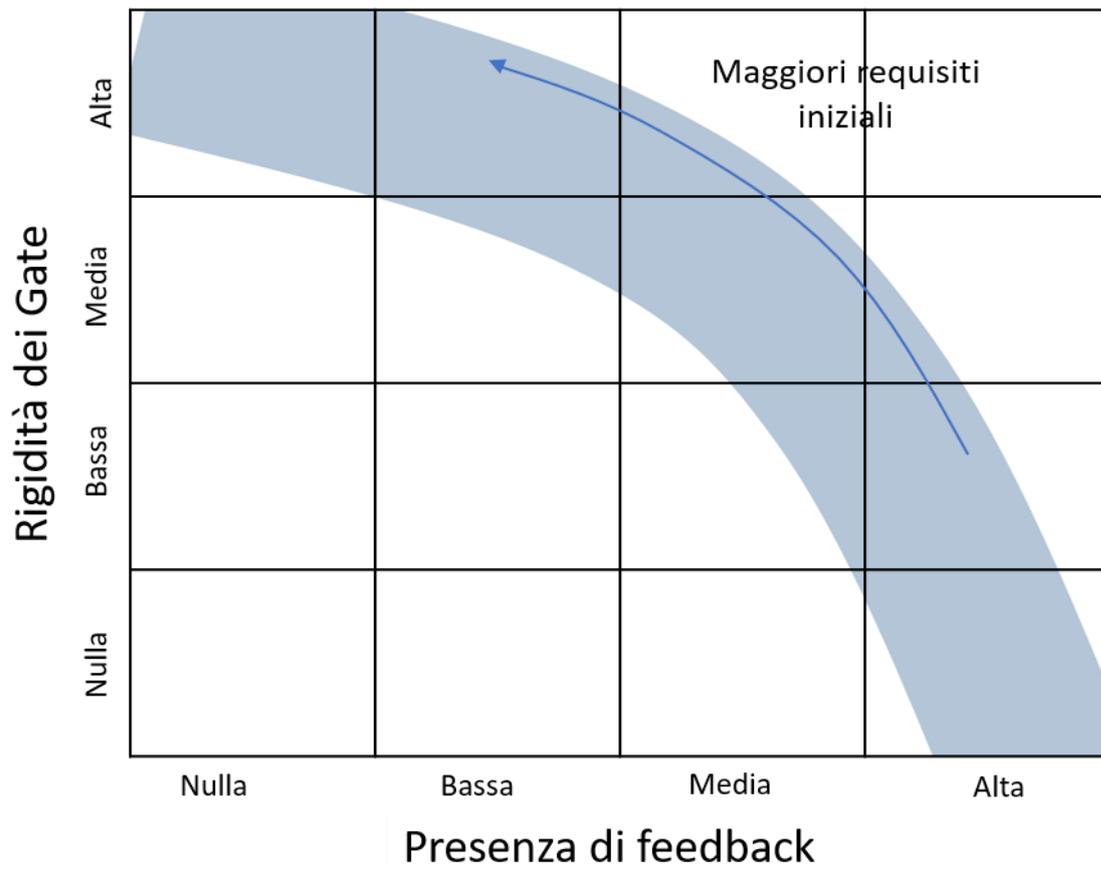


Figura 7.4 Rappresentazione della parametrizzazione della curva in quantità di requisiti iniziali

7.2 Caratteristiche per diversi livelli di definizione dei prerequisiti

Da questo studio di best practice inoltre, si possono identificare zone della curva in base alla quantità di requisiti congelati all'inizio dello sviluppo. Questo può essere utile per identificare il miglior framework adattato al livello di requisiti fissati e congelati al momento dell'inizio dei lavori.

A seconda dell'incertezza dei requisiti, dell'efficacia dei Gate e dei feedback richiesti ai clienti, nei casi di studio la metodologia Agile-Stage-Gate è stata modificata e adattata per rispondere alle esigenze dello sviluppo dei prodotti.

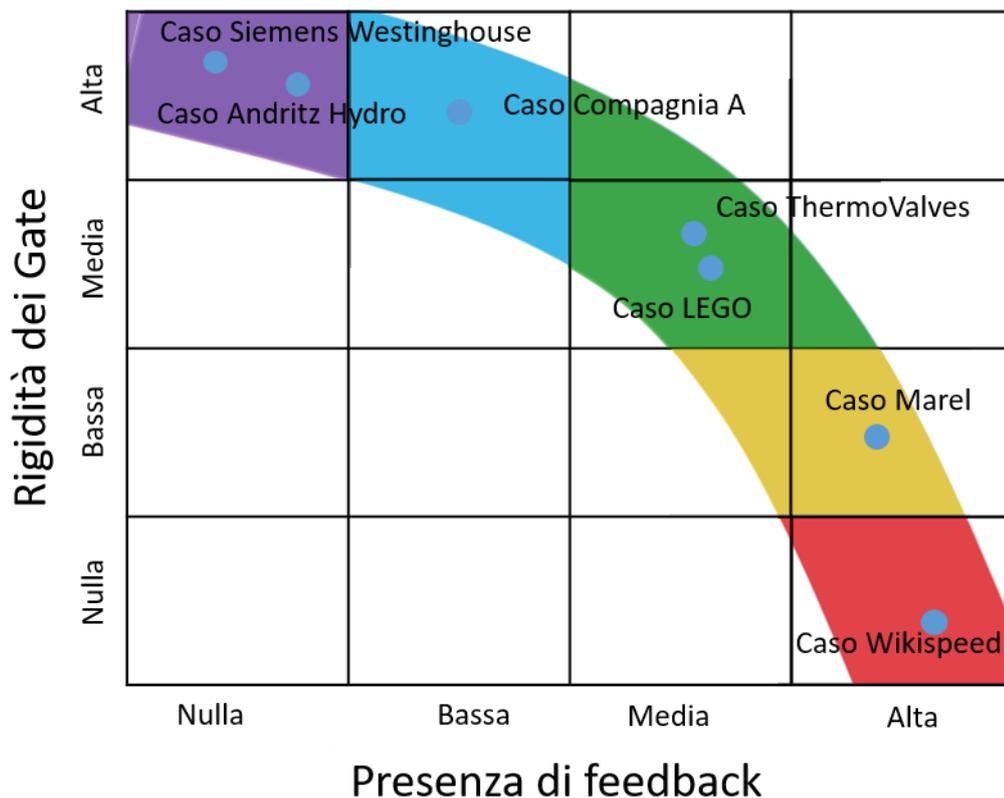


Figura 7.5 Posizionamento delle best practice secondo la suddivisione del livello di completezza dei prerequisiti

Cooper e Sommer [26] hanno formalizzato un framework chiamandolo Agile-Stage-Gate, ricavandolo dopo aver analizzato una serie di casi di studio [26]. Questo framework generale non può essere applicato ready-to-use in tutti i casi. Analizzando i casi di studio precedentemente citati, si possono identificare alcune caratteristiche che modificano il

framework proposto da Cooper e Sommer [26], dipendenti dal livello di definizione e dal congelamento dei requisiti all'inizio del processo di sviluppo del prodotto.

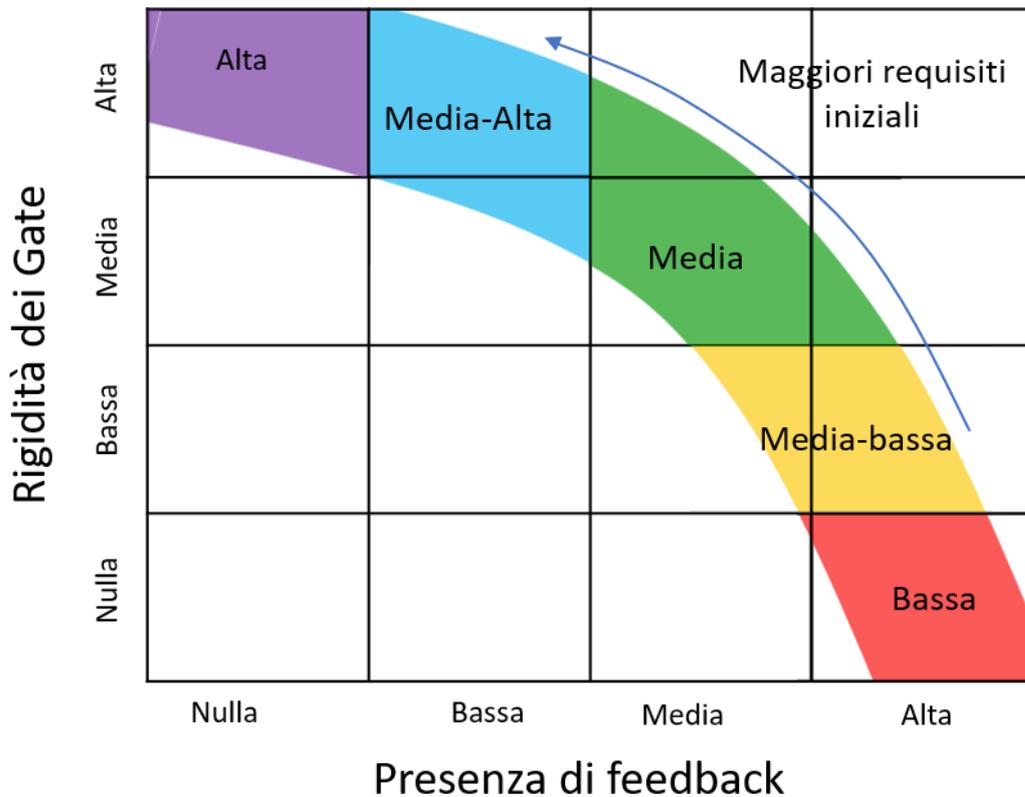


Figura 7.6 Divisione della curva nel livello di completezza dei prerequisiti all'inizio dello sviluppo

Si possono identificare come riportato in Figura 7.5 e Figura 7.6 cinque regioni a seconda del livello di definizione dei prerequisiti iniziali:

- Alta definizione dei requisiti iniziali
- Medio-alta definizione dei requisiti iniziali
- Media definizione dei requisiti iniziali
- Medio-bassa definizione dei requisiti iniziali
- Bassa definizione dei requisiti iniziali

7.2.1 Alta definizione dei requisiti iniziali

Nel caso non ci sia incertezza nei requisiti, si ha un bisogno limitato di feedback da parte dei clienti, in quanto i requisiti sono già stati definiti e congelati all'inizio del progetto. Questo comporta che la metodologia che più si adatta allo sviluppo in queste condizioni è la metodologia Stage-Gate la quale, se applicata in modo rigoroso, aumenta al massimo

l'efficienza dello sviluppo del prodotto, minimizzando i tempi di sviluppo attraverso controlli rigorosi nei Gate e milestone per evitare ritardi e fare in modo che solo i progetti che possano portare valore all'azienda vengano portati avanti ed a termine. I progetti che maggiormente si trovano in questa situazione sono i lavori di routine e i progetti nei quali i requisiti sono già fissati a livello legislativo, come nel caso di Siemens Westinghouse, in cui gli unici prerequisiti chiesti al cliente sono la tipologia d'impianto o di turbina e l'efficienza termica della stessa. La metodologia Stage-Gate, in queste condizioni, è uno strumento potente per le aziende, ma ha qualche svantaggio per quanto riguarda la pianificazione. Nella maggior parte dei casi questa viene preparata dal Management, che certe volte non hanno sufficiente conoscenza sul miglior modo di procedere nello sviluppo di un prodotto a livello operativo. In questo caso viene in aiuto il sistema Agile. Modificando lo Scrum, come nel caso Andritz Hydro, questo risulta essere, anche per progetti con requisiti molto definiti che non hanno bisogno di feedback da parte dei clienti, un importante strumento di micro planning. In questo caso di studi vengono usati alcuni degli strumenti dello Scrum che incrementano l'efficienza della metodologia Stage-Gate.

7.2.2 Media-Alta definizione dei requisiti iniziali

In questo caso si ha una definizione dei requisiti tale che i Gate sono uno strumento importante per decretare il proseguimento o meno del progetto. I metodi Agile, come riscontrato nei progetti di routine del caso della società A, vengono implementati principalmente a livello di sviluppo del prodotto, in quanto il Backlog è un documento che deriva da uno studio dei prerequisiti non totalmente definiti e congelati. Avendo gli sviluppatori la maggior parte dei requisiti fissati e una visione globale del prodotto, non sono necessari molti feedback da parte del cliente e il Product Owner può fare da tramite. Questo comporta una serie di vantaggi e svantaggi. Uno svantaggio è che durante il processo di sviluppo, se il Product Owner non è molto presente e non ha capacità di comunicare, gli sviluppatori non riescono a percepire i desideri del cliente ed a modificare i requisiti ancora non presenti o ambigui, per sviluppare un prodotto aderente alle richieste dell'utilizzatore finale. Lo sviluppo porterebbe comunque a un prodotto finito ma che potrebbe non essere di totale successo. Un altro svantaggio si presenta una volta sviluppato il prodotto, poiché questo viene congelato per la fase di test, produzione e

lancio. Ciò comporta un ritardo tra i desideri del cliente al momento dello sviluppo e i bisogni al momento del lancio e quindi dell'uso da parte dell'utilizzatore finale.

Un vantaggio è invece il fatto che lo sviluppatore riesce ad avere un quadro abbastanza definito, oltre che una pianificazione dettagliata nel momento della partenza dello sviluppo. Questo comporta meno stress per gli sviluppatori che hanno già in mente il punto d'arrivo ed una programmazione abbastanza dettagliata dei tempi e delle risorse da impiegare.

Questa metodologia viene usata in caso di uno sviluppo del prodotto non troppo innovativo ma che per alcuni requisiti deve essere aderente alle richieste del cliente. Infatti, come viene citato, nella Società A il Backlog è pensato per essere uno strumento flessibile, anche se la sua caratteristica non viene sfruttata in quanto, durante i progetti di routine, questi sono abbastanza ripetitivi.

7.2.3 Media incertezza dei requisiti

Con questa completezza dei prerequisiti è possibile poter costruire una pianificazione di massima del progetto e si possono usare con efficacia gli strumenti della metodologia Stage-Gate per valutare il procedere del lavoro. I Gate sono abbastanza flessibili per permettere una facilità di modifica della rotta durante il proseguire del progetto, ma rigidi per quanto riguarda la tempistica ed il budget assegnato inizialmente. La media definizione dei requisiti, inoltre, fa in modo che lo sviluppo del prodotto possa procedere restando aderente alle richieste attuali del cliente. I casi di studio presenti in questa regione, riportati in Figura 7.5, sono impegnati in progetti innovativi, anche se hanno una visione di alcuni requisiti essenziali che il prodotto finito deve avere.

In questo livello di prerequisiti fissati si può applicare la metodologia di sviluppo di prodotto ibrida proposta da Cooper e Sommer nel loro articolo [26].

7.2.4 Media-Bassa definizione dei requisiti iniziali

In questo livello di completezza dei requisiti iniziali, non è facile integrare le metodologie Agile allo sviluppo di prodotti fisici. Nel caso studiato di Marel GRB infatti, si è cercato di applicare Scrum al processo di sviluppo di un prodotto molto innovativo con certe difficoltà. In Marel GRB il Team ha deciso di non continuare l'implementazione dello

Scrum in quanto, così com'era strutturato, presentava difficoltà di applicazione. Le persone abituate a lavorare con la metodologia Stage-Gate riescono con fatica a concepire il fatto di spezzare il lavoro e di ripianificare, secondo le richieste del cliente, in qualsiasi stadio dello sviluppo. Inoltre la mancanza di una visione data dalla definizione dei requisiti prima dello sviluppo è un limite importante per gli sviluppatori.

In queste situazioni le figure importanti, quali lo Scrum Master e il Product Owner, sono fondamentali per lo sviluppo del prodotto con requisiti così incerti. Lo Scrum Master ha il ruolo fondamentale di guidare il Team di sviluppo nella pianificazione dello Sprint e nel Daily Scrum, mentre il Product Owner deve avere una visione globale del prodotto e guidare il Team di sviluppo nell'assecondare le richieste del cliente. Avendo i prerequisiti così poco definiti, è difficile fissare dei Gate per decidere se un progetto deve essere continuato o chiuso. Molte volte, infatti, si aggiungono o modificano requisiti a ogni iterazione.

La metodologia utilizzata nel caso di studio è una metodologia ibrida Agile-Stage-Gate, all'interno della quale l'Agile viene adottata per la gestione di intere parti di sviluppo. In queste parti non si utilizzano strumenti della metodologia Stage-Gate in quanto il progetto è incentrato completamente a seguire l'evolversi dei desideri dei clienti che forniscono feedback continui.

7.2.5 Bassa definizione dei requisiti iniziali

In questo caso si parla di una metodologia molto vicina a quella teorica dell'uso dell'Agile. Il prodotto è talmente innovativo ed i requisiti sono talmente incerti, che non si riesce a fare una pianificazione e a mettere dei paletti per usare i Gate in modo appropriato. In questo caso i Gate addirittura non vengono utilizzati e il feedback da parte dei clienti diventa un punto focale su cui basarsi per procedere con il progetto ed avere una conferma della validità del lavoro.

Con questa incertezza il Team di sviluppo viene caricato di tutta la responsabilità sulla pianificazione, essendo solo a breve termine, e dato che i normali progetti non hanno tempo e budget infiniti, anche della riuscita nel lancio di un prodotto completo nei tempi massimi concessi. Difficilmente questa metodologia si applica allo sviluppo di prodotti fisici, infatti il caso Wikispeed è risultato usare con successo questa metodologia, ma con dei limiti non indifferenti nel momento della consegna. Alla data di scadenza del progetto,

fissata per la valutazione e la conseguente premiazione dell'automobile migliore, Wikispeed non era un'autovettura completa. Infatti non resisteva alle intemperie ed era stretta per il passeggero. Questi difetti portano a non aver centrato l'obiettivo di sviluppo del prodotto nel tempo prestabilito. Inoltre, come la maggior parte delle metodologie Agile, i membri del Team devono essere competenti e con un talento sopra la media degli sviluppatori, per riuscire nello sviluppo del prodotto. Per i prodotti fisici, inoltre, c'è il problema dei tempi morti per la costruzione di un prototipo o dei test, in cui lo sviluppo si arresta e il Team dedicato esclusivamente a quel progetto si ferma.

CONCLUSIONI

L'analisi delle varie best practice in relazione alle metodologie di sviluppo di prodotto presenti in letteratura, ha mostrato come i framework sono stati modificati per adattarsi ai vari livelli di definizione dei prerequisiti.

Una nota importante nell'utilizzo dei vari framework di sviluppo di prodotto, è che è la metodologia a dover adattarsi al prodotto sviluppato, non il contrario.

In questo elaborato si sono potute classificare le varie metodologie di sviluppo di prodotto presenti in letteratura ed i casi di studio proposti, inserendoli in una matrice le cui grandezze sono: la rigidità dei Gate e la presenza di feedback da parte dei clienti lungo il processo di sviluppo di prodotto.

Queste grandezze sono state selezionate poiché la prima è la caratteristica principale della metodologia Plan-Driven, nella quale lo sviluppo di prodotto segue una pianificazione molto precisa e la rigidità dei Gate serve a controllare il lavoro del Team di sviluppo. Infatti, i Gate servono come momento di revisione del progetto in cui si verifica se potrà avere successo sul mercato e si deciderà se mantenerlo attivo o chiuderlo.

La seconda grandezza, la presenza di feedback da parte dei clienti, invece è la caratteristica principale delle metodologie di sviluppo del prodotto Agile. I feedback da parte dei clienti, sono il motore dello sviluppo di prodotto in quanto danno indicazioni chiare su come svilupparlo affinché sia più aderente possibile alle esigenze di mercato e quindi di sicuro successo.

Cooper e Sommer [26] hanno cercato di definire un framework che riuscisse ad integrare la metodologia Agile e Plan-Driven. Hanno chiamato la metodologia Agile-Stage-Gate, la quale riesce a incorporare i benefici della metodologia Plan-Driven e Agile eliminando i maggiori svantaggi. Una metodologia ibrida però non riesce ad integrare tutti i benefici delle due, infatti i Gate risultano più flessibili per permettere la ridefinizione dei requisiti, mentre si ha un congelamento dei requisiti prima della produzione per i prodotti fisici, con conseguente impossibilità di seguire i desideri attuali dei clienti.

I casi di studio proposti in questo elaborato, classificati secondo le grandezze proposte, possono essere disposti lungo una curva parametrizzata secondo il livello di definizione dei prerequisiti.

La curva è generata dalla contrapposizione tra le due grandezze con cui sono state classificate le metodologie ed i casi di studio presenti in letteratura. Infatti non è possibile

avere contemporaneamente feedback da parte dei clienti e una definizione dei requisiti tale da avere dei Gate molto rigidi.

Questa curva può essere divisa in cinque regioni e si possono analizzare le caratteristiche dei framework utilizzati dai casi di studio, a seconda del livello di definizione dei prerequisiti:

- **Alta definizione dei requisiti iniziali.** Si ha un totale congelamento dei requisiti di prodotto all'inizio dello sviluppo. La pianificazione è minuziosa, conoscendo bene il prodotto, è possibile avere Gate efficaci per valutare l'avanzamento dello sviluppo di prodotto. Viene adottata principalmente per progetti di routine e non innovativi.
- **Medio-alta definizione dei requisiti iniziali.** Si ha il congelamento di buona parte dei requisiti di prodotto all'inizio dello sviluppo. Si cerca, per il completamento, di avere feedback da parte dei clienti e Lead User. Anche questa metodologia viene adottata per progetti di routine, più focalizzati però sulle esigenze del cliente.
- **Media definizione dei requisiti iniziali.** Si usa questa metodologia per progetti innovativi. Ogni Stage è strutturato in modo da avere un continuo feedback da parte dei clienti ed i requisiti di prodotto vengono congelati in maniera graduale durante lo sviluppo fino al momento della produzione, quando sono completamente congelati. I Gate hanno un ruolo di controllo sull'avanzamento del lavoro, sul budget e sulle scadenze, ma riescono ad essere flessibili per permettere l'avanzamento di progetti con un certo grado di incertezza.
- **Medio-bassa definizione dei requisiti iniziali.** In questo caso si ha una pianificazione di massima iniziale. Si hanno dei requisiti iniziali che possono mutare durante il progetto. Viene impiegato per progetti molto innovativi in cui si ha solo una visione di massima del prodotto. Si usa per lo sviluppo la metodologia Agile per progredire restando più aderenti possibile alle richieste del cliente, fornite con feedback costanti.
- **Bassa definizione dei requisiti iniziali.** I progetti in questo caso sono estremamente innovativi. Si ha solo un'idea di prodotto da sviluppare e pochissime linee guida, sotto forma di requisiti iniziali da seguire. Si usa esclusivamente la metodologia Agile, si fissa all'inizio dello sviluppo un budget e una tempistica e si lavora a stretto contatto con i clienti fino a che uno dei due parametri fissati non viene superato.

Escludendo i casi Siemens Westinghouse Power Generation e Wikispeed che adottano una metodologia di sviluppo rispettivamente Stage-Gate e Agile, gli altri hanno adottato una metodologia ibrida. L'uso della metodologia Agile-Stage-Gate modificata, secondo le rispettive caratteristiche di prodotto, ha confermato che la metodologia Agile contiene al suo interno ottimi strumenti di micro planning che s'integrano perfettamente con la metodologia Stage-Gate usata per il macro planning.

Più ci si avvicina alla metodologia Stage-Gate, più si perdono le caratteristiche d'investigazione e risposta alle esigenze dei clienti durante lo sviluppo, fondamentale invece nell'Agile. Più ci si avvicina all'Agile, invece, più si perdono le caratteristiche della rigidità dei Gate e minuziosa pianificazione di tutto il progetto proprie della metodologia Stage-Gate.

BIBLOGRAFIA

- [1] R. G. Cooper, “Stage-Gate Systems : A New Tool for Managing New Products,” no. June, 1990.
- [2] R. G. Cooper, “Perspective: The Stage-Gate,” pp. 213–232, 2008.
- [3] K. T. Ulrich, S. D. Eppinger, and R. Filippini, *Progettazione e sviluppo prodotto*, McGraw-Hil. 2007, 2007.
- [4] D. W. Unger, “Product Development Process Design: Improving Development Response to Market, Technical, and Regulatory Risks,” 2003.
- [5] H. Chesbrough, “Open Innovation : A New Paradigm for Understanding Industrial Innovation,” 2006.
- [6] R. G. Cooper, “What ’ s Next ? : After Stage - Gate,” 2014.
- [7] B. Kent *et al.*, “Manifesto for Agile Software Development,” 2001. [Online]. Available: <http://agilemanifesto.org/>.
- [8] A. Begel and N. Nagappan, “Usage and perceptions of Agile software development in an industrial context: An exploratory study,” *Proc. - 1st Int. Symp. Empir. Softw. Eng. Meas. ESEM 2007*, pp. 255–264, 2007.
- [9] K. Schwaber and J. Sutherland, “The Scrum Guide The Definitive Guide to Scrum :,” no. July, 2016.
- [10] H. Takeuchi and I. Nonaka, “The new new product development game,” pp. 137–147, 1986.
- [11] “<http://www.excidion.com>,” 2014. .
- [12] Rkandha, “Agile Development Framework,” 2012. [Online]. Available: http://wiki.expertiza.ncsu.edu/index.php/CSC/ECE_517_Fall_2012/ch2a_2w6_ar .
- [13] L. Williams and A. Cockburn, “Agile Software Development: It’s about Feedback and Change,” *IEEE Softw.*, 2003.
- [14] M. A. Awad, “A Comparison between Agile and Traditional Software Development Methodologies,” *Univ. West. Aust.*, vol. 1, pp. 1–300, 2005.
- [15] B. Boehm and P. Papaccio, “Understanding and Controlling Software Costs,” *IEEE Softw.*, 1988.
- [16] C. Jones, *Applied Software Measurements*. 1997.
- [17] K. M. Eisenhardt and D. Sull, “Strategy as Simple Rules,” *Harv. Bus. Rev.*, vol. 79, no. 1, pp. 06–119, 2001.

- [18] M. Popendieck, "Lean Programming," 2001. [Online]. Available: <http://www.leanessays.com/2010/11/lean-programming.html>.
- [19] M. Fowler, "The New Methodology," 2005. [Online]. Available: <https://www.martinfowler.com/articles/newMethodology.html>.
- [20] A. MacCormack, "Product-Development Practices That Work : How Internet Companies Build Software," 2001.
- [21] A. Cockburn and J. Highsmith, "Agile Software Development: The People Factor," *IEEE Softw.*, 2001.
- [22] L. Constantine, "Methodological Agility," *IEEE Software*, 2001.
- [23] S. Ambler, "When Does(n't) Agile Modeling Make Sense?," 2005. [Online]. Available: <http://agilemodeling.com/essays/whenDoesAMWork.htm>.
- [24] B. Boehm, "Get Ready for Agile Methods , with Care," *IEEE Softw.*, 2002.
- [25] D. Karlström and P. Runeson, "Combining Agile Methods with Stage-Gate Project Management," *IEEE Softw.*, pp. 43–49, 2005.
- [26] R. G. Cooper and A. F. Sommer, "Agile-Stage-Gate: New idea-to-launch method for manufactured new products is faster, more responsive," *Ind. Mark. Manag.*, 2016.
- [27] D. Karlström and P. Runeson, "Integrating Agile Software Development into Stage-Gate Managed Product Development," *Empir. Softw. Eng.*, vol. 11, pp. 203–225, 2006.
- [28] T. Reynisdottir, "Scrum in Mechanical Product Development Case Study of a Mechanical Product Development Team using Scrum," 2013.
- [29] "Burn-Down chart," 2017. [Online]. Available: https://en.wikipedia.org/wiki/Burn_down_chart.
- [30] R. G. Cooper, "Agile-Stage-Gate Hybrids.," *Res. Technol. Manag.*, vol. 59, no. 1, pp. 21–29, 2016.
- [31] R. G. Cooper and A. F. Sommer, "The Agile-Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity," *J. Prod. Innov. Manag.*, vol. 33, no. 5, pp. 513–526, 2016.
- [32] R. G. Cooper, "New Products: What Separates the Winners From the Losers and What Drives Success?," *PDMA Handb. New Prod. Dev. 3Rd Ed.*, 2013.
- [33] W. Isaacson, *Steve Jobs: The exclusive biography*. New York: Simon & Schuster. 2011.
- [34] S. Denning, "How Agile can transform manufacturing : the case of Wikispeed,"

2014.

- [35] M. Jakubowski, “Open source ecology,” 2017. [Online]. Available: http://opensourceecology.org/wiki/Wikispeed_SGT01.
- [36] Wikispeed, “Wikispeed,” 2008. [Online]. Available: <http://www.wikispeed.com>.
- [37] D. J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*. 2009.
- [38] W. J. Watson, ““The “Success” of the Combined Cycle Gas Turbine’ Opportunities and Advances in International Power Generation,” 1996.
- [39] Marel, “Marel GRB,” 2017. [Online]. Available: <http://marel.com/>.
- [40] ANDRITZ, “Andritz Hydro AB,” 2017. [Online]. Available: <http://www.andritz.com/>.
- [41] N. Ovesen, “The Challenges of Becoming Agile,” Aalborg University, 2012.
- [42] LEGO, “LEGO Educational,” 2011. [Online]. Available: <http://www.legoeducational.com>.